



Technische Universität München
Ingenieur fakultät Bau Geo Umwelt
Professur für Computational Mechanics

LEVEL SET TOPOLOGY OPTIMIZATION FOR
CRASHWORTHINESS USING EVOLUTIONARY ALGORITHMS AND
MACHINE LEARNING

Mariusz Bujny

Vollständiger Abdruck der von der Ingenieur fakultät Bau Geo Umwelt der Technischen
Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:

Prof. Dr.-Ing. Kai-Uwe Bletzinger

Prüfer der Dissertation:

1. Prof. Dr.-Ing. habil. Fabian Duddeck
2. Prof. Dr. Thomas Bäck
3. Prof. Dr. ir. Fred van Keulen

Die Dissertation wurde am 07.04.2020 bei der Technischen Universität München eingereicht und
durch die Ingenieur fakultät Bau Geo Umwelt am 22.09.2020 angenommen.

Zusammenfassung

Aufgrund der wachsenden Komplexität des Entwicklungsprozesses in der Automobilindustrie wird es immer schwieriger, sich bei der Konstruktion von Fahrzeugkomponenten ausschließlich auf die ingenieurtechnische Intuition zu verlassen. Gleichzeitig ist der Einsatz von Simulationsmethoden in der Industrie mittlerweile ein gängiger Standard, der zu einer Transformation des traditionellen Entwurfsprozesses hin zum modellbasierten Konzept führt. Infolgedessen wird die automatische Erzeugung mechanischer Strukturen auf der Basis von Optimierungsalgorithmen in der Praxis immer häufiger eingesetzt. Insbesondere die Formulierung einer Entwurfsaufgabe als Problem der Topologieoptimierung bietet dem Computerprogramm die größte Flexibilität und ermöglicht eine Umverteilung des Materials innerhalb eines bestimmten Entwurfsraums, um die Leistungsmetriken für die definierten Belastungsbedingungen zu maximieren. Die meisten häufig verwendeten Methoden zur Topologieoptimierung nutzen analytische Sensitivitätsinformationen, um eine effiziente gradientenbasierte Optimierung durchzuführen, auch bei Problemen mit Millionen von Entwurfsvariablen. Einige wichtige Optimierungsprobleme, wie z. B. die strukturelle Crashtaughigkeit, weisen jedoch eine sehr hohe Komplexität auf, was sich in einer starken Nichtlinearität, einem hohen Grad an numerischem Rauschen, Bifurkationen und Diskontinuitäten der betrachteten Zielfunktionen und Nebenbedingungen widerspiegelt. Daher sind die gradientenbasierten Methoden in solchen Fällen normalerweise nicht direkt anwendbar und stattdessen werden alternative Ansätze verwendet, die auf starken Vereinfachungen bei der Modellierung oder heuristischen Annahmen beruhen. Infolgedessen sind sie hauptsächlich für bestimmte Fälle geeignet, und es sind weitere Untersuchungen zur Entwicklung allgemeinerer Methoden erforderlich. In dieser Arbeit wird daher ein Ansatz zur Topologieoptimierung vorgeschlagen, der auf evolutionären Algorithmen und einer niedrigdimensionalen Level-Set-Darstellung basiert und die Optimierung beliebiger quantifizierbarer Kriterien mithilfe expliziter Crash-Simulationen mit hoher Wiedergabetreue ermöglicht. Die Methode wird auf Grundlage von standardmäßigen linear-elastischen Benchmark-Problemen validiert und mit den aktuell vorhandenen Optimierungsmethoden für Crashtopologien unter Verwendung akademischer Testfälle sowie eines industriellen Optimierungsproblems verglichen. Die numerischen Experimente zeigen, dass mit dem vorgeschlagenen Verfahren erheblich bessere Strukturen erzielt werden können, jedoch auf Kosten einer hohen Anzahl notwendiger Crash-Simulationen. Um dies zu mildern, wird in dieser Arbeit vorgeschlagen, die verfügbaren Informationen effizient zu nutzen, indem maschinelles Lernen auf verschiedenen Ebenen des Optimierungsprozesses integriert wird: zur Modellierung der Antworten aus den teuren Crash-Simulationen, zur Approximation von Sensitivitätsinformationen, die in einer hybri-

den gradientenverstärkten evolutionäre Optimierung verwendet werden, und zur Vorhersage günstiger topologischer Variationen innerhalb einer adaptiven Topologieoptimierung. Die Ergebnisse zeigen, dass durch die Einbeziehung von Techniken des maschinellen Lernens in die evolutionäre Optimierung signifikante Leistungsverbesserungen erzielt werden können. Schließlich ermöglicht der generische Charakter der vorgeschlagenen Verfahren möglicherweise die Bewältigung eines breiten Spektrums von nicht standardmäßigen Strukturoptimierungsaufgaben, einschließlich Problemen bei der Fertigung, der Soft-Robotik, dem Entwurf komplizierter Systeme und vielen anderen.

Abstract

Due to the rising complexity of the development process in the automotive industry, it becomes very difficult to rely solely on the engineering intuition when designing car components. At the same time, the use of simulation methods in industry is now a common standard, leading to a transformation of the traditional design process towards the model-based concept. As a consequence, automatic generation of mechanical structures based on optimization algorithms is more and more frequently used in practice. In particular, formulation of a design task as a topology optimization problem gives the computer program the most flexibility, allowing for a redistribution of the material within a given design space, to maximize performance metrics for the defined loading conditions. Most of the commonly used topology optimization methods utilize analytical sensitivity information to perform efficient gradient-based search even for problems involving millions of design variables. However, some important optimization problems, such as the ones in structural crashworthiness, exhibit very high complexity, reflected in strong nonlinearity, high levels of numerical noise, bifurcations, and discontinuities of the considered objectives and constraints. Hence, the gradient-based methods are usually not directly applicable in such cases and alternative approaches, based on strong simplifications in the modeling or heuristic assumptions, are used instead. As a result, they are suitable mainly for specific cases, and a further research to develop more general methods is needed. Therefore, this thesis proposes a topology optimization approach based on evolutionary algorithms and a low-dimensional level-set representation, allowing for an optimization of arbitrary quantifiable criteria using high-fidelity explicit crash simulations. The method is thoroughly validated based on the standard linear elastic benchmark problems and compared to the state-of-the-art crash topology optimization methods using academic test cases as well as a real-world optimization problem. The numerical experiments show that considerably better structures can be obtained with the proposed method, however, at the cost of a high number of necessary crash simulations. To mitigate that, this thesis proposes to use efficiently the available information by integrating machine learning at different levels of the optimization process: for modeling of the responses from the expensive high-fidelity simulations, for approximating sensitivity information used in a hybrid gradient-enhanced evolutionary approach, and for prediction of favorable topological variations within an adaptive topology optimization. The results show that significant performance improvements can be obtained by incorporating machine learning techniques into the evolutionary search. Finally, the generic character of the proposed methods allows potentially for addressing a wide spectrum of non-standard structural optimization tasks, including problems in manufacturing, soft robotics, compliant mechanism design, and many others.

Acknowledgements

The research presented in this dissertation was conducted during my time as a Research Assistant at the Associate Professorship of Computational Mechanics of the Department of Civil, Geo and Environmental Engineering at the Technical University of Munich (TUM) from December 2015 until November 2018. The project was financed by the Honda Research Institute Europe GmbH (HRI-EU).

First of all, I wish to express my gratitude to my supervisors, Prof. Fabian Duddeck, Dr. Markus Olhofer, and Dr. Nikola Aulig for all their ideas, constant support, guidance, and optimistic attitude over the past years. This all would not be possible without the trust put in me by them and HRI-EU president Andreas Richter, who gave me the chance to work on this project. I would like to thank my HRI-EU advisor, Dr. Michael Gienger, and TUM mentor, Prof. Thomas Bäck, for the stimulating discussions and precious help in managing this project. For all the support in the final phase of this project, I would like to thank also Dr. Stefan Menzel, who, apart from my supervisors, motivated me strongly to finalize the writing of this thesis. Moreover, I wish to thank Prof. Fred van Keulen for completing the examination committee and taking his valuable time to read and assess this (voluminous) work.

For all the inspiring discussions and fruitful collaboration, I would like to thank my colleagues from the Associate Professorship of Computational Mechanics. In particular, I would like to thank Koushyar Komeilizadeh, Pablo Lozano, Dr. Duo Zeng, and Dr. Erich Wehrle, for the warm welcome at TUM and advice at the beginning of my PhD project. I really miss our discussions about politics, economics, finance, and world in general.

To all of my colleagues from HRI-EU, where I had the chance to work for over two years as a guest PhD student: being a part of this excellent environment was (and luckily still is) a great privilege for me. Especially, I would like to express my appreciation to the members of the Optimization & Creativity group, for all of the great ideas and the nice time we spent together drinking coffee and practicing German.

To all the students I was advising: Georgios Athanasiou, Nivesh Dommaraju, Lukas Krischer, Simon Kohlhase, Aude Mulard, Wei Ni, Elena Raponi, Muhammad Salman Yousaf – I hope they could learn from me as much as I learned from them. In particular, I would like to thank Elena Raponi and Lukas Krischer for their excellent work, which strongly contributed to this dissertation.

For the inspiring meetings and fascinating research collaboration, showing the true potential of the topology optimization methods we developed, I would like to thank our partners from the Honda Research & Development Americas and the Ohio State University: Duane Detwiler, Nathan Zurbrugg, Sanjaya Fonseka, and Satchit Ramnath. Visiting you was one of the most exciting experiences during my professional career.

I would like to thank my friends and family for all the support, despite my neglect over the past years. Especially, I am grateful to my wife, Anna, for her love and support in dealing with all the happy and difficult moments every single day. Last but not least, I would like to thank my parents, for the upbringing, awakening my interest in science, and all the support in my education.

Finally, to all the silent helpers I did not mention here: thank you for being there for me!

Contents

1	Introduction	23
2	State of the art.....	27
2.1	Structural topology optimization.....	27
2.2	Crashworthiness topology optimization	29
2.2.1	Equivalent static loads (ESL) methods.....	29
2.2.2	Ground structure approaches (GSAs)	31
2.2.3	Bubble and graph/heuristic approaches (GHAs)	32
2.2.4	Hybrid cellular automata (HCA) methods	33
2.2.5	Bi-directional evolutionary structural optimization (BESO) approaches.....	35
2.2.6	State-based representation (SBR) approaches	35
2.3	Evolutionary computation in structural topology optimization	36
2.4	Machine learning in structural topology optimization	37
2.4.1	Optimal design prediction.....	38
2.4.2	Representation learning	39
2.4.3	Sensitivity learning	39
2.5	Aim and objectives	40
3	Fundamentals of evolutionary computation.....	43
3.1	Evolutionary computation.....	43
3.2	Evolution strategies (ESs).....	44
3.3	Covariance matrix adaptation evolution strategy (CMA-ES)	47
3.4	Summary	47
4	Representations for evolutionary-based structural topology optimization	49
4.1	Taxonomy.....	50
4.2	Properties of the fitness landscape.....	51
4.3	Correlation of parameters and additional aspects	54
4.4	Conclusions	58
5	Evolutionary level set method (EA-LSM)	61
5.1	Introduction	61
5.2	Parametrization	63
5.3	Optimization algorithm	68
5.3.1	Algorithm overview	68
5.3.2	Initialization.....	69

5.3.3	Repair operators	71
5.3.4	Constraint handling.....	77
5.3.5	Normalization of design variables.....	78
5.3.6	Symmetries.....	80
5.4	Summary	80
6	Machine learning-based enhancements of EA-LSM.....	83
6.1	Exploiting approximate gradient information	84
6.1.1	Sensitivity analysis for estimation of approximate gradients.....	85
6.1.2	Gradient approximation via global equivalent static loads method	91
6.1.3	Gradient approximation by predicting sensitivities.....	92
6.1.4	Hybrid evolutionary algorithms	96
6.2	Utilizing surrogate modeling techniques.....	98
6.2.1	Kriging-guided level set method (KG-LSM)	99
6.3	Adaptive evolutionary level set method (A-EA-LSM)	107
6.3.1	Graph-based parametrization.....	108
6.3.2	Optimization algorithm	109
6.3.3	Learning-based topology variation	113
6.4	Summary	120
7	Validation by linear elastic cases.....	123
7.1	Two-dimensional topology optimization for minimum compliance	124
7.1.1	Evolutionary level set method (EA-LSM)	125
7.1.2	Hybrid evolutionary level set method (H-EA-LSM)	142
7.1.3	Kriging-guided level set method (KG-LSM)	147
7.1.4	Adaptive evolutionary level set method (A-EA-LSM)	154
7.2	Three-dimensional topology optimization for minimum compliance	160
7.2.1	Optimization problem	161
7.2.2	Results	161
7.3	Summary	167
8	Optimization of nonlinear crash cases.....	169
8.1	Two-dimensional topology optimization for crashworthiness	170
8.1.1	Evolutionary level set method (EA-LSM)	174
8.1.2	Hybrid evolutionary level set method (H-EA-LSM)	185
8.1.3	Kriging-guided level set method (KG-LSM)	194
8.1.4	Adaptive evolutionary level set method (A-EA-LSM)	197
8.2	Three-dimensional topology optimization for crashworthiness	200
8.2.1	Optimization problem	200
8.2.2	Results	202
8.3	Summary	208

9	Industrial applications with EA-LSM	211
9.1	Concept crash structures using additive manufacturing	212
9.2	Test case and optimization problems	213
9.3	Mass minimization under a node displacement constraint	220
9.4	Robustness.....	226
9.5	Mass minimization under an acceleration constraint.....	231
9.6	Summary	239
10	Conclusions and future work	241
10.1	Summary	241
10.2	Outlook	246
	Appendices	249
A	Methods.....	249
A.1	Features for learning-based topology variations	249
B	Algorithm settings	251
B.1	Parameter settings for SIMP.....	251
B.2	Parameter settings for CMA-ES.....	252
C	Results	253
C.1	Design evolution in EA-LSM for the cantilever beam case.....	253

List of Abbreviations

A-EA-LSM	Adaptive Evolutionary Level Set Method
AI	Artificial Intelligence
AM	Additive Manufacturing
ANN	Artificial Neural Network
BESO	Bi-directional Evolutionary Structural Optimization
CA	Cellular Automata
CAD	Computer-Aided Design
CEI	Constrained Expected Improvement
CHES	Concurrent Hybrid Evolution Strategy with Gradient Individual
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
CNN	Convolutional Neural Network
CSF	Component State Feature
DE	Differential Evolution

DL	Deep Learning
DoE	Design of Experiments
EA	Evolutionary Algorithm
EA-LSM	Evolutionary Level Set Method
EC	Evolutionary Computation
EGO	Efficient Global Optimization
EI	Expected Improvement
EP	Evolutionary Programming
ES	Evolution Strategy
ESL	Equivalent Static Load
ESO	Evolutionary Structural Optimization
FD	Finite Difference
FE	Finite Element
FEA	Finite Element Analysis
FEM	Finite Element Method
FFS	Full Factorial Sampling
GA	Genetic Algorithm
GAN	Generative Adversarial Network
GHA	Graph/Heuristic Approach
GHES	Global Hybrid Evolution Strategy with Gradient-based Local Search
GP	Gaussian Process

GSA	Ground Structure Approach
HCA	Hybrid Cellular Automata
HCATWS	Hybrid Cellular Automata for Thin-Walled Structures
HCMA-ES	Hybrid CMA-ES Method
H-EA-LSM	Hybrid Evolutionary Level Set Method
HIC	Head Injury Criterion
KG-LSM	Kriging-Guided Level Set Method
k-NN	k-Nearest Neighbors
LHS	Latin Hypercube Sampling
LR	Linear Regression
LSF	Level Set Function
LSM	Level Set Method
ML	Machine Learning
MLP	Multi-Layer Perceptron
MMC	Moving Morphable Component
MPI	Message Passing Interface
NURBS	Non-Uniform Rational Basis Spline
OC	Optimality Criteria
OLHS	Optimal Latin Hypercube Sampling
PCA	Principal Component Analysis
RBF	Radial Basis Function

RDO	Robust Design Optimization
SBR	State-Based Representation
SD	Steepest Descent
SIMP	Solid Isotropic Material with Penalization
SVM	Support Vector Machine
SVR	Support Vector Regression
TO	Topology Optimization
TOPS	Topology Optimization by Predicting Sensitivities
VAE	Variational Autoencoder

List of Symbols

a	Individual in an evolutionary algorithm
<i>a</i>	Acceleration magnitude
<i>B</i>	Penalty function used in KG-LSM
<i>B</i> ₁ , <i>B</i> ₂ , <i>B</i> ₃	Levels of constraint violation used for computation of <i>B</i> to penalize structurally infeasible designs in KG-LSM
$\alpha_i, \beta_i, \gamma_i$	Rotation angles of the <i>i</i> th MMC in 3D (Euler angles)
<i>C</i>	Compliance
<i>c</i>	Convergence velocity of an algorithm
CSF	Vector of Component State Features (CSFs)
<i>D</i>	Design domain
<i>d</i>	Intrusion magnitude
<i>d</i> _{ad}	Damping parameter in a derandomized step size adaptation
<i>d</i> _{dd}	Depth of the design domain in TO

δ	Exponent used in the exterior penalty method
E	Young's modulus
\mathbf{E}^0	Reference stiffness tensor
\mathbf{E}	Stiffness tensor
$\boldsymbol{\varepsilon}$	Elastic strain
ε	Tolerance value used in the exterior penalty method
f	Cost function
\mathbf{f}	Vector of nodal forces
f_{obj}	Objective function
$f_{obj_{opt}}$	Optimized value of the objective function
F	Force magnitude
Γ_t	Surface of a body
g_{C_j}	j^{th} inequality constraint
h_{C_k}	k^{th} equality constraint
h_{dd}	Height of the design domain in TO
h_i	Height of the i^{th} MMC in 3D
I	Improvement
\mathbb{I}	Space of individuals
$iter$	Iteration counter used in the optimization algorithms
$iter_{max}$	Maximal number of iterations of an optimization algorithm
\mathbf{K}	Stiffness matrix

κ	Penalty factor used in KG-LSM
λ	Number of offspring individuals
λ_V	Lagrange multiplier for volume
l_i	Length of the i^{th} MMC
L_k	Penalty for the k^{th} equality constraint in the exterior penalty method
m	Mass of the structure
M	Number of level set basis functions / MMCs
m_{req}	Required mass of the structure
μ	Number of parent individuals
μ_s	Mean of a surrogate model
n	Dimensionality of the optimization problem
n_s	Number of samples for training of machine learning (surrogate) models
n_{eq}	Number of equality constraints
n_{ineq}	Number of inequality constraints
n_p	Number of parameters per basis function / MMC
ν	Poisson's ratio
o	Component overlap ratio
Ω	Part of the design domain occupied by the material
$\mathcal{P}(iter)$	Population of individuals at generation $iter$
Φ	Global level set function

Φ_G	Gaussian cumulative distribution function
ϕ_G	Gaussian probability density function
ϕ_i	i^{th} local level set function
P_j	Penalty for the j^{th} inequality constraint in the exterior penalty method
$\partial\Omega$	Boundary of the material phase
Ψ	Correlation matrix
q	Exponent used in the formula for the local level set function
$R_{t_1}, R_{t_2}, R_{t_3}$	Rotation matrices
res	Simulation residual
ρ	Normalized density used to define material distribution
ρ_{min}	Minimal normalized density
\mathbf{r}_k	Vector of observed responses for the k^{th} model
s	Vector of strategy parameters for ES
σ	Standard deviation in the mutation operator of ES
σ_s	Standard deviation of a surrogate model
σ_{init}	Initial standard deviation in the mutation operator of ES
σ_{min}	Minimal standard deviation in the mutation operator of ES
s	Step size (learning rate) used in the gradient-improvement operator
t	Time

τ_0	Learning rate in ES
τ'	Global learning rate in ES
τ	Local learning rate in ES
θ_i	Rotation angle of the i^{th} MMC in 2D
t_i	Thickness of the i^{th} MMC
t_{min}	Minimal thickness of an MMC
\mathbf{t}	Boundary tractions
\mathbf{U}	Space of kinematically admissible displacements
\mathbf{u}	Vector of nodal displacements
V	Volume of the structure
v	Velocity magnitude
V_f	Volume fraction
V_{req}	Required volume of the structure
w_{dd}	Width of the design domain in TO
\mathbf{x}	Point in the 2D/3D design domain, i.e. $\mathbf{x} = [x, y]^T$ or $\mathbf{x} = [x, y, z]^T$
x_{0i}	x -coordinate of the center of the i^{th} MMC
x_{end1i}, x_{end2i}	x -coordinates of the ends of the i^{th} MMC
x', y', z'	Coordinates in a transformed coordinate system
y_{0i}	y -coordinate of the center of the i^{th} MMC
y_{end1i}, y_{end2i}	y -coordinates of the ends of the i^{th} MMC
\mathbf{z}	Vector of design variables

z_{0i}	z -coordinate of the center of the i^{th} MMC
z_{end1i}, z_{end2i}	z -coordinates of the ends of the i^{th} MMC
ζ	Step length in the finite-differencing
$\mathbf{z}^{(i)}$	i^{th} sample of a design variable vector
z_i	i^{th} component of the vector of design variables
z_i^L	Lower bound on the i^{th} component of the vector of design variables
\mathbf{z}_{opt}	Vector of design variables for which the objective function reaches the optimum
z_i^U	Upper bound on the i^{th} component of the vector of design variables
\mathbf{Z}	Set of sampled vectors of design variables for KG-LSM

Chapter 1

Introduction

For decades, automotive industry has been striving for lightweight design. Through a tremendous effort of generations of engineers, we were able to develop very complex vehicle structures, meeting ambitious requirements concerning fuel consumption, driving comfort, and safety. So far, the design process was based mostly on trial and error as well as engineering intuition and know-how, collected over all the past years. Nowadays, the design targets for the car industry are even more demanding. Shorter design cycles, very strict CO₂ emission regulations, as well as high expectations of the customers regarding the comfort and safety, make the design process highly interdisciplinary and challenging. As a result, very often it is difficult to rely on intuition when designing complex car components, which have to meet stringent requirements related to a growing number of versatile disciplines.

In particular, among different design aspects, vehicle crashworthiness plays a very important role. Each year, ca. 1.2 million people die in car accidents worldwide (Fang et al. (2016)) and many more are severely injured, which makes vehicle crash one of the main health concerns of a modern society, with major socioeconomic implications. Despite the huge improvement in terms of vehicle crashworthiness within the last decades, strongly related to the development of numerical simulation methods such as the Finite Element Method (FEM), crashworthiness design remains one of the most challenging fields of structural engineering, involving highly complex physical phenomena. As such, it poses severe difficulties for the human designers, with crash components frequently exhibiting very non-intuitive mechanical behavior.

With the development of numerical simulation methods as well as the growing power of com-

putational hardware, so-called simulation-driven design becomes an attractive alternative to the traditional trial-and-error design process. By formulating the design problem as a mathematical optimization task, simulation software can be used to provide an optimal design for given criteria in a fully automated way. As a consequence, the development process can be considerably accelerated and the number of costly prototypes reduced.

Among different formulations of optimization problems for structural design, Topology Optimization (TO) is the most complex one, giving the computer program the full freedom to redistribute material within a predefined design space, for given loading conditions. In particular, crashworthiness TO belongs to the most difficult classes of TO problems, due to highly nonlinear character of the optimization objectives and constraints, which are usually highly multi-modal, involve discontinuities, bifurcations, and high levels of numerical noise. This results directly from the high-fidelity simulation models used today, which account for nonlinear material behavior, buckling, large deformations, contact, or even material failure. Consequently, standard gradient-based TO approaches are not applicable here and alternative methods have to be developed.

Due to its practical relevance, complexity, and a high potential for improvements, crashworthiness TO constitutes a very important and interesting field of research. This work primarily focuses on development of crash TO methods addressing the shortcomings of the existing approaches, which, to mitigate computational cost, rely on simplifications of the load cases or utilization of heuristic update rules, leading to a considerably limited generality. To achieve that, this thesis presents an alternative way, based on utilization of the methods from the domain of Artificial Intelligence (AI), such as evolutionary computation, memetic algorithms, supervised learning for regression and classification problems. To guarantee high computational efficiency, manufacturability, as well as correct physical behavior of the structures in the crash simulations, a low-dimensional representation based on a level-set description is used.

The proposed methods are validated on linear elastic static problems and applied to nonlinear crash scenarios, considering both 2D and 3D design spaces. Finally, the core method proposed in this dissertation, Evolutionary Level Set Method (EA-LSM), is used for a real-world design problem of a conceptual hybrid S-rail with an additively-manufactured metal joint.

Although the methods discussed in this work are developed mainly in the context of crashworthiness applications, all of them have generic character, which makes them potentially also useful in other fields of structural TO, where the gradient-based approaches are not applicable. Moreover, these approaches can work with commercial, black-box simulation software, which allows for embedding them in established workflows. Finally, the proposed methods can be also applied in case of the problems where gradient-based methods are used, however, the considerable computational effort would likely prohibit that.

This dissertation is structured as follows. In Chapter 2, the current state of the art of crashworthiness TO, as well as the methods based on evolutionary optimization and utilizing machine learning in structural TO, are discussed. In Chapter 3, a short introduction to Evolutionary Computation (EC), necessary for understanding the methods developed in this work, is given. Chapter 4 provides an intuitive justification of the choices made on the representation level for evolutionary-based structural TO. Chapter 5 introduces the main contribution of this thesis, the EA-LSM. In Chapter 6, machine-learning-based enhancements of EA-LSM are discussed, including the concept of an adaptive EA-LSM. To validate the proposed methods, in Chapter 7, an evaluation on linear elastic cases is presented. Subsequently, in Chapter 8, the methods are evaluated on non-linear crash cases. Finally, Chapter 9 addresses the question of feasibility of using EA-LSM in industrial setting, by optimizing the topology of a 3D-printed metal joint in a hybrid S-rail. Chapter 10 concludes the work and discusses promising directions for the future research.

Chapter 2

State of the art

This chapter summarizes the state of the art with a special focus on methods relevant to this work. First of all, structural TO is briefly introduced in Section 2.1. In Section 2.2, state-of-the-art TO methods for crashworthiness are discussed in more detail. Particular attention is paid to the generality of the methods, to motivate the solutions proposed in this work. Subsequently, Section 2.3 discusses the existing structural TO methods using Evolutionary Algorithms (EAs). Finally, since a significant part of this work uses Machine Learning (ML) approaches to enhance the optimization process, a critical review of the recent methods using ML in TO is given in Section 2.4. The presented discussion leads directly to the definition of aim and objectives of this work, described in Section 2.5.

2.1. Structural topology optimization

Structural TO targets finding optimal material distribution for given criteria, within a defined design domain, and under specified boundary conditions. These methods are used nowadays mainly in early design phases, as an inspiration for the novel design concepts. Recently, with the advances in integrating manufacturing constraints into TO (Langelaar (2016); Liu and Ma (2016); Vatanabe et al. (2016); Gaynor and Guest (2016)) and the progress in Additive Manufacturing (AM), TO is considered also as a tool for the part design. In general, TO methods can be divided into two main categories, i.e. density-based (Bendsøe and Sigmund (2004)) and level-set methods (Dijk et al. (2013)), which are discussed below.

The most commonly used structural TO methods are the density-based approaches, including homogenization methods (Suzuki and Kikuchi (1991); Hassani and Hinton (1998)) and SIMP¹-based methods (Bendsøe (1989); Zhou and Rozvany (1991); Rozvany et al. (1992)). In density-based methods, the design domain is divided into a mesh of parametrized elements. The parameters, e.g. element densities scaling the Young's modulus, are adjusted by the optimization algorithm to minimize given objective functions, while satisfying specified constraints. Such parametrization results typically in thousands, up to millions of design variables, which are modified by gradient-based optimizers (Bendsøe and Sigmund (2004); Aage et al. (2015)) or heuristic² methods (Xie and Steven (1997); Yang et al. (1999); Tovar (2004)). Highly efficient gradient-based methods allow for addressing various problems, involving such criteria as compliance minimization (Sigmund (2014)), stress constraints (Duysinx and Bendsøe (1998); Bruggi and Duysinx (2012)), eigenmode maximization (Pedersen (2000)), maximization of fundamental buckling load (Neves et al. (1995)), pressure loads (Hammer and Olhoff (2000)), compliant mechanisms (Sigmund (1997)), and many others. Heuristic methods (Xie and Steven (1997); Yang et al. (1999); Tovar (2004)) are typically used in situations when analytical sensitivities are not available. To overcome typical problems with density-based approaches, e.g. checkerboard patterns, mesh dependency, and intermediate densities, different techniques have been developed (Bendsøe and Sigmund (2004)). However, the problem of intermediate densities still appears in most cases, even when the SIMP interpolation scheme is used. As a result, level-set methods have been proposed, to guarantee development of clear boundaries between material and void.

In Level Set Methods (LSMs) (Sethian and Wiegmann (2000); de Ruiter and van Keulen (2004); Allaire et al. (2004, 2005)), an implicit level-set function is used to describe material distribution, and its 0th iso-contour defines the location of the material interface. The clear definition of material boundaries is crucial from the point of view of manufacturability of the optimized topologies and resulted in a growing popularity of LSMs. In standard LSMs, the level-set function is updated via solving the Hamilton-Jacobi partial differential equation (Allaire et al. (2004)), which incorporates analytically derived sensitivities. The level-set function is usually defined in terms of local basis functions, e.g. FEM basis functions (Dijk et al. (2012)), Radial Basis Functions (RBFs) (de Ruiter and van Keulen (2004); Wang and Wang (2006)), or spectral parametrization (Gomes and Suleman (2006)), which highly influence the smoothness of the material distribution.

Taking advantage of the main concept of LSMs, i.e. definition of the material distribution based on an implicit function, different techniques using parametrized geometric shapes as local basis functions emerged (Van Miegroet and Duysinx (2007); Guo (2014); Norato et al. (2015)). From this perspective, those methods can be perceived as a part of a broader class of LSMs (Dijk et al. (2013); Wein et al. (2019)). However, there is a fundamental difference between the standard LSMs

¹ Solid Isotropic Material with Penalization (Bendsøe and Sigmund (2004)).

² Heuristics are methods employing certain rules, formulated usually based on earlier observations. They do not have to lead to an optimal solution, but are sufficient for finding good designs, leading therefore to realization of short-term goals.

(Sethian and Wiegmann (2000); Wang et al. (2003); Allaire et al. (2004)) and the parametrizations based on geometric shapes, where the Hamilton-Jacobi equation is not solved to evolve the material interface, but the variables parametrizing the shapes are changed explicitly by the optimizer. As a consequence, there is no clear consensus in the community regarding classification of this group of methods as LSMs, and different terms were proposed: Moving Morphable Components (MMCs) (Guo (2014)), Geometry Projection (Norato et al. (2015)), or Feature-Mapping Methods (Wein et al. (2019)) suggested recently as a unified name. In this work, however, similarly to our previous publications (Bujny et al. (2016b,c, 2017a, 2018); Raponi et al. (2017)), we follow the oldest classification (Dijk et al. (2013)) and treat this type of methods as a part of a broader category of LSMs.

2.2. Crashworthiness topology optimization

Crashworthiness belongs to the most difficult areas of application of TO. Due to high levels of numerical noise, physical bifurcations, and discontinuities caused by material failure, contacts and large plastic deformations, the analytical derivation of sensitivities is very difficult and even numerical estimates of gradients are not always reliable (Duddeck (2008); Aulig (2017); Ortmann and Schumacher (2013)). Therefore, crash TO is normally not addressed with gradient-based approaches directly, and alternative methods have to be used. In general, the methods for crashworthiness TO can be divided into the following categories:

- Equivalent Static Loads (ESL) methods.
- Ground Structure Approaches (GSAs).
- Bubble and Graph/Heuristic approaches (GHAs).
- Hybrid Cellular Automata (HCA) methods.
- Bi-directional Evolutionary Structural Optimization (BESO) approaches.
- State-Based Representation (SBR) approaches.

Below, the description of the most representative methods and algorithms from these categories is given.

2.2.1. Equivalent static loads (ESL) methods

The earliest attempts of using TO in crashworthiness involved primarily the use of ESL techniques. The main idea in ESL is to replace the dynamic crash case with a set of static loads roughly representing the loading conditions during the crash event. Since dynamic and nonlinear effects are not taken into account in such a case, those methods are appropriate mainly for optimization of structures not undergoing large plastic deformations, e.g. the safety cell. Once static load cases are defined, standard gradient-based TO methods for linear elasticity (Bendsøe and Sigmund (2004)) can be used to minimize the compliance.

One of the most important aspects of ESL techniques is the type of approximation of the dynamic loads. From this perspective, ESL approaches can be divided into global and local ESL methods. In the global ESL methods, the static loads cover larger areas of the design domain and are usually defined coarsely based on the individual judgement of the optimization expert. In the early approaches (Cavazzuti et al. (2010); Christensen et al. (2012)), each of the dynamic load cases was approximated with just a single static load case, thus not capturing the variations of the load over time in the approximated crash scenario. Each ESL case was considered separately and the structure was optimized simultaneously for different loading conditions by considering a multi-objective optimization problem. Later on, Duddeck and Volz (2012) proposed an extension, where several static load cases were used to represent different stages of the crash event. This accounted both for the variations of the loads in time and location, so this approach is much more suitable for grasping the dynamic characteristics of the crash event. Moreover, the derivation the ESLs in this case was relying on general energy considerations, based on average force levels and free crush lengths in the energy-absorbing zones, which much better reflected the requirements for structural crashworthiness.

The second group of ESL methods, i.e. the local ESL approaches (Yi et al. (2012); Park (2010)), rely on a much more fine-grained definition of the static loads. The main idea is to use the nonlinear crash simulation to compute the nonlinear displacements at different moments of the crash event. In the next step, the nonlinear displacements are multiplied by the corresponding linear stiffness matrix, yielding a vector of the nodal forces. The application of the resulting nodal forces to a linear elastic case should obviously result in the same displacements as in the nonlinear crash scenario. Each of the points of time during the nonlinear crash simulation is considered separately and, therefore, the multiplication of the displacements by the stiffness matrix results in several independent load cases. Similarly to the global ESL approach proposed by Duddeck and Volz (2012), multi-load-case TO techniques are used to derive the optimal design. Once the topology is optimized for the first set of ESL cases, the behavior of the structure in a crash scenario is simulated and the process is repeated. The entire optimization is therefore realized in two nested loops – in the outer loop the crash simulation is performed and the equivalent static load cases are extracted, while in the inner loop, gradient-based multi-load-case TO for linear elastic statics is applied to optimize the structure for the current loading conditions. Since the overall process is fully automated, the implementation of this method in engineering software is possible and was done in GENESIS (Vanderplaats (2019)). Several works on optimization of automotive structures based on this software have been published and the results were promising (Erhart et al. (2012); Müllerschön et al. (2013); Salway and Zeguer (2013); Kim et al. (2017); Karev et al. (2018, 2019)). However, even the local ESL approach, potentially offering much better accuracy in representing the nonlinear crash behavior than the global ESL, relies on the fundamental assumption that the crash loads can be replaced with static forces, which is questionable in general case. Moreover, it is arguable if the stiffness matrix corresponding to the undeformed structure can be used to derive

the nodal forces (Duddeck et al. (2016)). Another important assumption in the local ESL is that the minimization of the compliance in the inner loop would guide the optimization towards the optimum of the problem addressed in the nonlinear crash case. Finally, due to the computation of the equivalent loads on the nodal level, element deletion, which is crucial for crash TO to avoid unphysical behavior, should be taken with a special care.

Although ESL techniques are computationally very efficient, the simplifications they use severely restrict the applicability of those methods in general case. Regardless of that, they remain one of the main approaches used in the industry for large-scale TO, thanks to their simplicity and the low computational cost.

2.2.2. Ground structure approaches (GSAs)

Similarly to ESL methods, GSAs belong also to one of the oldest approaches for crash TO. In GSA, the optimized part, e.g. a car body structure, is modeled as a system of elementary macro-elements (beams), which exhibit a simplified crash behavior. That is a clear advantage over the ESL approaches, since nonlinear effects can be taken into account. The optimization starts with a design composed of a dense lattice of macro-elements, referred to as the ground structure. For the simplified crash simulation model, the analytical sensitivities can be derived, and efficient gradient-based optimization techniques can be used to find the best topological concept. Usually, size parameters, such as thicknesses of the macro-elements, are used as design variables. The elements are deleted from the ground structure if the thickness drops below a certain threshold.

In particular, the methods proposed by Fredricson et al. (2003), Pedersen (2003), and Torstenfelt and Klarbring (2007) belong to the most representative GSAs. Fredricson et al. (2003) proposed to use GSA in optimization of frame structures with flexible joints. However, only static load cases were addressed here and the method was applied only to small academic examples. Pedersen (2003) applied GSA to achieve a predefined energy absorption history in a simplified crash scenario. The simplification involved the use of quasi-static finite element (FE) solver for a 2D frame composed exclusively of beam elements and plastic hinges. The macro-elements could therefore undergo large translations and rotations, allowing for much more realistic modeling of the crash behavior than in the ESL methods. Finally, Torstenfelt and Klarbring (2007) used GSA for optimization of a 3D car body structure relying on a parametrized model composed of beams and joints. However, also in this case, only static load cases and equivalent static counterparts of the crash load cases were considered.

The main limitation of the GSAs remains the accuracy of the simulation models. In particular, contact modeling is not taken into account and the simplified models require significant calibration effort. Moreover, the final result of the optimization strongly depends on a particular layout and the number of macro-elements in the ground structure. The parametrizations used usually allow only for changes in size, and not in shape or positions of the elements, which also significantly restricts the number of design concepts that can be derived.

2.2.3. Bubble and graph/heuristic approaches (GHAs)

Another important class of methods for crash TO are bubble as well as graph and heuristic approaches (GHAs). Crashworthiness TO based on the bubble method (Eschenauer et al. (1994); Eschenauer and Schumacher (1997)) is a three-stage approach. Firstly, for a given topology of the initial design, shape optimization of its geometry is carried out. Secondly, the topology of the design is changed by insertion of an infinitesimal hole, also referred to as the "bubble", into the structure. The positioning of the hole is based on analytical expressions proposed by Schumacher (2005). Depending on the considered objectives, different criteria for hole positioning can be used. Schumacher (2005) proposes closed expressions for the stiffness maximization and minimization problem. The approach is mathematically justified by a strong relation to the concept of topological derivative (Novotny and Sokołowski (2013)), which is frequently used in level-set structural TO (Allaire et al. (2005); Norato et al. (2007)). Thirdly, the shape of the inserted hole as well as the remaining structural boundaries are optimized. The entire process is repeated until convergence, indicated by insertion of the hole on the variable boundary. The main limitation of the method remains the fact that the criteria for positioning of the holes are based on maximization or minimization of the stiffness of the part for the corresponding linear elastic case. It is not clear how much this is correct in general case. Furthermore, the method has been applied only to relatively simple academic test cases. Taking into account the similarity of the bubble method to the topological derivative, Weider and Schumacher (2018) proposed recently a topological derivative method for crash objectives based on meta-modeling. However, this is still a work-in-progress and additional research on contact and strain rate effects is necessary (Weider and Schumacher (2018)).

The bubble method inspired a development of an approach for graph and heuristic based topology optimization (Ortmann and Schumacher (2013); Ortmann (2015)). In this method, a topology is represented using a graph, whose vertices encode the information about the cross-sections, such as wall thickness or curvature. The edges of the graph are used to store the information about the topological connections of the design. Manufacturing constraints and graph correction checks are also performed in the proposed framework (Ortmann and Schumacher (2013)). In GHAs, crashworthiness criteria are tackled directly to optimize thin-walled components. The method is used mainly for optimization of rib patterns in the thin-walled extrusions (Ortmann and Schumacher (2013); Ortmann (2015)), and recently also to composite profiles (Schneider et al. (2018, 2019)), as well as for optimization of the rib layout on surfaces (Schneider and Schumacher (2018)). In all of the cases, the TO problem remains always two-dimensional. The optimization itself is carried out in two nested loops. The outer loop introduces topological changes by adding new walls within the empty chambers of the profile. The topological modifications are introduced based on heuristic rules extracted from the expert knowledge in the area of crashworthiness design. The rules were formulated based on a wide study (Schumacher and Ortmann (2013)) done in cooperation with major German car companies, where more than 150 engineering design rules were identified.

Based on that, 7 most important heuristics, e.g. "Support Fast Deforming Walls", "Remove Small Chambers", or "Balance Energy Density", were described mathematically and used within GHAs. In the inner loop, an optimization of shape and size of the ribs, for a fixed topology, is carried out using Genetic Algorithms (GAs). The optimization process is continued until the next topological modifications no longer result in the improvements of the optimization criteria.

Although GHAs, unlike the state-of-the-art methods discussed previously, rely on high-fidelity crash simulations and address crash-specific criteria directly, the heuristic rules they use remain arguable. One has to take into account also the higher computational cost of the method resulting from the utilization of GAs in the inner loop. The method has been also applied only to 2D TO, and it is not clear if it could be extended to 3D problems and how would it perform in these cases. Most probably, the heuristic rules would also heavily depend on the type of the optimization problem as well as the design representation used. Moreover, acquiring such rules in the way proposed by Schumacher and Ortmann (2013) is very cumbersome and it is not clear if such rules can be identified in general case. Finally, to the best of our knowledge, the method has never been evaluated on linear elastic static problems, where it could be compared to the standard benchmark cases with the known optima. The topological modifications are restricted exclusively to addition of walls in the empty chambers, i.e. not allowing to cross any other walls, and it is questionable if this process could yield global optima even for simple static examples. Nevertheless, from the point of view of engineering practice, where the global optimality is not of the main interest, GHA can be used successfully to improve the design of crash-relevant components based on explicit crash simulations, which makes it a valuable tool for the industrial applications.

2.2.4. Hybrid cellular automata (HCA) methods

Another category of methods, very popular also in engineering practice, comprises the HCA techniques. HCA methods rely on the concept of Cellular Automata (CA), well known in computer science (Wolfram (2002)). In CA, the space is initially filled with a structured grid of cells, which can take one of a finite number of states. The evolution of the state of the entire system is achieved by successive updates of the states of individual cells, which depend on the states of their neighbors. HCA techniques employ this idea, but the updates are additionally based on the global information about the entire system. This approach has been used for the first time in structural TO as a tool for simulation of the bone remodeling process (Tovar (2004)). Since the method relies on the heuristic optimization criterion, being homogenization of strain energy density, it can be used in the cases when the gradient information is not available. As a result, adaptations of this approach, described below, have been successfully used in TO of crash structures.

Patel (2007) as well as Mozumder (2010) proposed HCA techniques for crash topology and topometry optimization of solid and shell structures, respectively. In these cases, the optimality criterion relies on the assumption that the energy absorption in the finite elements should be homogenized to maximize energy absorption of the entire structure. Similarly to GHAs, these methods use

high-fidelity explicit crash simulations. However, thanks to the used heuristics, the computational costs are considerably lower, which makes them very attractive for the industrial applications. As a consequence, these HCA methods were used as a basis for the development of commercial TO software for crash, LS-TaSC (LSTC (2011)), which has been successfully used in optimization of industrial problems (Salway and Zeguer (2013); Aulig et al. (2018)).

Despite the popularity of the standard HCA approach described above, it is arguable if the energy homogenization criterion can be used for optimization of crash structures in general case. In particular, a lot of crash-relevant car body components are thin-walled structures, e.g. metal crash boxes. It is well known that for this type of structures the energy absorption during crash is maximized via progressive folding (Kanter (2006)), which is characterized by occurrence of local plastic hinges with high energy concentrations. Therefore, the maximization of the overall energy absorption, in this case, does not correspond to homogenization of energy absorption on the element level. In order to address that problem, Hunkeler (2013) proposed a modification of HCA, a Hybrid Cellular Automata for Thin-Walled Structures (HCATWS) approach, where cells consist of groups of shell finite elements, e.g. rectangular walls. The energy absorption is homogenized therefore among the walls, allowing for an inhomogeneous energy distribution on the local, finite element level. The method can therefore allow for creation of local plastic hinge lines, which is crucial in case of optimization of thin-walled components. The method has been successfully applied to optimization of crash components under axial and oblique impact (Duddeck et al. (2016)) as well as transverse bending (Zeng and Duddeck (2017); Zeng (2018)).

All in all, HCA methods, thanks to their low computational costs³ as well as the ability to use explicit crash simulations directly, are very valuable in engineering practice. However, one has to keep in mind that none of the HCA approaches is a formal optimization method, and the optimality of the structures obtained with these methods cannot be guaranteed. In particular, the assumption that energy absorption should be homogenized either on the element level, as in the standard HCA, or on the macro-cell level, as in HCATWS, is questionable and can be true only in specific cases. Therefore, the optimization of crash-relevant objective functions cannot be tackled directly and one can only hope that the heuristic optimization criteria lead to improved designs for the problem at hand. Finally, HCA methods have difficulties to address problems with more than one constraint (Zeng (2018)), which is a severe limitation in industrial settings, where the problems are usually highly constrained. One possible solution of that problem is to look at the optimization task from the perspective of design exploration (Matejka et al. (2018)), where HCA can provide many different design concepts, based on the changes of boundary conditions, constraint values, or weights of individual load cases in multi-objective optimization scenarios (Aulig et al. (2018); Ramnath et al. (2019)). Alternatively, one can consider the above mentioned criteria for design variation as global design variables for an optimization problem, where HCA-optimized structures play the

³ Usually, 20–70 nonlinear FE simulations are needed for the HCA/HCATWS optimization to converge (Patel (2007); Mozumder et al. (2012); Zeng and Duddeck (2017); Raeisi et al. (2019)).

role of the design representation (Roux (2016)). As a result, the user can choose between different design concepts, based on various design criteria, which have not been taken into account in the optimization process itself. Anyway, the price to pay is an increased computational cost of such an approach, whereas the optimality of the structures w.r.t. the considered selection criteria is not guaranteed.

2.2.5. Bi-directional evolutionary structural optimization (BESO) approaches

Evolutionary Structural Optimization (ESO) is a TO approach proposed first by Xie and Steven (1997). In contrast to the methods proposed in this work, ESO does not use EAs, but is based on a simple heuristic rule for "evolving" the design, which assumes that inefficient material has to be gradually removed from the structure. The efficiency of the material can be measured using e.g. equivalent (von Mises) stress in the finite elements. If the ratio of the von Mises stress in a given element to the maximum von Mises stress in the entire structure drops below a certain threshold, referred to as the rejection ratio (Xie and Steven (1997); Huang and Xie (2010)), the element is deleted from the finite element model.

An extension of ESO, Bi-directional Evolutionary Structural Optimization (BESO), allows both for deletion as well as addition of material to the structure. The method was originally developed for optimization for statics (Yang et al. (1999)), but later on has been also applied to optimization of energy-absorbing structures (Huang et al. (2007)). More recently, the method has been further improved by combining BESO with Entropy Tabu Search Simulated Annealing (Christensen (2015)) and successfully applied to large-scale crash TO problems.

To summarize, since BESO methods do not use EAs, but, like HCA techniques, rely on heuristic optimization rules, they are applicable only to selected optimization problems. The advantage of those approaches is the ability to work with explicit crash simulations as well as their relatively low computational cost.

2.2.6. State-based representation (SBR) approaches

The last group of methods discussed here proposes a modification on the level of design representation. In all of the cases discussed above, the optimization relies on the changes of design variables parametrizing individual unit parts composing the structure, usually arranged in a regular grid. This results in a large number of design variables, which prohibits the use of non-gradient optimizations techniques, such as EAs or Efficient Global Optimization (EGO) (Jones et al. (1998); Forrester et al. (2008)).

SBR approaches propose to cluster the elements of the design according to their structural state, e.g. stress, internal energy density (Aulig and Olhofer (2016b)), or material density after initial topology/topometry optimization (Liu et al. (2017)). The clusters represent therefore finite elements exhibiting similar structural behavior and can be used as a low-dimensional representation

of the design. In such a case, a single density or thickness design variable can be assigned to the entire cluster of finite elements and updated according to non-gradient optimization methods, which address the optimization objectives and constraints directly. Alternatively, evolutionary-optimized update signals that emulate the sensitivity of the design variables can be used, as well (Aulig and Olhofer (2016b); Aulig (2017)). Although this approach is very promising, and has been successfully applied to an intrusion minimization problem in a crash scenario (Aulig and Olhofer (2016b); Aulig (2017)), it requires expensive re-training and more testing would be required to confirm the usefulness of this method. Recently, SBRs have been also used for optimization of an S-rail in an axial impact scenario (Liu et al. (2017)), as well as multi-material TO (Liu (2018)).

All in all, the SBR methods constitute a growing field of research and small-scale applications show their high potential. They can optimize crash structures of different types, including highly important thin-walled structures, and rely on high-fidelity explicit crash simulations. By reducing the dimensionality of the optimization problem, they allow for the use of such techniques as EAs or EGO, which target the relevant crashworthiness objectives and constraints directly, instead of using heuristic assumptions as in the state-of-the-art methods discussed previously. However, it is not clear how good is the representation based on clusters constructed according to the structural state of the design, especially for EAs or EGO. The high importance of representations used for EAs has been emphasized in many works in the EC community (Rothlauf (2006); Richter et al. (2015, 2016, 2018)). Obviously, the quality of the obtained designs would depend on the ability of a parametrization to represent superior solutions. At the same time, the representation determines the difficulty of the optimization problem for the evolutionary optimizer. Most probably, using different features for clustering would result also in considerably different design representations, so it is not clear which one should be used. Since the structural state of the design changes during the optimization, the clustering should be repeated over and over again, thus changing the representation of the design. For non-gradient optimization approaches, such as EAs or EGO, this means that the collected information about the landscape of the optimization problem is lost, which is a serious disadvantage, increasing the computational costs considerably. On the other hand, when the representation is kept unchanged during the entire optimization, non-gradient optimization techniques can be used more efficiently, but the representation could considerably limit the number of attainable designs, resulting in a sub-optimal solution. Finally, compared to the methods described before, the computational costs of EAs or EGO are considerably higher, even for a relatively low number of clusters, which might limit the applicability of the SBR methods in the industrial setting.

2.3. Evolutionary computation in structural topology optimization

In general, the structural optimization community has been very critical about using non-gradient methods, in particular EAs, in TO (Sigmund (2011)). To great extent, this criticism arises from the

assumption that standard grid representations (as for density-based methods) are used, which was the case for the early approaches (Sandgren (1990); Chapman et al. (1994)). The high number of design variables resulting from such a parametrization made the evolutionary-based optimization highly inefficient. As a result, the following criteria have been formulated as prerequisites for developing non-gradient TO methods (Sigmund (2011)), where at least one should be fulfilled:

- Discretization – non-gradient methods should be able to deal with models with at least 1000 finite elements; they should perform better than or at least as good as gradient-based methods based on numerical sensitivities; the optimized topologies should be free from artifacts such as checkerboard patterns or one-node-connected elements.
- Problem type – non-gradient methods should deal with optimization problems not solvable with standard gradient-based methods.

From this perspective, development of non-gradient methods for crashworthiness TO, which is the goal of this work, is justified already by satisfying the second condition.

The key limitation of EAs, being strong dependency of their performance on the number of design variables, motivated a development of alternative representations for non-gradient TO. A comprehensive overview of different representations for evolutionary-based TO has been given by Aulig and Olhofer (2016a). Apart from already mentioned grid representations (Sandgren (1990); Chapman et al. (1994)), graph representations (Ahmed et al. (2013)), level-set representations based on RBFs (de Ruiter and van Keulen (2000)) and Kriging (Hamza et al. (2013)), indirect representations (Steiner et al. (2008); Pedro and Kobayashi (2011)), and others (Aulig and Olhofer (2016a)), have been investigated for evolutionary-based TO. Still, the limited quality of the results as well as high computational costs, reaching millions of finite element simulations, made prohibitive practical applications of the most of the evolutionary-based TO methods in areas different from linear elasticity.

2.4. Machine learning in structural topology optimization

Recent advances in the domain of ML (Bishop (2007)), and especially Deep Learning (DL) (LeCun et al. (2015)), inspired structural optimization community to explore the potential of using those methods in the field of TO. ML methods can be integrated into TO in many different ways. Based on the methods from the literature, one could distinguish the following use cases of ML in TO:

- Optimal design prediction.
- Representation learning.
- Sensitivity learning.
- Surrogate-assisted TO.
- Learning of design rules.

The two last categories, to the best of our knowledge, have not been addressed in the literature before, and are novel fields of research, introduced in this thesis and our previous publications. As such, they will be not addressed in this section. Below, the existing contributions belonging to the first three groups are discussed.

2.4.1. Optimal design prediction

Optimal design prediction is probably one of the most straightforward strategies of using ML in TO. Assuming that the standard TO methods can be used for generation of optimal designs for a given optimization problem, one can create a dataset of optimal topologies for different boundary conditions and constraints, e.g. different volume fractions. Provided that the number of such samples is large enough, ML or even DL methods can be used for prediction of optimal designs based exclusively on the input information about loads, supports, and the volume fraction.

For the first time, this approach was proposed by Ulu et al. (2014), where 1000 optimal topologies for a compliance minimization problem were collected. Different topologies were obtained by varying randomly the position and components of a single force placed in the design space. To reduce the dimensionality of the problem, Ulu et al. (2014) use Principal Component Analysis (PCA) on the element densities. As a result, the dimensionality of the design representation is reduced from 80×40 elements to 80 PCA weights. A neural network is used to predict the PCA weights based on a vector of four values specifying the x - and y -position of the force and its components. A linear combination of the principal components multiplied by the predicted weights yields the prediction of the density field of the optimal design.

A similar approach was proposed by Lei et al. (2018), where a representation based on MMCs was used. Different samples are generated here by varying the position of a single load in the vertical direction only. In spite of the fact that the number of design variables in this case is much lower than for density-based representations, the authors use PCA to further reduce the dimensionality. Finally, Support Vector Regression (SVR) and k -Nearest Neighbors (k -NN) are used as the models for prediction of the PCA weights.

More recently, Yu et al. (2018) proposed to use Convolutional Neural Networks (CNNs) directly to predict the optimal density distribution. As in the previous cases, the supports are kept always the same and only the position and orientation of a single concentrated force, as well as the volume fraction, are varied to generate 100,000 optimal designs, being the training samples for a CNN. Finally, a conditional Generative Adversarial Network (GAN) is used to upscale the predicted design to a high resolution.

The latest approaches (Sosnovik and Oseledets (2017); Banga et al. (2018); Zhang et al. (2019)) propose to overcome the key limitation on the generalization of the methods described above, where the optimal topology is predicted based on the input specifying the boundary conditions explicitly. Those approaches assume that the optimal topology can be predicted either based on

the material distribution after the initial iterations of TO (Sosnovik and Oseledets (2017); Banga et al. (2018)), or exclusively based on the initial distribution of structural state fields, such as displacements or strains (Zhang et al. (2019)). The assumption made by Zhang et al. (2019) is therefore very similar to the one used in the approaches proposed in our previous works (Krischer (2018); Bujny et al. (2018)) as well as this thesis, where the sensitivities or topological variations are predicted based on the structural state of the design.

To summarize, the ML-based methods for predicting optimal designs constitute a growing field of research. Most of the approaches proposed so far, however, are limited only to very specific cases, allowing for prediction of the optimal solution only for a single load and constant supporting conditions. On the other hand, potentially more general methods (Zhang et al. (2019)) require hundreds of thousands of samples even for the cases when the supports remain the same. Therefore, to make the model truly generalizable for more arbitrary boundary conditions, the effort for generating the training samples would be extremely high.

2.4.2. Representation learning

ML techniques can be used also more indirectly in TO, by identifying favorable representations, which could result in a reduced number of design variables. From this perspective, the representation learning techniques address a similar question as the SBR approaches discussed in Section 2.2, or even feature-mapping methods (Wein et al. (2019)), used in this work.

In one of the first approaches, Guo et al. (2018) used TO to generate 15,000 optimal topologies for a heat conduction problem. In the next step, topologies were used to train an augmented Variational Autoencoder (VAE), providing a low-dimensional representation of the design. Finally, the designs were optimized for a new set of problems using GAs and the learned latent representation. A similar idea was employed later on in structural TO (Oh et al. (2019)) by using GANs.

Although the representation learning approaches look very promising, they also require a lot of training data, which has to be generated via costly TO. Once the low-dimensional representation is learned, it has to be used again in a costly non-gradient optimization, thus reducing the effort only partially. Finally, it is not clear how good are the learned representations if the optimization problem changes strongly. Compared to the feature-mapping methods (Wein et al. (2019)), like the MMC techniques (Guo (2014)), DL-based representations are also considerably harder to understand and interpret.

2.4.3. Sensitivity learning

Another technique relying on the indirect use of ML methods, is the TO by Predicting Sensitivities (TOPS), proposed first by Aulig and Olhofer (2014b). In this approach, based on numerical estimation of the gradients of objectives and constraints, ML models of sensitivities are constructed and used for gradient-based TO. The models rely on input features related to the structural state of the design, extracted from the finite element simulations. Later on, Aulig and Olhofer

(2014a), proposed a modification of this approach by learning the update signal models using neuro-evolution.

TOPS is a very promising approach, which can be used for learning sensitivity models in case if analytical expressions are not available. Compared to the methods discussed before, ML techniques can be used here based on much lower number of training samples, which are relatively cheap to obtain. Furthermore, the obtained models can possibly generalize much better to the new cases. The main problem remains the correctness of the models, which required frequent re-training (Aulig (2017)). Therefore, relying fully on the correctness of ML models and using them directly in gradient-based optimization methods might lead to sub-optimal solutions or even to a complete divergence of the optimization.

2.5. Aim and objectives

Given very high complexity of vehicle structures produced today, rising material costs, stringent CO₂ emission targets, as well as the pressure from the market to shorten development cycles, the use of numerical support in otherwise very time-consuming process of identifying optimal crashworthy structures becomes very important. However, the state-of-the-art crashworthiness TO methods available today suffer from very limited generality, due to the strong simplifications they make or heuristic assumptions they use. As a result, they are applicable only to very specific cases and only some optimization criteria can be taken into account.

On the other hand, non-gradient optimization methods, such as EAs or surrogate-assisted methods, can target optimization for arbitrary quantifiable criteria. EAs can deal with problems involving high levels of numerical noise, bifurcations, and discontinuities. The main challenge remains the high dimensionality of TO problems, which makes most of the previously proposed evolutionary-based TO approaches not applicable to crashworthiness.

LSMs, in particular approaches using geometric shapes as local basis functions, accommodate two features crucial for the crashworthiness TO, being 0-1 material distribution and the possibility to significantly reduce the number of design variables. The first property is very important for a proper simulation of crash structures and manufacturability of the obtained topologies. The elements occupying the non-material region have to be deleted from the mesh, since even small regions with very low Young's modulus can completely distort the crash behavior of the structure due to their plastic incompressibility, often assumed in the material modeling. The second property is of very high importance when evolutionary or surrogate-assisted methods are used, since their performance heavily depends on the number of design variables.

Finally, ML techniques offer a great potential to reduce the computational costs of TO. However, most of the approaches available today are not fully reliable, or need tremendous learning effort to work only on a small number of cases. Therefore, in this work, different ML techniques are

considered mainly as enhancements of expensive evolutionary-based search, allowing for a better utilization of the data collected during the optimization or integration of the knowledge learned in a prior offline process.

As a result, the aim of this work is to evaluate the feasibility of using approaches from the domain of AI, such as EAs and ML, together with a low-dimensional level-set representation, for optimization of crash structures.

In particular, this can be decomposed into the following objectives:

- Development of an approach for optimization of crash structures using EAs and a suitable low-dimensional representation of 2D and 3D designs.
- Reduction of computational costs via hybrid approaches utilizing approximate gradient information, obtained with use of physical surrogates and supervised learning techniques.
- Evaluation of the potential of using surrogate modeling techniques for TO of low-dimensional problems.
- Development and evaluation of an adaptive evolutionary approach utilizing learning-based topology variations.
- Evaluation of applicability of the evolutionary-based methods to optimization of real-world crash structures.

From the perspective of the optimization process, the proposed ML enhancements realize the following concepts to improve the convergence speed and reliability of the evolutionary search:

- Estimation of sensitivity information for the hybrid optimization methods.
- Prediction of objectives and constraints like in surrogate-based optimization.
- Biasing the mutation operator towards preferred designs.

Figure 2.1 illustrates the core concepts used in this thesis. The main contribution of this work is the EA-LSM for crashworthiness optimization. The method, though, has a generic character and can be potentially also used in different application areas of structural TO. To facilitate an optimization with an adaptive representation, Adaptive Evolutionary Level Set Method (A-EA-LSM), is proposed. The key difference between EA-LSM and A-EA-LSM is the way in which topological variations take place. Since EA-LSM relies exclusively on a level-set representation and evolutionary operators such as mutation and recombination, the optimizer is blind to the modifications of topology, which happen on the phenotype level as a result of modifications of the genotype according to the evolutionary process. In contrast, in A-EA-LSM, the topology is explicitly encoded in the genotype in a form of graph, and topological modifications are introduced using specialized topology variation operators. Finally, to reduce the computational costs of the

purely evolutionary-based techniques, methods from the domain of ML, such as sensitivity learning, prediction of topology variations (directly related to the concept of learning of design rules), as well as surrogate modeling techniques, are used.

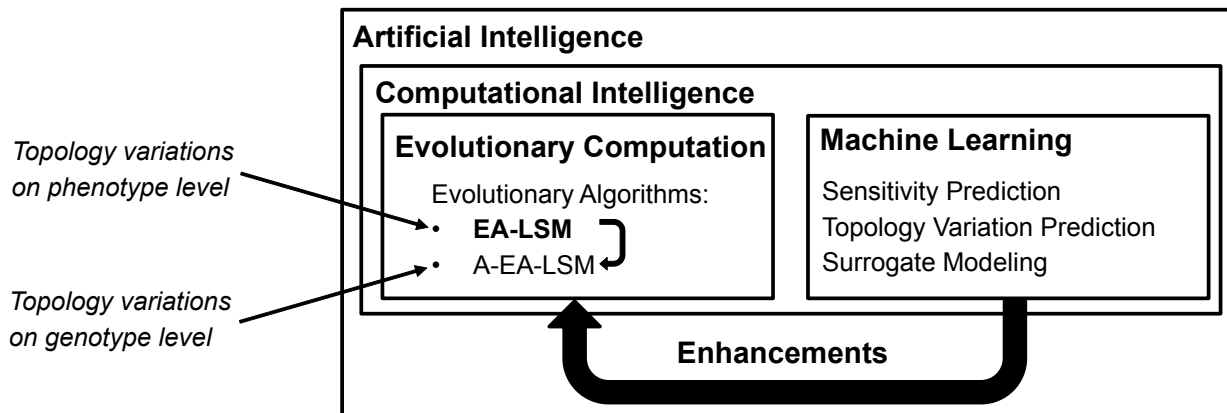


Figure 2.1 Graphical overview of the core concepts used in the thesis.

Chapter 3

Fundamentals of evolutionary computation

This chapter briefly introduces methods from the field of EC, necessary for understanding the TO approaches proposed in this work. In particular, in the following sections, Evolution Strategies (ESs), which are suitable for solving continuous optimization problems, are discussed. These methods, with some problem-specific modifications, discussed in Chapter 5, are the core of the approaches for solving crash TO problems used in this thesis.

3.1. Evolutionary computation

EC originated in 1960s and resembles models based on the principles of Darwinian evolution (Darwin (1859)). In spite of the variety of different types of methods, e.g. Genetic Algorithms (GAs), Evolution Strategies (ESs), Evolutionary Programming (EP), most of them are based on the following concepts:

- *Genotype* is an object of the search space defined by the representation. In the context of engineering optimization, genotype would correspond to the vector of design variables parametrizing the design.
- *Phenotype* is an object of the solution space, determined by a genotype-phenotype mapping. For numerical optimization, phenotype would be equivalent to the finite element representation of geometry of the design generated for a given vector of design variables (genotype).
- *Individual* is typically defined by a genotype and additional parameters encapsulating its knowledge about the fitness landscape, e.g. step sizes in ESs.

- *Fitness* of the individual is determined by the behavior of the phenotype (e.g. finite element mesh) in the environment (e.g. crash load cases). In case of computer-based optimization, the fitness of an individual would be measured based on the quantities calculated using a numerical simulation, relevant for the considered optimization criteria, e.g. compliance or intrusion of an impactor. The *fitness function* maps the behavior of the individual to a fitness value.
- *Population* is a group of individuals competing with each other based on their fitness.
- *Recombination* produces new offspring individuals by combining the genotypes (and potentially also strategy parameters, e.g. step sizes in ESs) of the individuals from the current parent population.
- *Mutation* introduces small random variations of the genotype. In case of engineering optimization problems, the mutation operator would modify the vector of design variables parametrizing the design.
- *Selection* chooses the offspring individuals composing the parent population in the next generation according to the fitness values.

A typical EC optimization starts with an initialization of the population of individuals, whose fitness values are computed in the evaluation step. Subsequently, an evolutionary process, based on recombination, mutation, and selection, starts. Selection operator realizes the famous concept of "survival of the fittest" by eliminating the individuals of low fitness values from the population. Recombination operator combines the genotypes of multiple (parent) individuals to produce new (offspring) individuals. By combining the genotypes of the fittest individuals obtained after the selection step, one can explore different solutions, hoping that further improvement can be obtained. Finally, the mutation operator introduces random variations of the genotype in order to find better solutions in the neighborhood of the existing individuals. The process is continued until certain convergence criteria are satisfied.

3.2. Evolution strategies (ESs)

The following section gives the formal definitions of the evolutionary operators and the ES itself, used later on to describe the methods proposed in this thesis. The notations used here follow the conventions introduced by Bäck and Schwefel (1993).

ESs were first proposed by Ingo Rechenberg and Hans-Paul Schwefel in Germany in 1970s (Rechenberg (1971); Schwefel (1977)). In contrast to GAs, ESs focus on mutation rather than recombination as the main source of genotypic variations. A typical ES can be summarized in the form of a following pseudocode:


```

iter := 0;
initialize  $\mathcal{P}(0) := \{\mathbf{a}_1(0), \dots, \mathbf{a}_\mu(0)\} \in \mathbb{I}^\mu$ ;
evaluate  $\mathcal{P}(0) : \{f(\mathbf{a}_1(0)), \dots, f(\mathbf{a}_\mu(0))\}$ ;
while ( $\iota(\mathcal{P}(\textit{iter})) \neq \textit{true}$ ) do
    recombine:  $\mathcal{P}'(\textit{iter}) := r_{\Theta_r}(\mathcal{P}(\textit{iter}))$ ;
    mutate:  $\mathcal{P}''(\textit{iter}) := m_{\Theta_m}(\mathcal{P}'(\textit{iter}))$ ;
    evaluate  $\mathcal{P}''(\textit{iter}) : \{f(\mathbf{a}_1''(\textit{iter})), \dots, f(\mathbf{a}_\lambda''(\textit{iter}))\}$ ;
    select:  $\mathcal{P}(\textit{iter} + 1) := s_{\Theta_s}(\mathcal{P}''(\textit{iter}))$ ;
    iter := iter + 1;
end

```

Algorithm 1: Standard Evolution Strategy (Bäck and Schwefel (1993)).

with $f : \mathbb{I} \rightarrow \mathbb{R}$ being the cost (fitness) function to be minimized (maximized). \mathbb{I} denotes the space of individuals, and $\mathbf{a} \in \mathbb{I}$ is an individual. The size of the parent population is denoted by $\mu \geq 1$, while $\lambda \geq \mu$ is the size of the offspring population.

A population of parents at generation *iter* is expressed as $\mathcal{P}(\textit{iter}) = \{\mathbf{a}_1(\textit{iter}), \dots, \mathbf{a}_\mu(\textit{iter})\}$. The recombination operator can be defined as a mapping $r_{\Theta_r} : \mathbb{I}^\mu \rightarrow \mathbb{I}^\lambda$, which transforms the space of parent individuals \mathbb{I}^μ to a space of offspring individuals \mathbb{I}^λ . A mutation operator of a form $m_{\Theta_m} : \mathbb{I}^\lambda \rightarrow \mathbb{I}^\lambda$, completes the reproduction stage. The recombination as well as the mutation are controlled with use of operator parameter sets Θ_r and Θ_m , respectively. Finally, the selection operator is defined as $s_{\Theta_s} : \mathbb{I}^\lambda \rightarrow \mathbb{I}^\mu$, and selects the fittest individuals to construct the population of parents in the next generation. Similarly to the recombination and mutation operators, it is controlled by a set of parameters Θ_s . The termination criterion for the algorithm is denoted by $\iota : \mathbb{I}^\mu \rightarrow \{\textit{true}, \textit{false}\}$.

Most frequently, recombination is applied both to design variables as well as strategy parameters of the mutation operator. Several different types of recombination were proposed in the past and can be summarized as follows (Bäck and Schwefel (1993)):

$$z'_i = \begin{cases} z_{S,i} \text{ without recombination,} \\ z_{S,i} \text{ or } z_{T,i} \text{ discrete recombination,} \\ z_{S,i} + \chi \cdot (z_{T,i} - z_{S,i}) \text{ intermediate recombination,} \\ z_{S_i,i} \text{ or } z_{T_i,i} \text{ global, discrete recombination,} \\ z_{S_i,i} + \chi_i \cdot (z_{T_i,i} - z_{S_i,i}) \text{ global, intermediate recombination,} \end{cases} \quad (3.1)$$

where z_i is the i^{th} component of the vector of design variables $\mathbf{z} \in \mathbb{R}^n$, where n is the dimensionality of the optimization problem. S and T stand for two randomly selected individuals, while $\chi \in [0, 1]$ is a random variable drawn from a uniform random distribution. For the global recombination, parent individuals S and T as well as the parameter χ are taken independently for each of the components of the vector of design variables.

Usually, the best results are obtained for discrete recombination on design variables and intermediate recombination on the strategy parameters (Bäck and Schwefel (1993)). Therefore, this approach was also employed in this work when using standard ESs. Historically, the most common form of intermediate and global recombination is using a constant value of $\chi = \frac{1}{2}$.

After applying the recombination operator to generate offspring individuals, they are modified by means of mutation. A version of mutation operator with individual step sizes varies first the step sizes σ_i and then the vector of design variables \mathbf{z} according to the following expressions:

$$\begin{aligned}\sigma'_i &= \sigma_i \cdot \exp(\tau' \cdot N(0, 1) + \tau \cdot N_i(0, 1)), \\ z'_i &= z_i + \sigma'_i \cdot N_i(0, 1),\end{aligned}\tag{3.2}$$

with τ' and τ being global and local learning rates, respectively. $N(0, 1)$ and $N_i(0, 1)$ stand for realizations of a random variable with a normal distribution with mean 0 and standard deviation 1. The notation $N_i(0, 1)$ indicates that the random variable is sampled independently for each value of the counter i .

In case of the simplest variant of ES, using a single step size σ for all components of the vector of design variables \mathbf{z} , Equation (3.2) can be simplified to:

$$\begin{aligned}\sigma' &= \sigma \cdot \exp(\tau_0 \cdot N(0, 1)), \\ z'_i &= z_i + \sigma' \cdot N_i(0, 1),\end{aligned}\tag{3.3}$$

where τ_0 is the learning rate.

Schwefel (1995) proposes to use the following values of the learning rates:

$$\begin{aligned}\tau &= \frac{1}{\sqrt{2\sqrt{n}}}, \\ \tau' &= \frac{1}{\sqrt{2n}}, \\ \tau_0 &= \frac{1}{\sqrt{n}}.\end{aligned}\tag{3.4}$$

The mutation mechanism in the form presented above allows for evolution of the strategy parameters during the optimization process by the algorithm itself. This is frequently referred to as the "self-adaptation" mechanism, first formulated by Schwefel (1987).

Finally, the evaluation step is followed by selection, where the best individuals are selected to create a new population of parent individuals. Among different types of selection operators (Bäck (1996)), the most popular are (μ, λ) -selection and $(\mu + \lambda)$ -selection. In the first variant, the new

population of parent individuals is formed exclusively out of the best individuals from the offspring population, which is frequently referred to as the non-elitist selection. In the second variant, the best individuals out of a combined population of parents and offspring are selected, which is commonly known as elitist selection. Please note that both variants have deterministic character, in contrast to the stochastic (proportional) selection operator used typically in GAs (Bäck (1996)). In this work, (μ, λ) -selection is used due to its better performance on multi-modal and noisy problems (Bäck and Schwefel (1993)).

3.3. Covariance matrix adaptation evolution strategy (CMA-ES)

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) (Hansen and Kern (2004)) is a derandomized ES, which adapts the covariance matrix of the normal distribution used in the mutation operator according to the previous search steps (Hansen and Ostermeier (2001)). This concept is very similar to the idea used in the gradient-based quasi-Newton methods, which estimate the Hessian matrix in an iterative fashion during the optimization process. Instead, CMA-ES adapts only the parameters of the normal distribution, which define the distribution of the offspring. CMA-ES was initially designed for dealing with small population sizes and has demonstrated its robustness and usefulness as an efficient local search method (Hansen and Ostermeier (1996)).

In particular, CMA-ES has shown to be very efficient in minimizing unimodal functions (Hansen and Ostermeier (2001)). The method turned out to be also superior on ill-conditioned and non-separable optimization problems as well as in real-world applications (Hansen and Kern (2004)). CMA-ES was extended later on by introducing the rank- μ -update (Müller et al. (2002); Hansen et al. (2003)), which allowed for improving its performance for large population sizes, as well. Finally, the Python implementation of CMA-ES (Hansen (2016)) has been made available and was used in this work.

3.4. Summary

In this chapter, a necessary background on EC was given. In particular, standard ESs and the state-of-the-art CMA-ES, were briefly discussed. These methods, with problem-specific modifications, are the basis for the approaches proposed in Chapters 5 and 6. Finally, the usefulness of these methods for solving optimization problems in TO is demonstrated in Chapters 7, 8, and 9.

Chapter 4

Representations for evolutionary-based structural topology optimization

Evolutionary optimization methods offer a great potential to address a wide spectrum of different problems. However, the efficiency of any evolutionary-based optimization approach is strongly dependent on the used representation and its relation to the evolutionary operators. Arciszewski et al. (1995) define representation as a computational description of an engineering system expressed in terms of attributes. Depending on the representation, not only the dimensionality of the optimization problem can be changed, but also the difficulty of the optimization task itself. Therefore, finding representations that transform the landscape of the fitness function in a way that it is easier to optimize by means of evolutionary operators such as mutation or recombination is a broad research field alone (Rothlauf (2006)). Additionally, aspects such as evolvability of representations (Menzel (2011); Lehmann and Menzel (2012); Richter et al. (2015, 2016, 2018)) or structural attainability (Guirguis et al. (2015)), aiming to find parametrizations that allow for reaching as many meaningful designs as possible with a minimal number of design variables, are of high importance. As a result, the choice of a particular design representation, especially for high-dimensional TO tasks, is a fundamental problem, which determines the success of the entire approach. Therefore, this chapter provides an intuitive justification of the choice of the design representation used in this work.

First of all, in Section 4.1, the taxonomy of design representations used in non-gradient TO is briefly discussed. Subsequently, based on simple experiments, advantages and disadvantages of different low-dimensional representations of topology are discussed. In particular, Section 4.2 dis-

cusses the influence of different types of representations on the properties of the resulting fitness landscape. The discussion is continued in Section 4.3, by focusing on the correlation of parameters resulting from some types of representations as well as additional important aspects. Finally, Section 4.4 concludes the chapter and presents prerequisites for development of good representations for evolutionary-based TO problems.

4.1. Taxonomy

Aulig and Olhofer (2016a) gave a comprehensive overview of different representations used in non-gradient TO. They distinguished three main categories, being: grid, geometric, and indirect representations (Figure 4.1).

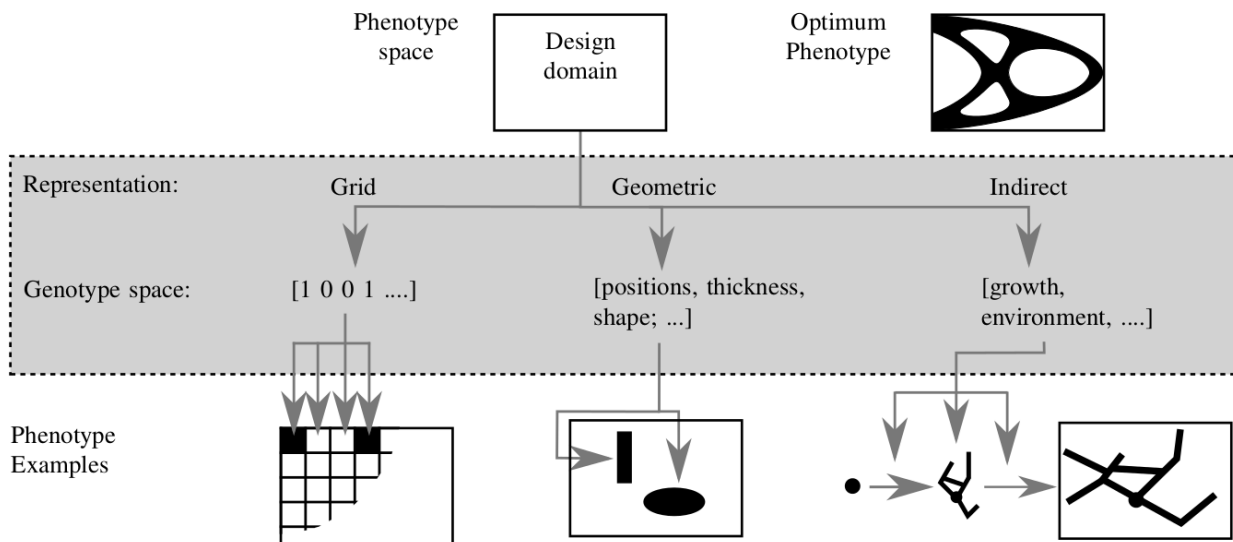


Figure 4.1 Taxonomy of representations for the black-box TO according to Aulig (2017).

The first approaches in the black-box continuum TO were based on the grid representation (Sandgren (1990); Chapman et al. (1994)). In a typical grid representation, the design domain is divided into a fixed grid of cells, whose densities can take values in a continuous range between 0 (no material) and 1 (full material), or, alternatively, one of two values – 0 or 1. As a consequence, the approaches of the second type are referred also to as bit-array encodings (Aulig and Olhofer (2016a)). Such representations result typically in thousands up to millions of design variables, which usually correspond directly to the number of finite elements in the mesh, and offer the highest flexibility to represent different designs. This leads to a tremendous number of possible combinations, requiring millions of evaluations to optimize a structure.

The second group encapsulates different types of implicit representations, which can be some kind of movable shape primitives or parametrized functions defining the location of material. The phenotype in these cases is obtained by a mapping of the lower-dimensional representation on a fixed grid of cells. The main advantage of this type of representations is a possible reduction of

the dimensionality of the original optimization problem, which is no longer correlated with the resolution of the finite element mesh.

The third group, with recent works by Aulig (2017); Liu et al. (2017); Ulu et al. (2014); Guo et al. (2018), consists of indirect representations, which parametrize the design implicitly, e.g. based on the structural state of the initial design in a crash simulation. The huge diversity of methods within this group, as well as their indirect character, makes a direct comparison very difficult. Due to their complexity and the variety of ongoing research, those methods will be not considered in this chapter.

4.2. Properties of the fitness landscape

In order to understand better the properties of different representations and their influence on the fitness landscape, let us consider a simple, 2D optimization problem as shown in Figure 4.2. A cantilever beam is fixed at the left-hand side and loaded on the right-hand side with a static point force \vec{F} . The goal is to minimize the compliance (maximize stiffness) of the structure. There are no constraints applied.

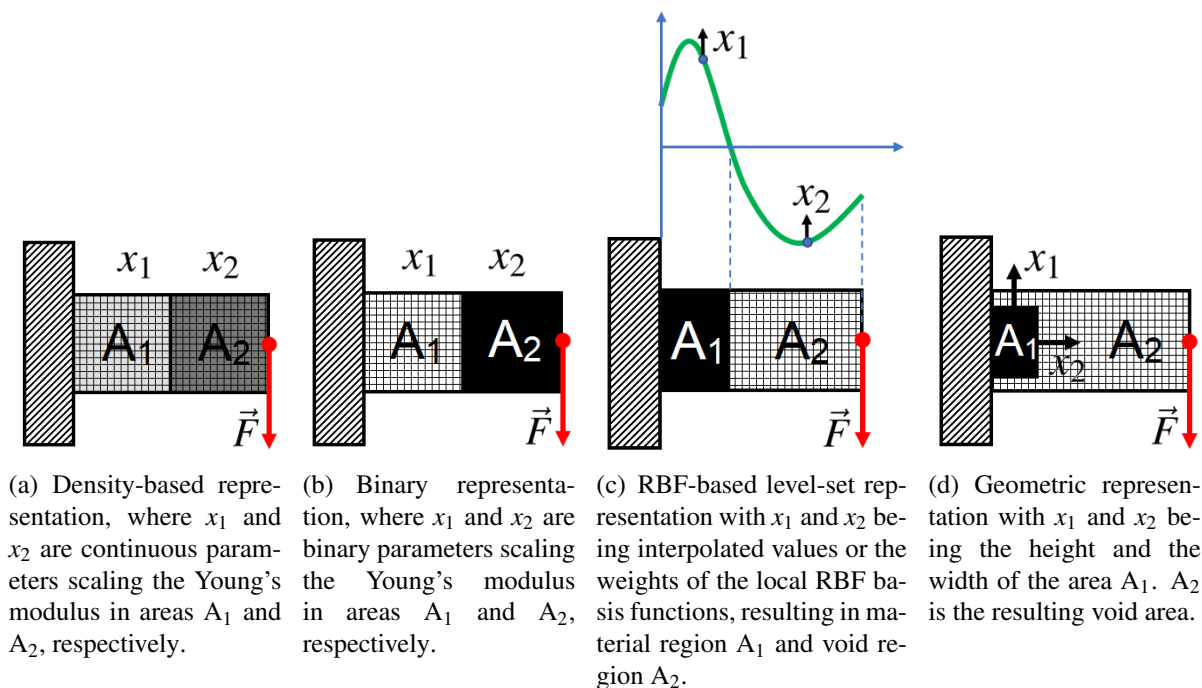


Figure 4.2 Four representation types used for the investigations on a simplified 2D cantilever beam problem.

Figure 4.2(a) shows how the design can be parametrized using a density-based grid representation. The design variables x_1 and x_2 define the density of the material in two regions of the design domain, A_1 and A_2 , respectively. More specifically, the density $x \in [0, 1]$ is used to scale the Young's modulus of the material $E(x) = E_0 \cdot \max(x, x_{min})$, where E_0 is the reference Young's modulus value

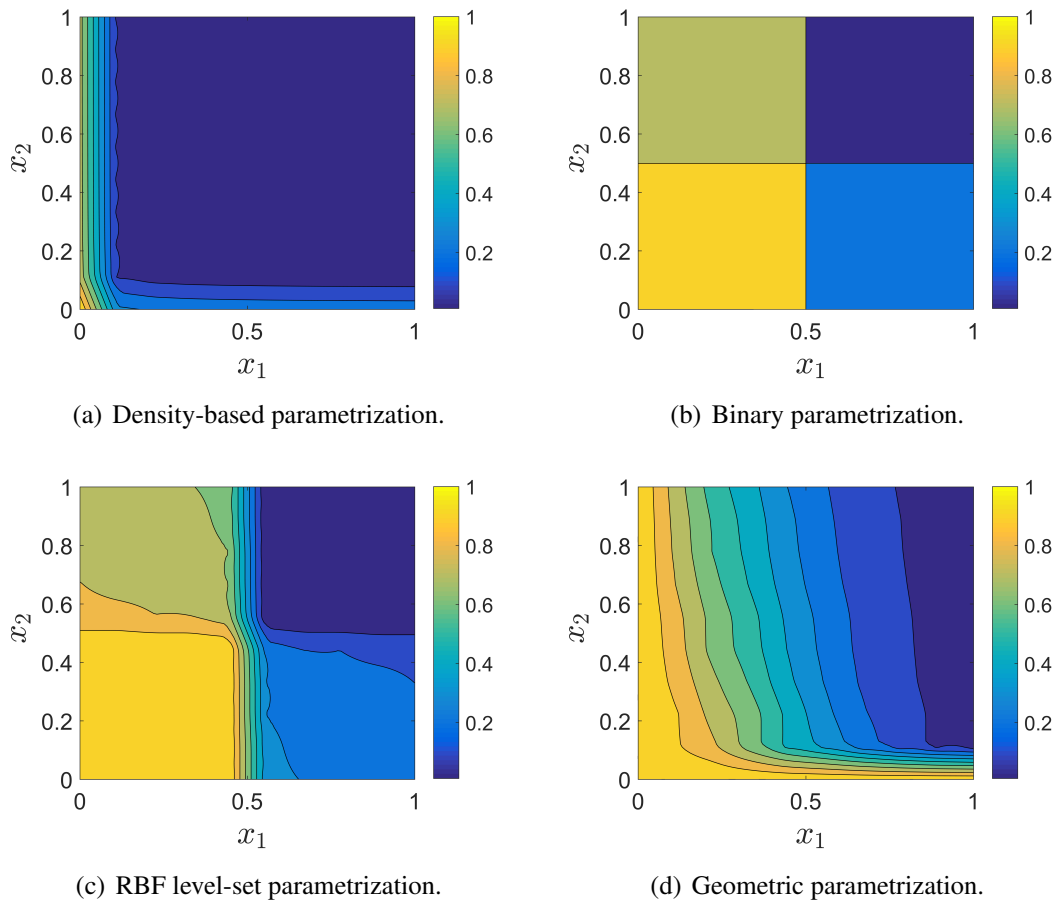


Figure 4.3 Contour plots of compliance in dependency of design variables x_1 and x_2 , for parametrizations defined in Figure 4.2.

and x_{min} is a small value, e.g. 0.01, introduced to avoid problems with numerical instabilities in the finite element simulation. In such a case, the contour lines of the objective function being minimized (compliance) look as shown in Figure 4.3(a). From the perspective of evolutionary optimization, even this simple optimization problem would pose several difficulties. First of all, the objective function is strongly nonlinear, taking very high values for the designs with at least one cell having very low density. In higher-dimensional search spaces, this could potentially lead to multi-modality of the considered objective functions. Moreover, as the density of both regions increases, the objective function plateaus, requiring the algorithm to adapt constantly the step size to the changing optimization landscape. The second problem is a strong correlation between the design variables – the biggest improvement can be obtained only if both design variables change by the same amount, at the same time. Thus, this information should be used to modify the probability distribution used by the mutation operator. This requires an additional effort for learning the landscape of the optimization problem, which can be achieved only with more sophisticated EAs. In this case, utilization of simpler algorithms, e.g. an ES with a single step size, would result in a worse performance, especially for high-dimensional problems, which are inevitable in TO using this type of representation.

Despite the popularity of the density-based TO methods, in many applications, including crashworthiness, it is crucial to parametrize the design in such a way that it does not allow for intermediate densities. The simplest idea to obtain this is to use a binary representation, where every grid cell can be either occupied by full-density material or void, i.e. $x \in \{0, 1\}$. This representation is depicted in Figure 4.2(b). The contour plot for compliance minimization problem is shown in Figure 4.3(b). For the sake of illustration, the binary design variables were considered as continuous with value of 1 for $x_i \geq 0.5$, $i \in \{1, 2\}$, and 0 otherwise. This type of problems is usually solved with use of optimization algorithms well-suited for discrete optimization tasks, such as GAs. However, one should note that discrete optimization problems are usually more difficult to solve than their continuous counterparts.

A possible solution of the problems of the intermediate densities and the continuity of the parametrization comes with LSMs. The main idea in LSMs is to define the material distribution implicitly, with use of a level-set function (LSF). The regions occupied by material correspond to positive values of the LSF, while the regions occupied by void are indicated by negative LSF values. As a result, clear boundaries between material and void phases are obtained. A parametrization based on the LSM concept was proposed by Hamza et al. (2013), where the Kriging model (Forrester et al. (2008)) is used as a LSF. The model interpolates a set of control points, whose positions are treated as design variables. Similar ideas were explored also by other authors (de Ruiter and van Keulen (2000); Pehnek et al. (2016)). This approach can reduce significantly the dimensionality of the optimization problem compared to the grid representation, while being still able to represent a wide range of different designs. Figure 4.2(c) shows an example of a 2D level-set parametrization interpolated with Gaussian RBFs. Without loss of generality, we consider weights for a linear combination of RBFs as design variables instead of addressing the interpolation task as done by Hamza et al. (2013). Figure 4.3(c) shows that, essentially, the landscape of the objective function being optimized does not differ too much from the one for the binary representation (Figure 4.3(b)). Again, there are four large plateaus, corresponding to the situations as depicted in Figure 4.2(c), where, assuming that x_1 remains constant, even relatively large variations of the parameter x_2 would not correspond to any changes of the material distribution. For an optimizer, the plateaus are very difficult to deal with, since the gradient of the objective function in this area equals zero and all of the individuals existing on a plateau are equally good. As a result, the choices made by the selection operator are arbitrary and the search becomes a random walk (Pearson (1905)). On the other hand, as x_2 increases, a lot of variation in the material distribution would happen in a very small range of values of the x_2 parameter. More formally, this is related to the strong nonlinearity in the mapping of the design variables to the material distribution, which is demonstrated in Figure 4.4. The nonlinearity of the mapping, and consequently, even stronger nonlinearity of the objective function, is a difficult problem for the optimizer, which has to constantly adapt its parameters to the changing optimization landscape. The problem of the strong nonlinearity in case of the RBF representation could be probably partially solved by a careful choice of the lower and upper limits

for the design variables, so that the optimizer operates mainly in an approximately linear range, but it might be difficult in general case due to the influence of the neighboring basis functions on the LSF in the region of interest (Figure 4.5).

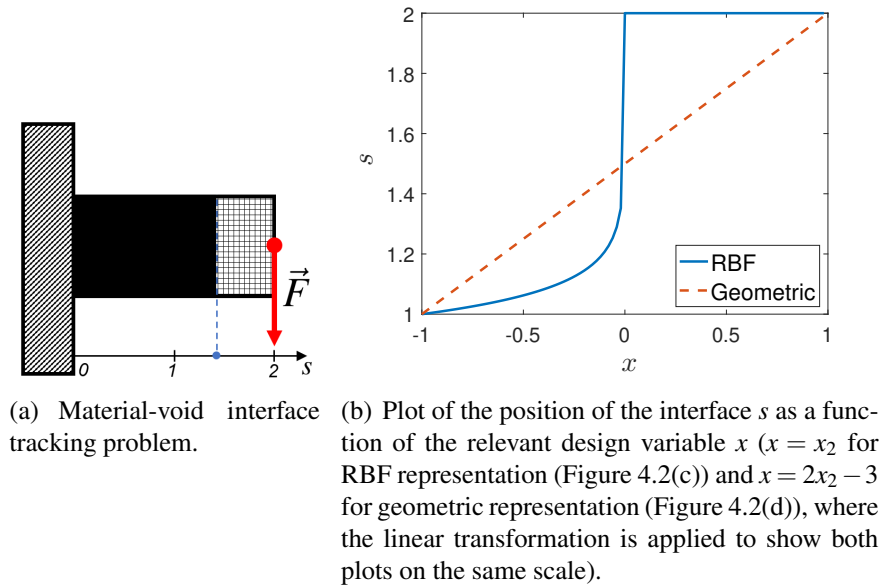


Figure 4.4 Problem of the strong nonlinearity in the mapping between design variables and the material distribution for the Gaussian RBF-based level-set parametrization. Given $x_1 = 1$ in the representation as in Figure 4.2(c), the position s of the interface between material and void (Figure 4.4(a)) would change as depicted in Figure 4.4(b) (solid blue line) with the changes of the second design variable x_2 . For comparison, in the geometric representation (Figure 4.2(d)), the position of the interface would change linearly with the variation of the design variable x_2 (dashed red line in Figure 4.4(b)).

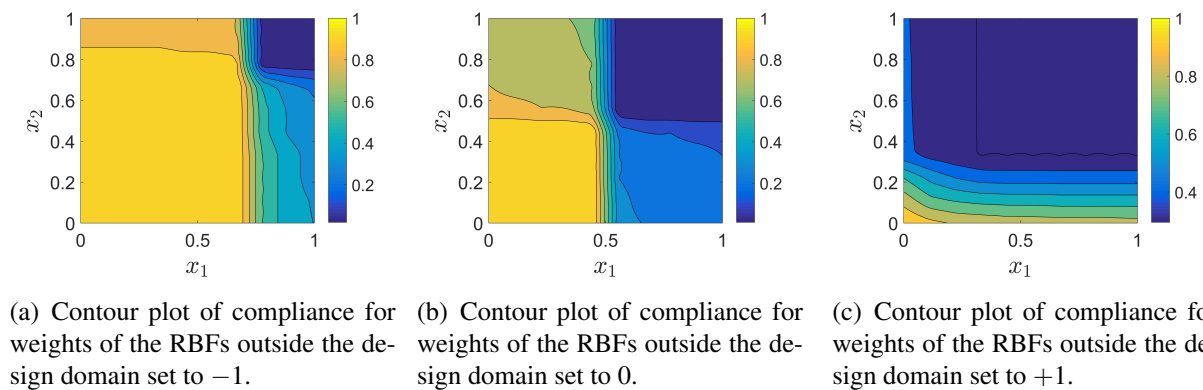


Figure 4.5 Variations of the contour plots of compliance for the 2D optimization problem depicted in Figure 4.2(c), depending on the weights of the neighboring RBFs, outside the design domain (fixed parameters).

4.3. Correlation of parameters and additional aspects

All of the representations discussed in the previous section are based on a concept of a localized parametrization, where the topology is defined by parameters having only a local influence on the material distribution. Moreover, each parameter is usually associated with a fixed region in the

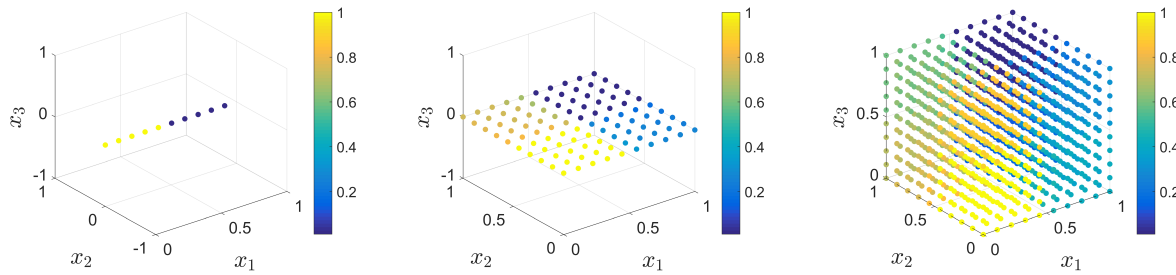
design domain. This results in several problems from the point of view of non-gradient optimization:

- Density-based, binary, and RBF-based representations require extensive handling of structurally infeasible designs. Since all possible solutions in a grid or a FE mesh can be represented, it is very likely to obtain disconnected structures, which might even not connect the load and the supports. Hence, even with a reduced dimensionality for RBF-based or similar representations, the size of the search space is very large, with plenty of structurally infeasible solutions.
- In grid representations, including density-based (Figure 4.2(a)) and binary (Figure 4.2(b)), typical discretization problems such as single node connected elements and checkerboard patterns occur. However, the filtering or projection methods used in gradient-based optimization (Bendsøe and Sigmund (2004)) are not suited for the black-box optimizers.
- The local character of the representation results in a strong coupling between design variables, which makes the optimization problem difficult.

Let us focus on the last point from the list above. As can be seen in Figure 4.3(c), only a correlated change of both x_1 and x_2 results in a significant improvement in the stiffness of the structure. This is logical, since if any of the two regions is not filled with material, the stiffness drops rapidly, with the void regions being modeled with the weak (ersatz) material (Sigmund and Maute (2013); Zhang et al. (2016)). A correlated change of both design variables is a difficult task for the evolutionary optimizer, since it has to carefully choose the parameters of the distribution used in the mutation operator and find a small region in the search space, corresponding to the topology that connects the load to the support. This problem becomes much more important in high-dimensional search spaces, which are essential for representing structures at a sufficient level of complexity. Figure 4.6 illustrates this problem based on a cantilever beam with 1, 2, and 3 grid cells (in case of grid representation) or control points (in case of RBF representation). As the number of grid cells or control points along the cantilever beam increases, so does the dimensionality of the optimization problem. As a result, the relative size of the region in the search space corresponding to a structure connecting the load to the support decreases exponentially fast, which is a special case of the curse of dimensionality, a problem arising in many fields of computer science.

Figure 4.7 shows an extension of the cantilever beam problem to an 18-dimensional search space. In order to connect the load to the support, a correlated change of 6 design variables is necessary, which is already a very challenging task for the optimizer and requires a very explorative search to find a small region in the search space corresponding to a structurally-feasible design. In practice, in order to be able to represent simple topologies with a sufficient detail level, even higher-dimensional cases have to be considered. Hamza et al. (2013) consider 2D TO problems with 46 and 52 design variables, which require up to 1 million FEM evaluations with use of a GA¹

¹ Although the inferiority of GAs with respect to the ESs for continuous optimization problems has been widely demonstrated (Bäck (1996)), it seems that for this representation type very explorative search is needed, justifying the use of GA.



(a) Compliance values for a 1D problem, with a 1 grid cell or control point along the cantilever beam. (b) Compliance values for a 2D problem, with 2 grid cells or control points along the cantilever beam. (c) Compliance values for a 3D problem, with 3 grid cells or control points along the cantilever beam.

Figure 4.6 Illustration of the relationship between the relative size of the region in the search space corresponding to a material distribution connecting the load to the supports and the dimensionality of the optimization problem in the grid (Figures 4.2(a) and 4.2(b)) or the RBF level-set (Figure 4.2(c)) representation. Nave blue points represent designs of very low compliance (the connection exists). The size of this region changes as $\mathcal{O}(v^n)$, where $v \in (0, 1)$ and n is the number of grid cells or control points.

to converge. This is a prohibitive computational cost in the context of expensive crash simulations, not to mention 3D crash TO problems².

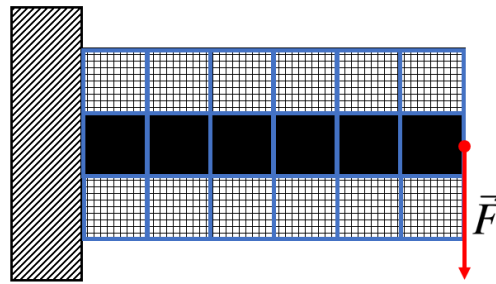


Figure 4.7 An 18-dimensional TO problem parametrized with a fixed grid of RBFs/cells.

In geometric representations, like the one depicted in Figure 4.2(d), the main problems described above – the nonlinearity of the material distribution mapping, strong correlation between the design variables, and the exponential decrease of the size of the search space region corresponding to connected designs – seem to be solved to a great extent. Firstly, as one can see in Figure 4.3(d), the contour lines for the compliance minimization problem are very uniformly spaced, with a dense concentration in the region corresponding to a very small height of the geometric component. This uniformity results partially from the linearity of the mapping between the material distribution and the design variables depicted in Figure 4.4(b) and is very convenient for the evolutionary optimizer, which does not have to adapt constantly the mutation step size to the changing optimization environment. Of course, in general case, the optimization problem itself might be highly nonlinear, but in this way, at least the nonlinearity coming from the geometry mapping can be avoided³.

² Assuming two symmetry planes for a 3D TO problem, in order to achieve a similar detail level as in the 2D case, the number of design variables should rise from 52 to 208. Using a simple rule of thumb for EA, with the convergence speed $c \sim \mathcal{O}(\frac{1}{n})$ (Bäck (2014)), with n being the dimensionality of the optimization problem, the number of necessary FEM evaluations would rise to a couple of millions.

³ A similar problem was observed when using binary encoding in GAs (Bäck (1996)). To avoid the nonlinearity

Secondly, a strong dependency of the compliance on the design variable x_1 , corresponding to the elongation of the component, can be observed. This seems to be obvious, since the principal goal of the optimization in this case is to connect the load to the support, which corresponds to a significant rise in the stiffness of the structure, where the thickness of the connection plays a secondary role. As shown in Figure 4.8, in a problem with three design variables, there is still only one dominant direction in the search space, making the effective optimization problem one-dimensional. This tendency should hold also for higher-dimensional problems, with more geometric components of higher complexity, since there would be always a reduced set of components that play the most important role from the perspective of carrying the load. This property helps considerably to overcome the curse of dimensionality, which was the main problem of the grid representations. Together with the ability of more sophisticated ESs to neglect the unimportant search directions through a proper adaptation of the step sizes, this should allow for a significant reduction of the effective dimensionality of the optimization problems, resulting in a much lower number of necessary FEM evaluations. Thirdly, as shown in Figure 4.9, it seems that through an appropriate parametrization of geometric components, a very strong decoupling of design variables can be obtained. This contrasts with the grid representations, where only a correlated change of several design variables could lead to a significant improvement of the structure.

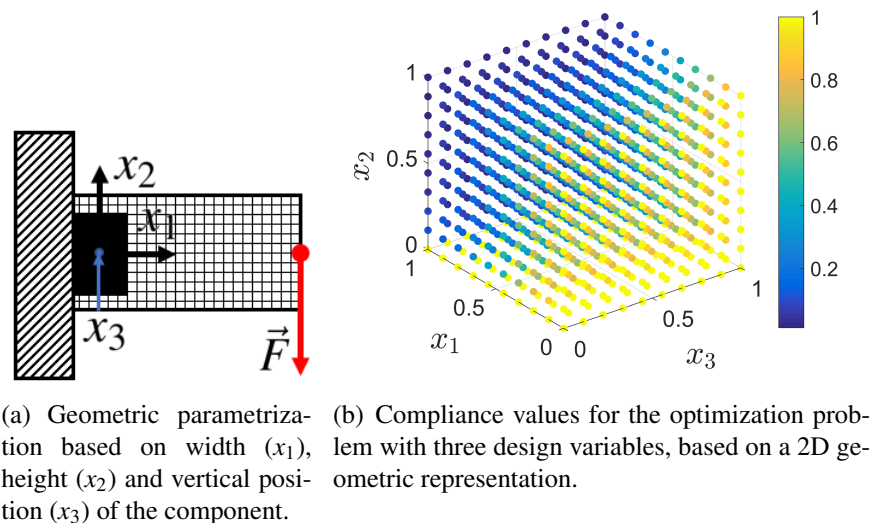


Figure 4.8 A three-dimensional optimization based on the geometric representation.

The analysis presented above is based on very simple optimization problems, with very few design variables. Obviously, the higher-dimensional problems are much more complex, and the conclusions derived here do not have to hold in general case, especially in case of crash TO. Nevertheless, if some of the problems discussed above appear already in case of a simple compliance minimization task, it seems reasonable to assume that they may appear in more complicated cases, as well.

introduced by the binary encoding, Gray code was proposed.

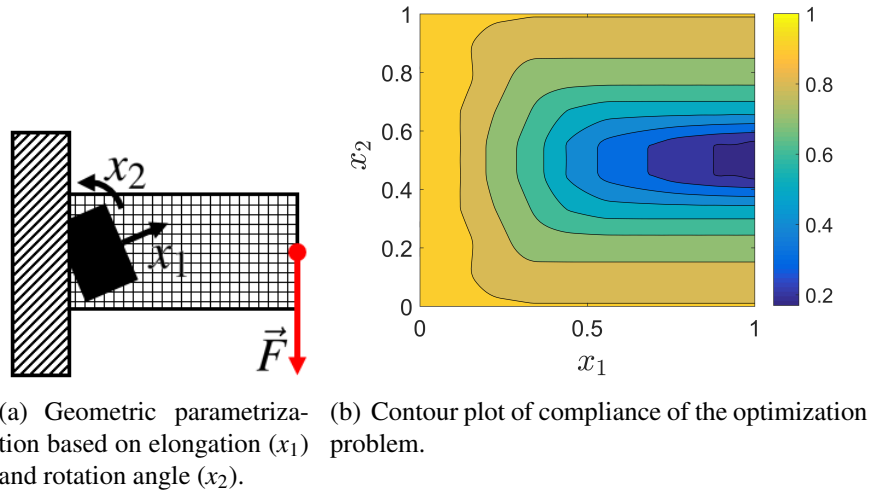


Figure 4.9 Decoupling of the design variables in a geometric representation. A two-dimensional optimization problem shown in Figure 4.9(a) could be in principle replaced by two 1D optimization problems, which could be solved one after another.

4.4. Conclusions

Taking the considerations from the previous sections, as well as the discussions from Chapter 2, Section 2.3 into account, the following prerequisites for an efficient representation for evolutionary-based structural TO can be formulated:

- The dimensionality of the optimization problem should be reduced as much as possible, since it influences strongly the number of (potentially very expensive) FEM evaluations necessary to converge to the optimum.
- With a low number of design variables (genotype), the representation should allow for an expression of as wide range of designs (phenotypes) as possible, but only with features useful from the point of view of the problem being solved. For instance, if the optimal designs in structural TO are frequently composed of beam-shaped components, justified also from the mechanical point of view (Michell (1904)), it makes sense to include this feature already at the representation level. Moreover, representations allowing for generation of a large number of designs not feasible from the structural point of view, e.g. not connecting the load to the supports, should be avoided.
- One should avoid representations with inherently correlated design variables, where very slow explorative search might be needed to obtain satisfactory solutions.
- If a given representation results in a more linear mapping between design variables and the material distribution, it might also make the optimization problem at hand easier to solve. Therefore, such a representation should be favored over representations introducing strong nonlinearities already in the genotype-phenotype mapping.

From the perspective of engineering requirements for TO, the following features of the represen-

tation are favorable:

- Representations allowing for an easy transformation of the geometric description to the finite element and parametric Computer-Aided Design (CAD) models are of high importance for the product development process.
- Representations using 0-1 material distribution, where the elements in the non-material regions (density equal 0) are deleted from the finite element mesh and no intermediate density elements exist, are a must for crashworthiness applications. Additionally, it is much easier to interpret and manufacture such designs.
- The optimization results should be possibly independent of the particular finite element mesh used for the simulations. Artifacts such as single-node connected elements or checkerboard patterns should be additionally avoided.
- In particular, from the perspective of crashworthiness applications, the representation should allow for a use of different than only density-based mapping to material domain. In particular, conforming mappings (Dijk et al. (2013)), allowing for avoiding zig-zag-like finite element meshes, which lead to stress singularities not appropriate for TO using plasticity or failure, are recommended (Maute et al. (1998)).
- Relying on representations widely used in the structural optimization community would be an advantage, allowing potentially for combining the evolutionary-based methods with efficient gradient-based optimizers.

Following the requirements listed above, in the next chapter, a level-set-based representation suitable for solving crash TO problems with use of non-gradient optimization techniques, including evolutionary optimization, is described.

Chapter 5

Evolutionary level set method (EA-LSM)

As discussed in Chapter 2, the state-of-the-art approaches for crash TO rely mostly on very strong assumptions about the properties of the optimization problem and simplify it considerably. Those assumptions, having frequently a heuristic character, are often arguable and might be not true in general case. This motivates the development of novel TO approaches that would be free from those heuristic assumptions. This chapter introduces a possible solution of this problem via utilization of EAs and a suitable low-dimensional representation based on a level-set description. The chapter is structured as follows. Section 5.1 gives an introduction to the proposed method and justifies the choices made in this work. In Section 5.2, the parametrization used in EA-LSM is formally defined. Section 5.3 describes the underlying optimization algorithm. Finally, Section 5.4 summarizes this part of the contribution and discusses the potential and limitations of the proposed method.

5.1. Introduction

In Chapter 4, the role of representations in EC has been broadly discussed. Design of an appropriate representation as well as dedicated search operators fitting well its properties is crucial for the performance of any optimization process (Rothlauf (2006)). In the context of evolutionary structural TO, we distinguished such aspects as low-dimensional representation of the design, small size of the search space and validity of the mutated designs, generality of the method, 0-1 material distribution, direct connection to CAD modeling as well as compatibility with the mainstream TO methods to be vital for the success of a method.

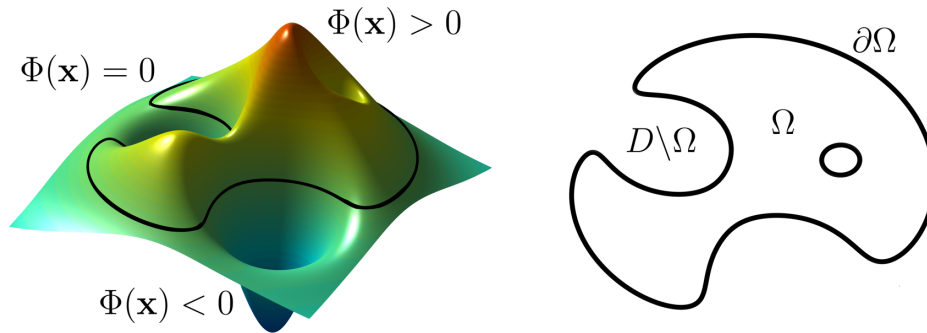


Figure 5.1 Illustration of the core concept in the LSMs (Bujny et al. (2017a)). The LSF Φ (left) defines the regions occupied by material ($\Phi(\mathbf{x}) > 0$) and void ($\Phi(\mathbf{x}) < 0$), where $\mathbf{x} = [x, y]^T$ denotes the vector of Cartesian coordinates in a 2D design domain. As a result, the interface between those two phases (right) is clearly defined by $\Phi(\mathbf{x}) = 0$. This is the key difference w.r.t. the density-based methods (Bendsøe and Sigmund (2004)).

Following those requirements, this work proposes a representation based on an implicit description inspired by the LSM (Allaire et al. (2004, 2005); Dijk et al. (2013)). In contrast to the density-based approaches (Bendsøe and Sigmund (2004)), the design is not parametrized explicitly by defining the densities of the voxels filling the design domain, but implicitly, by specifying the LSF indicating regions occupied by material and void (Figure 5.1). As a result, the interface between material and void is clearly defined, solving the main problem of density-based approaches, being ambiguous intermediate material densities. Unlike the methods based on explicit boundary description, e.g. with NURBS (Cervera and Trevelyan (2005)), LSMs allow for a very convenient treatment of any topological changes, by modifying the values of the LSF. Additionally, unlike density-based approaches, most LSMs do not involve such numerical artifacts as checkerboard patterns or single-node-connected elements. Finally, thanks to decoupling of the finite element mesh and the material parametrization, the number of design variables can be considerably reduced by defining the LSF in terms of local basis functions, which is a crucial property exploited in this work.

The method proposed here is not an LSM in a classical sense. Most of TO LSM approaches rely on an analytical gradient information and use it to propagate the LSF boundaries by solving the Hamilton-Jacobi partial differential equation (Dijk et al. (2013)). In this work, the global LSF is composed out of parametrized geometric basis functions, which are directly modified by the optimizer by changing parameters describing their geometric features. Those basis functions can be viewed as elementary beam components, which can deform and move inside the design domain. Due to those properties, representations of this type are frequently referred to as Moving Morphable Component (MMC) frameworks (Guo (2014)). The use of such representation complies with the conclusions from Chapter 4, taking advantage of the fact that any type of topology can be decomposed into a finite number of components. In particular, topologies optimized for various structural criteria (e.g. maximization of stiffness or buckling loads) with gradient-based methods can be often very easily decomposed into several beam components. This observation motivates using beam-like components as the elementary building blocks of TO approaches

Beside already mentioned benefits coming from using MMC-based representation, it offers much more control over the geometric features of the design, e.g. length scale or curvature (Guo (2014)). This is very difficult in density-based approaches, where no geometry information is explicitly embedded into the representation. In particular, it becomes very important once manufacturing constraints are considered and plays a central role for approaches addressing the problem of structural feature control, e.g. maximum/minimum length scale, minimum curvature, geometric complexity. Within the MMC framework, components with curved skeletons (Guo et al. (2016)) and more complex shapes (Zhang et al. (2015)), or even hollow structures (Bai and Zuo (2020)) can be modeled.

Finally, the consistency of the used representation with the gradient-based MMC approaches offers the possibility to benefit from the research developments of these methods. The approaches based on an explicit parametrization of the design with use of geometric components (Guo (2014); Norato et al. (2015)) gained a lot of attention in the past four years and are slowly becoming widely accepted in structural TO community. This is especially interesting in the context of hybrid methods, where the analytical gradient information can be used to enhance the evolutionary search (Section 6.1.4).

The key advantage of the representation used in this work, being the possibility to significantly reduce the number of design variables, is also its main drawback. Obviously, with limited number of MMCs, the number of designs which can be represented is considerably lower than for the density-based approaches. As a result, for some problems, density-based methods can yield better results even if they do not address specific objectives directly, thanks only to a much better structural attainability.

5.2. Parametrization

As mentioned in the introduction, the parametrization of the design in this work is based on the implicit level set description (Figure 5.1). The design is defined with use of a set of local level set basis functions (or MMCs), which can overlap and freely move within the design domain. Below, a formal description of the proposed representation is presented.

The global level set function, Φ , is given by:

$$\begin{cases} \Phi(\mathbf{x}) > 0, & \text{if } \mathbf{x} \in \Omega, \\ \Phi(\mathbf{x}) = 0, & \text{if } \mathbf{x} \in \partial\Omega, \\ \Phi(\mathbf{x}) < 0, & \text{if } \mathbf{x} \in D \setminus \Omega, \end{cases} \quad (5.1)$$

where \mathbf{x} is a position vector in the design domain D and Ω is the area inside the design domain occupied by the material phase. As a consequence, $D \setminus \Omega$ is the part of the design domain occupied

by void, and $\partial\Omega$ refers to the boundary between material and void.

Similarly, the local level set basis function of the i^{th} MMC is defined as follows:

$$\begin{cases} \phi_i(\mathbf{x}) > 0, & \text{if } \mathbf{x} \in \Omega_i, \\ \phi_i(\mathbf{x}) = 0, & \text{if } \mathbf{x} \in \partial\Omega_i, \\ \phi_i(\mathbf{x}) < 0, & \text{if } \mathbf{x} \in D \setminus \Omega_i, \end{cases} \quad (5.2)$$

with Ω_i being the region in the design domain D where the i^{th} MMC is located. As a result, the part of the design domain occupied by the material is:

$$\Omega = \bigcup_{i=1}^M \Omega_i, \quad (5.3)$$

where M is the number of MMCs. In this work, a local level set basis function as proposed by Guo (2014)¹ is used. For $D = \mathbb{R}^2$, $\mathbf{x} = [x, y]^T$, it takes the following form:

$$\begin{aligned} \phi_i(\mathbf{x}) = & - \left(\left(\frac{\cos \theta_i (x - x_{0i}) + \sin \theta_i (y - y_{0i})}{l_i/2} \right)^q \right. \\ & \left. + \left(\frac{-\sin \theta_i (x - x_{0i}) + \cos \theta_i (y - y_{0i})}{t_i/2} \right)^q - 1 \right), \end{aligned} \quad (5.4)$$

where (x_{0i}, y_{0i}) is the position of the center of the i^{th} MMC (Figure 5.2(a)) of length l_i and thickness t_i . The rotation of the component with respect to the horizontal direction is defined by the rotation angle θ_i . Similarly to Guo (2014), we use $q = 6$ as a modeling exponent, which results in shapes of MMCs close to rectangular. We assume that such shapes better represent bar-like connections present usually in optimal designs than, for instance, elliptical MMCs obtained for $q = 2$. Figure 5.2 shows the resulting parametrization with the LSF given by Equation (5.4).

In order to map the LSF to the finite element model, as in the most of TO approaches (Bendsøe and Sigmund (2004)), a standard density-based geometry mapping (Dijk et al. (2013)) is used². In such a case, the relation between the global LSF Φ and the material density $\rho(\mathbf{x})$ at position $\mathbf{x} \in D$ is defined as:

$$\rho(\mathbf{x}) = H(\Phi(\mathbf{x})), \quad (5.5)$$

where:

$$\Phi(\mathbf{x}) = \max(\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots, \phi_M(\mathbf{x})), \quad (5.6)$$

¹ In the later works, Guo et al. (2016); Zhang et al. (2016) refer to this LSF as to the Topology Description Function (TDF) (de Ruiter and van Keulen (2004)).

² More sophisticated geometry mapping techniques, known from the field of level set TO, like immersed boundary or conforming mapping (Dijk et al. (2013); Kudela et al. (2015)) could be also used with the presented approach. We use the density-based mapping due to its simplicity and much lower computational costs. However, similarly to density-based optimization approaches, this mapping results in a jagged geometry, showing frequently unrealistic stress concentrations during the crash simulations.

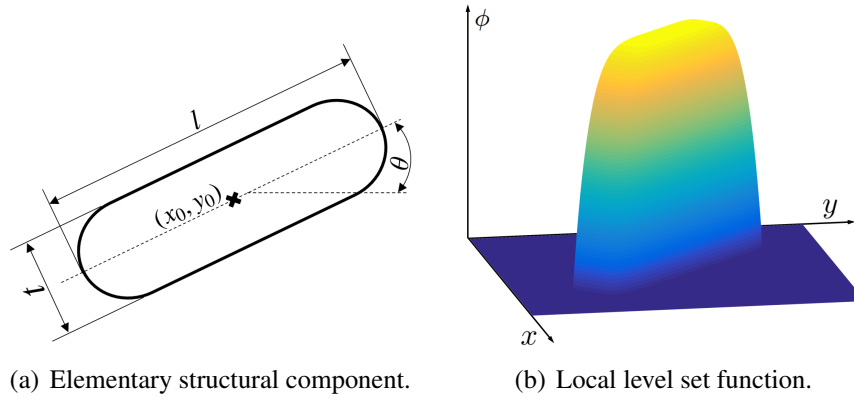


Figure 5.2 Parametrization of the MMC and the corresponding LSF (with negative values set to zero) (Guo (2014); Bujny et al. (2016c,b, 2017a)).

and $H(x)$ denotes the Heaviside function:

$$H(x) = \begin{cases} 0, & \text{if } x < 0 \\ 1, & \text{if } x \geq 0. \end{cases} \quad (5.7)$$

In case of static, linear elastic TO cases, the finite elements with $\rho(\mathbf{x}) = 0$ are assigned a very small density ρ_{min} , e.g. $\rho_{min} = 0.01$. Such a weak material is frequently referred to as ersatz material (Sigmund and Maute (2013); Zhang et al. (2016)) and is used to guarantee better numerical stability in the finite element simulation. The density ρ is used then to scale the stiffness tensor \mathbf{E} as follows:

$$\mathbf{E}(\mathbf{x}) = \rho(\mathbf{x}) \mathbf{E}^0, \quad (5.8)$$

where \mathbf{E}^0 is the reference stiffness tensor.

In contrast, in dynamic, nonlinear crash cases, the elements with $\rho(\mathbf{x}) = 0$ are removed from the finite element mesh, which is crucial to guarantee a proper physical behavior of the simulated model³.

Equations (5.1)–(5.8) define the parametrization of the mechanical design. The entire procedure of transforming the parametric description to design properties resulting from the numerical simulation is shown in Figure 5.3.

The approach described above can be easily extended to 3D representations. In such a case, $D = \mathbb{R}^3$

³ For the plasticity models assuming incompressibility, e.g. von Mises or Tresca, even the elements with highly penalized material properties show plastic strain and therefore keep the volume constant in the deformation.

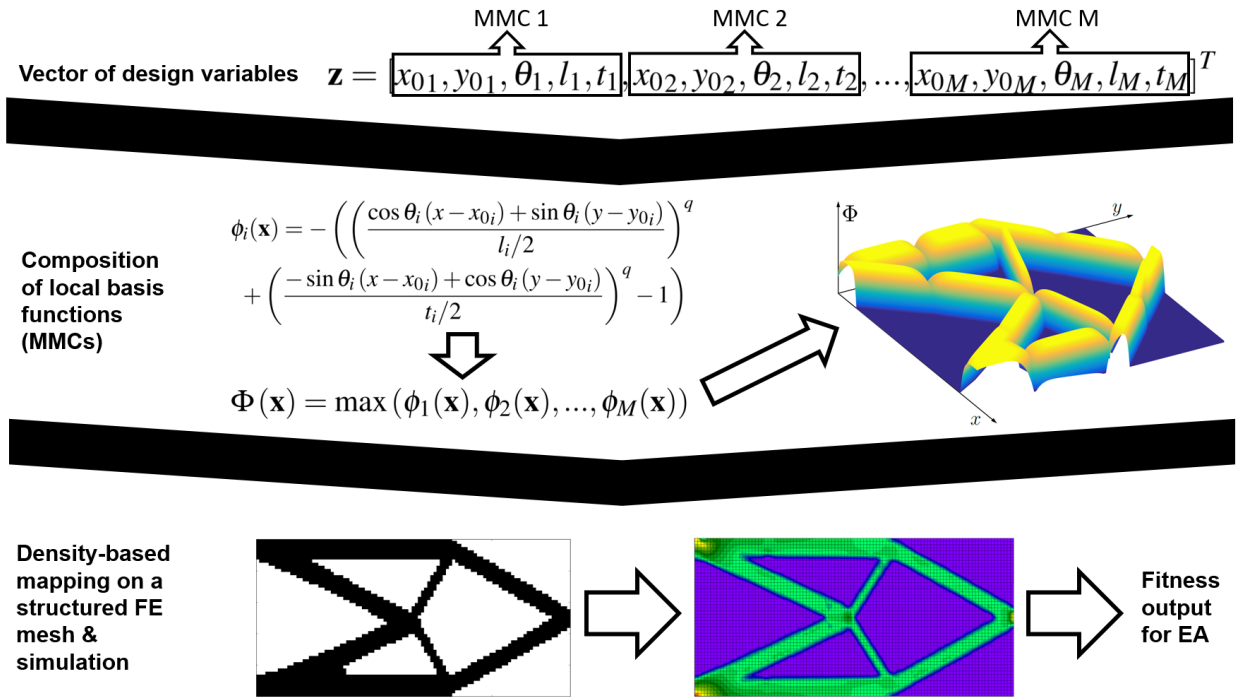


Figure 5.3 Illustration of the mapping of design variables to the mechanical design used for evaluation of structural performance via FEM in EA-LSM (Bujny et al. (2016c, 2017a)).

and $\mathbf{x} = [x, y, z]^T$. As a result, the local level set function defining the i^{th} component, is given by:

$$\phi_i(\mathbf{x}) = - \left(\left(\frac{x'}{l_i/2} \right)^q + \left(\frac{y'}{t_i/2} \right)^q + \left(\frac{z'}{h_i/2} \right)^q - 1 \right), \quad (5.9)$$

with:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_{t_3} \cdot R_{t_2} \cdot R_{t_1} \cdot \begin{bmatrix} x - x_{0i} \\ y - y_{0i} \\ z - z_{0i} \end{bmatrix}, \quad (5.10)$$

and:

$$R_{t_1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i \\ 0 & \sin \alpha_i & \cos \alpha_i \end{bmatrix}, R_{t_2} = \begin{bmatrix} \cos \beta_i & 0 & \sin \beta_i \\ 0 & 1 & 0 \\ -\sin \beta_i & 0 & \cos \beta_i \end{bmatrix}, R_{t_3} = \begin{bmatrix} \cos \gamma_i & -\sin \gamma_i & 0 \\ \sin \gamma_i & \cos \gamma_i & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (5.11)$$

where, similarly to the 2D parametrization, (x_{0i}, y_{0i}, z_{0i}) denotes the position of the center of the i^{th} component (Figure 5.4) of length l_i , thickness t_i , and height h_i . The translation and rotation of the component are realized by a transformation of the coordinate system using Equations (5.10)

and (5.11). The rotation transformation is carried out according to the Euler angles: α_i , β_i , and γ_i . Figure 5.5 illustrates the rotation transformation according to Equations (5.10) and (5.11). In this work, as for the 2D parametrization, $q = 6$ is used.

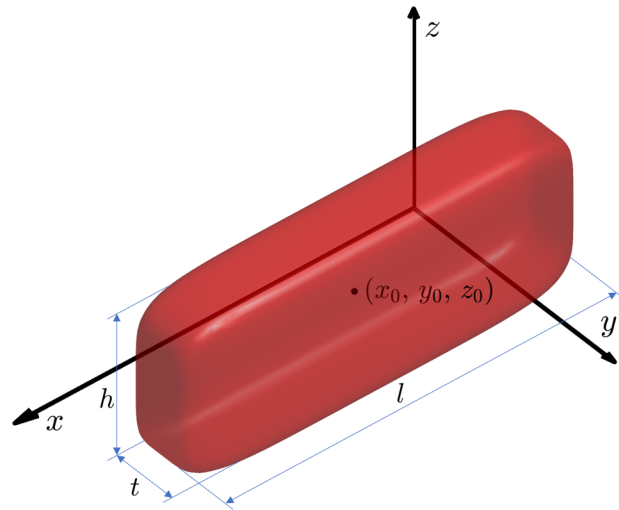


Figure 5.4 0th iso-surface of a level set basis function given by Equation (5.9).

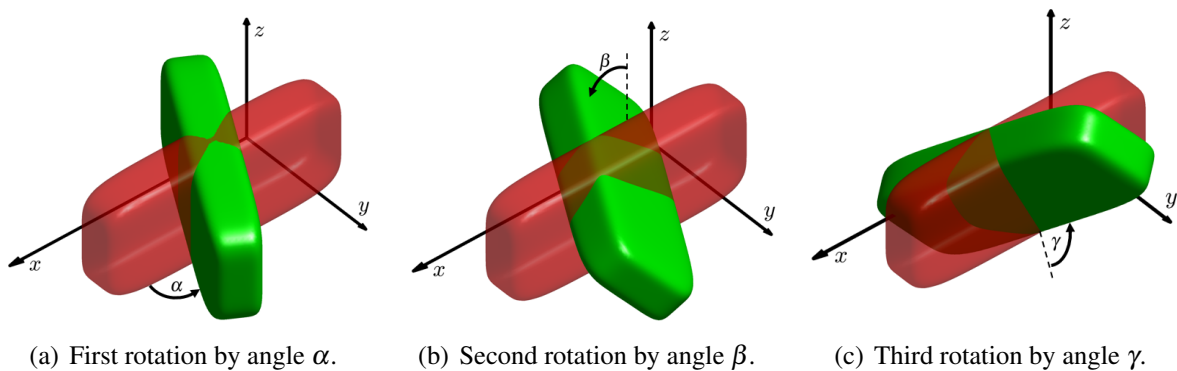


Figure 5.5 Illustration of the rotation transformation according to Equations (5.10) and (5.11). Plots of the 0th iso-surface of the local LSF (Equation (5.9)) before transformations (red) and after consecutive rotations according to the Euler angles α , β , and γ (green).

Similarly to the 2D case, a finite set of local LSFs (Equation (5.9)) is used to compose the global LSF according to Equation (5.6). The mapping of the global LSF to the material density is realized with use of Equation (5.5).

The parametrization can be further extended by introducing level set basis functions of different forms and complexity. Guo et al. (2016) proposed for instance to use curved MMCs or MMCs with variable thickness over the length. However, it is not clear which parametrization is better. On one hand, higher-order MMC representations can describe more complex shapes, but involve more design variables per element. On the other hand, any design can be represented with use of a sufficiently high number of low-order MMCs, as illustrated in Figure 5.6.

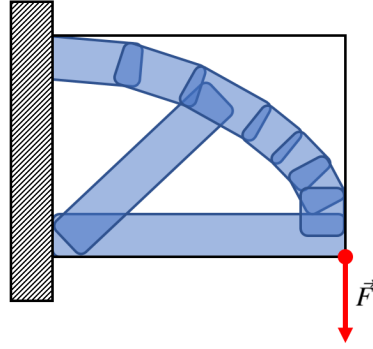


Figure 5.6 Illustration of how a curved shape can be approximated with a finite number of straight components after Guo (2014).

5.3. Optimization algorithm

In general, optimization problems addressed in this work can be expressed in the following form:

$$\begin{aligned}
 & \min_{\mathbf{z}} f_{\text{obj}}(\mathbf{z}), \mathbf{z} \in \mathbb{R}^n; \\
 & \text{s.t. } \mathbf{res}(t) = \mathbf{0}; \\
 & \quad g_{C_j}(\mathbf{z}) \leq 0, j = 1, \dots, n_{\text{ineq}}, \\
 & \quad h_{C_k}(\mathbf{z}) = 0, k = 1, \dots, n_{\text{eq}}, \\
 & \quad z_i^L \leq z_i \leq z_i^U,
 \end{aligned} \tag{5.12}$$

where f_{obj} is the objective function to be minimized and $\mathbf{z} = [z_1, z_2, \dots, z_n]^T$ denotes the vector of design variables, being a collection of all the parameters defining M basis functions (see Figure 5.3). As a result, the dimensionality of the optimization problem is $n = n_p \cdot M$, where n_p is the number of parameters per basis function, equal to 5 for 2D and 9 for 3D representation in our case. Functions g_{C_j} and h_{C_k} are the inequality and equality constraints, respectively. Additionally, for each design variable z_i , a lower bound z_i^L and an upper bound z_i^U can be defined. Finally, the condition $\mathbf{res}(t) = \mathbf{0}$ expresses the dynamic equilibrium of the structure at time t .

5.3.1. Algorithm overview

As the optimization method, EA-LSM uses EAs, in particular ES and CMA-ES, to solve structural TO problems using the level-set-based parametrization defined in Section 5.2. These methods exhibit better performance on continuous optimization problems than other EAs (e.g. GAs) (Bäck (1996)). As a result, with a few problem-specific modifications, the optimization algorithm resembles the standard ES/CMA-ES as described in Chapter 3. In particular, our initial experiments (Bujny (2015); Bujny et al. (2016c)) showed that the state-of-the-art CMA-ES optimization algorithm frequently offers superior convergence velocity in terms of the number of fitness/cost function evaluations compared to a standard ES. However, standard ES turned out to be still very useful in the numerical experiments with very large populations, evaluated in parallel on a compu-

tational cluster, where the goal was to maximize the convergence velocity in terms of generations. CMA-ES with larger than recommended (Hansen and Ostermeier (2001); Hansen (2016)) population sizes gave results of inferior performance compared to ES in terms of the achieved fitness values. Therefore, both standard ES and CMA-ES were used as optimization algorithms in this work. The outline of the optimization algorithm with the proposed problem-specific modifications is depicted in Figure 5.7.

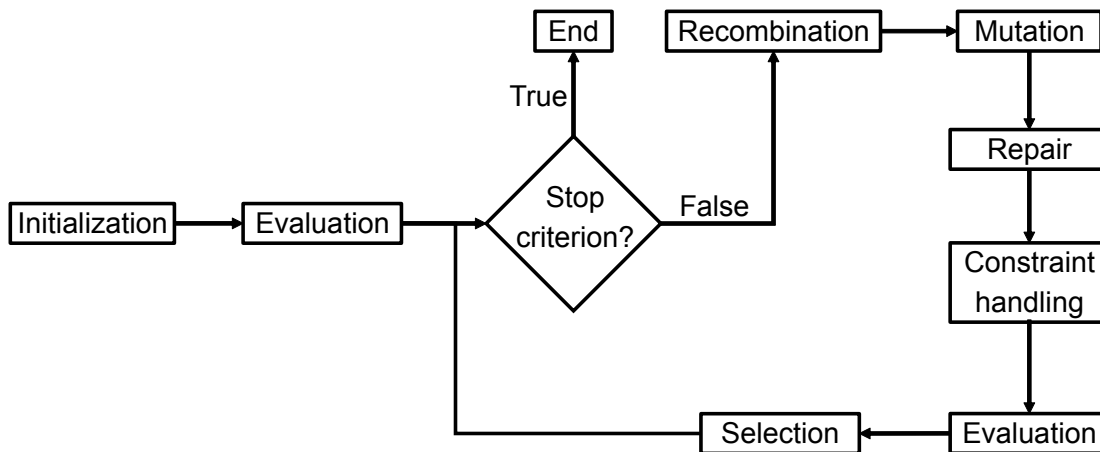


Figure 5.7 Outline of the optimization algorithm used in EA-LSM.

In the initialization step, the initial population of individuals is defined. Each of the μ individuals created at this stage is characterized by a vector of design variables \mathbf{z} encoding the layout of level set basis functions, and a set of strategy parameters \mathbf{s} . In the next step, the evaluation of the initial population with use of FEM, e.g. dynamic nonlinear crash simulations, is carried out. After that, the algorithm enters the main optimization loop consisting of the standard evolutionary operators such as recombination, mutation, and selection as well as problem-specific repair and constraint handling operators. Sections 5.3.2, 5.3.3, and 5.3.4 give a description of the operators tailored to the proposed TO approach. The main optimization loop continues to execute until certain stopping criteria are satisfied. In this work, usually a fixed budget of FE evaluations is assumed and therefore, the optimization stops after reaching a defined number of iterations $iter_{max}$, often referred to as generations in the context of evolutionary optimization.

5.3.2. Initialization

It is a well-known fact that the initialization of an EA can strongly influence its convergence speed and the quality of the final solution (Rahnamayan et al. (2007); Maaranen et al. (2007)). In general, if no information about the optimum is available, a random initialization is the most frequently used approach (Rahnamayan et al. (2007)). In such cases, the initial population should cover the feasible region as uniformly as possible, giving the algorithm the possibility to reach a large part of the feasible search space by means of recombination.

In case of TO approaches based on explicit parametrization with geometric components, an initial

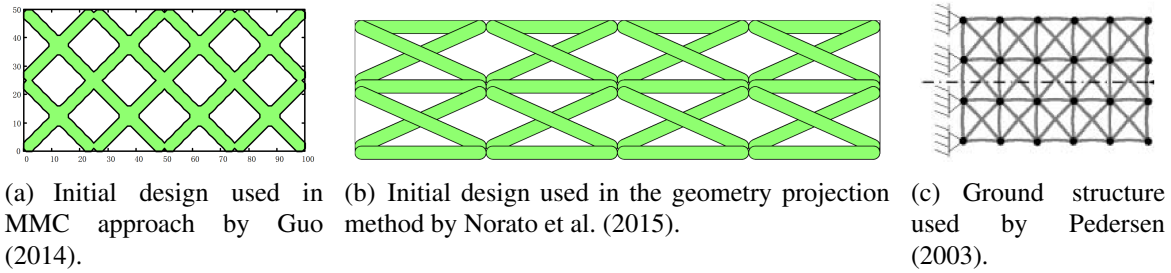


Figure 5.8 Initial designs used in gradient-based approaches using explicit component-based representations.

distribution as shown in Figures 5.8(a) and 5.8(b) is frequently used (Guo (2014); Norato et al. (2015)). Indeed, this type of design offers very good space-filling properties and usually also good structural performance thanks to its similarity to lattice or truss structures. In principle, one could also view the method proposed in this work as an extension of the Ground Structure Approach (GSA), where ground structures as depicted in Figure 5.8(c) are commonly used. Consequently, in this work, diagonal initial designs as depicted in Figure 5.9 are used as a reference.

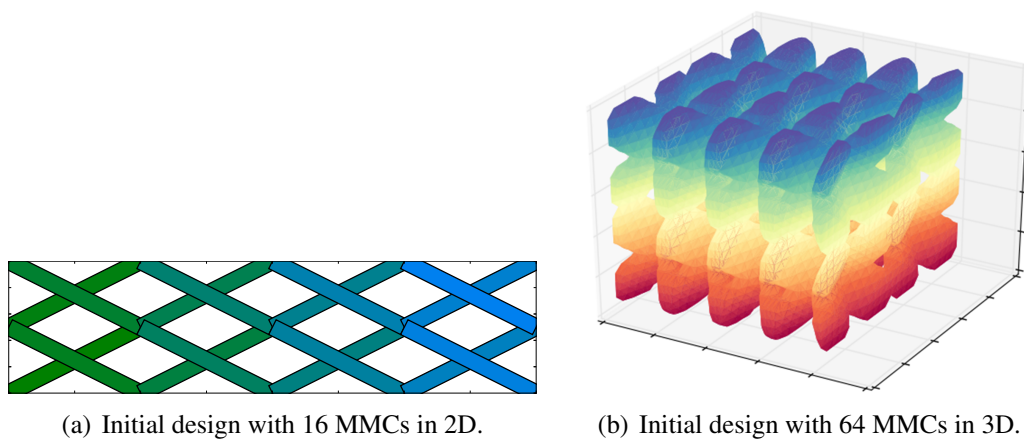


Figure 5.9 Diagonal MMCs' layouts used as reference designs for generation of initial population of individuals, for 2D (left) and 3D (right) problems.

Once the reference topology is generated, the initial population is created by varying the design variable vector of the reference design according to the normal distribution with mean 0 and standard deviation σ , as in the mutation operator. This spreads the initial population around the reference design, creating moderate genetic diversity for small values of σ . For the used representation, resulting in a search space with many structurally infeasible regions (e.g. disconnected structures), this seems to be a reasonable way of creating the initial population. Random initialization would very likely result in structurally-infeasible solutions, while the proposed approach creates mostly connected designs thanks to the feasibility of the reference topology. Additionally, due to the redundancy of the used representation (Rothlauf (2006)), expressed in the possibility of representing the same design with several different vectors of design variables (Figure 5.10), high diversity in the genotype space might result in a low diversity of phenotypes. All in all, using uniform ran-

dom distribution to generate very diverse initial population of individuals is not a good solution in this case and the above mentioned arguments speak in favor of using initial population normally distributed around a good reference design.

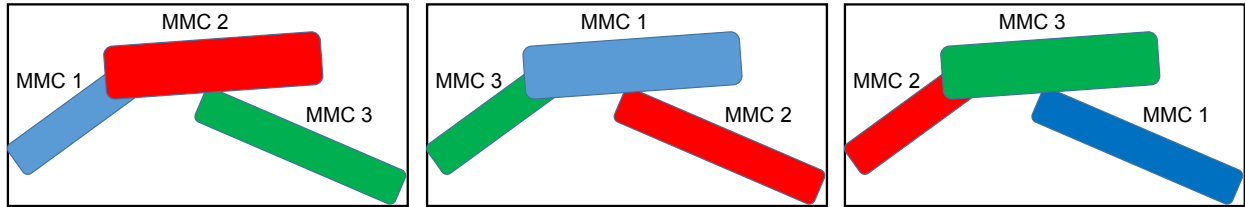


Figure 5.10 Illustration of the redundancy of the used representation. Several different vectors of design variables, corresponding to different configurations of MMCs, can be used to describe exactly the same material distribution.

5.3.3. Repair operators

This section introduces the repair operators, which help to avoid some unfavorable artifacts that can appear during the optimization with EA-LSM. The artifacts are usually associated with the fact of using MMC-based representation together with evolutionary optimizers, but the observations made in this section might be also helpful for improving the performance of the gradient-based optimization approaches as proposed by Guo (2014). Please note that the usage of the repair operators, depending on the problem, can be treated as optional and using pure ES/CMA-ES would already yield very good results in most cases.

Minimal step size

In case of the density-based mapping (Equation (5.8)), where the global LSF is mapped directly on a fixed finite element mesh, a problem due to the discretization of space in the finite element model arises. As illustrated in Figure 5.11, if the change of any design variable parametrizing the design is smaller than a certain value, related to the resolution of the mesh, exactly the same finite element model will be obtained. As a result, the performance of the structure will be exactly the same, resulting in a small plateau in the optimization landscape. Such plateaus pose usually difficulties for ESs, due to the lack of gradient of the objective function, and can result in a premature convergence of the algorithm.

To overcome this problem, a minimal step size σ_{min} can be enforced by introducing a repair operator applied after the mutation step. If any of the step sizes drops below σ_{min} during the self-adaptation process, it is mapped back to σ_{min} . Thanks to the used normalization of the design variables, described in Section 5.3.5, usually the same value of σ_{min} can be used for different types of parameters, representing e.g. thickness or positions of the MMCs.

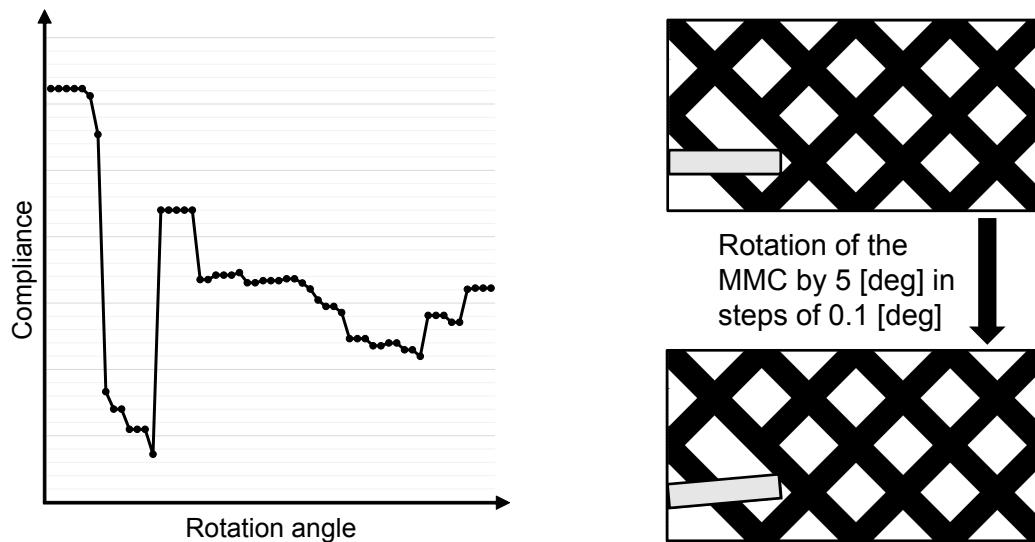


Figure 5.11 Illustration of the discontinuity of the objective function due to the density-based mapping on a fixed finite element mesh. Rotation of a specified component by a too small value can result in no change in the resulting finite element model and, therefore, in no change of the objective function.

Thin components

Another problem associated with the density-based geometry mapping that might appear during the optimization is the presence of MMCs of a very low thickness. Such MMCs can produce artifacts such as single-node-connected or isolated finite elements, influencing the quality of the final design. A simple yet effective solution of this problem is an application of a repair operator, which sets the thickness of a component to zero, once it drops below a certain threshold t_{min} .

Overlapping components

In some cases, one of the key properties of the MMC approach, being the ability of the components to overlap, might pose difficulties for stochastic optimizers. An illustration of such a situation is given in Figure 5.12. Suppose that a component fully covers another one as shown in Figure 5.12(a), and the goal of the optimizer is to minimize the overall volume (or mass) of the structure. To start reducing the volume, the optimizer would have to decrease the design variable x_1 first, since modifications of x_2 , corresponding to the thickness of the green component (fully covered by the black one) would have no effect on the overall volume of the structure. At some point, the optimization will lead to a situation when both of the components will have similar thicknesses. Then, the only possibility to reduce the volume of the structure would be to decrease thicknesses of both of the components simultaneously. This results in a similar problem as with the grid representations (see Chapter 4), when only a correlated mutation of all design variables could result in a significant improvement of the objective function. In case of the overlapping components, a situation where at least one of the components does not decrease its thickness would result in a mapping to exactly the same phenotype. The higher the number of overlapping components, the

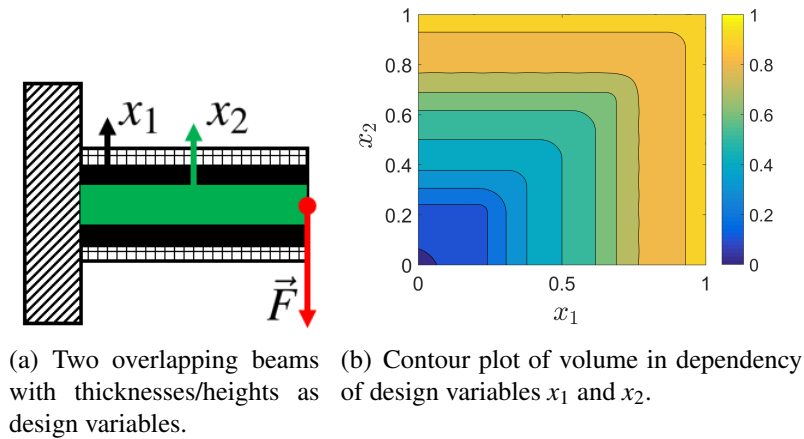


Figure 5.12 Illustration of the problem with overlapping components.

more difficult it would be to generate a correlated change of all relevant design variables by means of a random mutation, which might finally lead to a premature convergence of the optimization.

In order to avoid such situations, a repair operator for modifying the overlapping components can be introduced. In this work, the volume (or area) of the overlapping parts is computed using a structured grid, as shown in Figure 5.13. The overlap o is estimated by calculating the ratio of the elements of the structured grid occupied by the dashed MMC and the union of the other MMCs (red area) to the total number of elements occupied by the dashed MMC only. Once the ratio of the overlapping volume to the total volume of a component reaches o_{max} , the thickness of the component is set to zero. Each component is compared against the union of the remaining components to calculate its overlap with the rest of the structure, yielding a simplified design without large overlaps between MMCs at the end of the process. Please note that the computation of the overlap is done each time based on the updated MMCs. Namely, if the thickness of a component was already set to zero when computing the overlap for the i^{th} MMC, this change will be taken into account when computing the overlap for the $(i+1)^{\text{th}}$ MMC. As a result, it cannot happen that two strongly overlapping MMCs are removed simultaneously, resulting in a complete deletion of a significant amount of material. However, the decision which of the overlapping components should be eliminated is quite arbitrary, especially if only very high o_{max} are considered. As a result, in this work, this results from the order in which the MMCs are compared with each other, which depends only on the MMC index i .

In practice, situations when several components overlap strongly with each other happen relatively rarely. Usually, only some parts of the components overlap and the problem is not as severe as shown in Figure 5.12(b). However, even in such cases, a strong correlation between design variables is observed and can slow down the optimization process considerably.

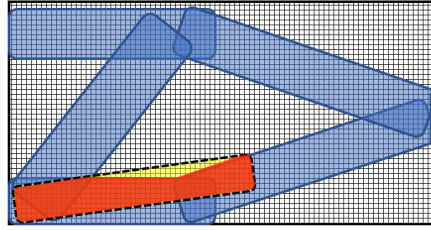


Figure 5.13 Illustration of an MMC (dashed lines) strongly overlapping with the rest of MMCs.

Components outside the design domain

The proposed representation requires a special handling of the MMCs at the boundary of the design domain. During the optimization, it might come to situations as illustrated in Figure 5.14. When one of the ends of the i^{th} component extends beyond the design domain, in some cases, the cost function would no longer change with changes of the parameters x_{0i} , y_{0i} or l_i , resulting in a small plateau in the optimization landscape. As a result, some of the components, which play an important role from the structural point of view, might always extend to the boundaries of the design domain. Figure 5.14 illustrates that problem for a case of compliance minimization under a volume constraint. The marked components (dashed red rectangles) are very long, although it is not justified from the mechanical point of view. The upper parts (dotted green rectangles) do not contribute significantly to the stiffness of the structure, but increase its volume. It would be much more efficient to remove the upper parts and increase the thickness of the other components in the structure. Nevertheless, the optimizer is not able to achieve that, since small variations of the length or positions of the components in the mutation step do not result in any change of the cost function. As a result, the components can change their length according to a random walk (Pearson (1905)), due to the lack of selective pressure.

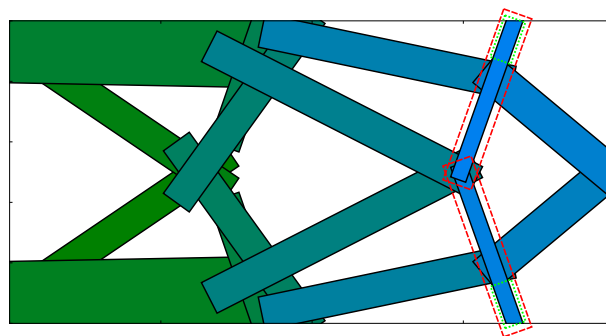


Figure 5.14 Illustration of the problem with components extending outside the design domain for a cantilever beam compliance minimization with a volume constraint. The MMCs marked using red dashed lines play an important role from the mechanical point of view, but contribute to the volume of the structure much more than necessary. In such a situation, the optimizer does not reduce their thickness to zero, but is also not able to shorten the beams to eliminate the redundant parts (green dotted rectangles) due to the plateau in the optimization landscape.

In principle, the problem characterized above could be solved with use of penalty methods described in the next section. However, this would make the optimization problem itself much more difficult to solve, by considerably changing the landscape of the cost function. In particular, in the

situations where the population of individuals focuses in the region close to the constraint, a lot of them might be rejected due to constraint violation, resulting in a much slower convergence of the optimization.

An alternative approach is to explicitly treat the problem with a suitable repair operator. For a 2D TO problem and a rectangular design domain, one can easily find the coordinates of the ends of the i^{th} component:

$$(x_{end_{1i}}, y_{end_{1i}}) = \left(x_{0i} - \frac{l_i \cos(\theta_i)}{2}, y_{0i} - \frac{l_i \sin(\theta_i)}{2} \right) \quad (5.13)$$

$$(x_{end_{2i}}, y_{end_{2i}}) = \left(x_{0i} + \frac{l_i \cos(\theta_i)}{2}, y_{0i} + \frac{l_i \sin(\theta_i)}{2} \right) \quad (5.14)$$

The repair algorithm calculates the coordinates of the ends of all the MMCs and moves them back to the boundary of the design domain, if necessary:

$$x_{end_{bi}} = \begin{cases} 0, & \text{if } x_{end_{bi}} < 0 \\ w_{dd}, & \text{if } x_{end_{bi}} > w_{dd}, \end{cases} \quad (5.15)$$

$$y_{end_{bi}} = \begin{cases} 0, & \text{if } y_{end_{bi}} < 0 \\ h_{dd}, & \text{if } y_{end_{bi}} > h_{dd}, \end{cases} \quad (5.16)$$

where b is an index indicating the end of an MMC (1 or 2), w_{dd} is the width, and h_{dd} is the height of the rectangular design domain.

Finally, for each of the varied components, the parameters x_{0i} , y_{0i} , θ_i , and l_i are calculated based on the following formulas:

$$x_{0i} = \frac{x_{end_{1i}} + x_{end_{2i}}}{2} \quad (5.17)$$

$$y_{0i} = \frac{y_{end_{1i}} + y_{end_{2i}}}{2} \quad (5.18)$$

$$l_i = \sqrt{(x_{end_{1i}} - x_{end_{2i}})^2 + (y_{end_{1i}} - y_{end_{2i}})^2} \quad (5.19)$$

$$\theta_i = \begin{cases} \arccos\left(\frac{x_{end_{2i}} - x_{end_{1i}}}{l_i}\right), & \text{if } y_{end_{1i}} < y_{end_{2i}} \\ \pi - \arccos\left(\frac{x_{end_{2i}} - x_{end_{1i}}}{l_i}\right), & \text{if } y_{end_{1i}} > y_{end_{2i}}. \end{cases} \quad (5.20)$$

Please note that, due to symmetry, the same MMCs are obtained for θ_i and $\theta_i + k\pi$ with $k \in \mathbb{Z}$, which was used in Equation (5.20).

Similarly, for a 3D TO case, for the i^{th} MMC, the coordinates of its ends can be easily found as follows:

$$\begin{bmatrix} x_{end_{1i}} \\ y_{end_{1i}} \\ z_{end_{1i}} \end{bmatrix} = R_{t_3} \cdot R_{t_2} \cdot R_{t_1} \cdot \begin{bmatrix} -\frac{l_i}{2} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} x_{0i} \\ y_{0i} \\ z_{0i} \end{bmatrix}, \quad (5.21)$$

$$\begin{bmatrix} x_{end_{2i}} \\ y_{end_{2i}} \\ z_{end_{2i}} \end{bmatrix} = R_{t_3} \cdot R_{t_2} \cdot R_{t_1} \cdot \begin{bmatrix} \frac{l_i}{2} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} x_{0i} \\ y_{0i} \\ z_{0i} \end{bmatrix}. \quad (5.22)$$

If necessary, the ends of the MMC can be mapped back to the design domain using Equations (5.15), (5.16) and additionally:

$$z_{end_{bi}} = \begin{cases} 0, & \text{if } z_{end_{bi}} < 0 \\ d_{dd}, & \text{if } z_{end_{bi}} > d_{dd}, \end{cases} \quad (5.23)$$

where d_{dd} is a parameter defining the depth of the design space.

In addition, we find the angle of rotation of the MMC around its longitudinal axis ω_i , which is assumed to remain the same after moving of the ends of MMCs inside the design domain:

$$\omega_i = \begin{cases} \arccos(r_{\omega_z}), & \text{if } r_{\omega_y} > 0 \\ \pi - \arccos(r_{\omega_z}), & \text{if } r_{\omega_y} < 0, \end{cases} \quad (5.24)$$

where r_{ω_y} and r_{ω_z} :

$$\begin{bmatrix} r_{\omega_x} \\ r_{\omega_y} \\ r_{\omega_z} \end{bmatrix} = R_{t_3} \cdot R_{t_2} \cdot R_{t_1} \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. \quad (5.25)$$

In order to find the values of the MMC parameters after moving the ends into the design space, the position of the center and the length of the MMC is calculated based on Equations (5.17) and (5.18) as well as the following formulas:

$$z_{0i} = \frac{z_{end_{1i}} + z_{end_{2i}}}{2}, \quad (5.26)$$

$$l_i = \sqrt{(x_{end_{1i}} - x_{end_{2i}})^2 + (y_{end_{1i}} - y_{end_{2i}})^2 + (z_{end_{1i}} - z_{end_{2i}})^2}. \quad (5.27)$$

The rotation angles α_i , β_i , and γ_i are found by solving a nonlinear system of Equations (5.22) and (5.25) for the fixed value of ω_i and updated values of x_{0i} , y_{0i} , z_{0i} , and l_i . This can be done numerically using e.g. the Newton-Raphson method (Gil et al. (2007)).

The proposed approach can be extended to deal with design spaces of more complex shapes, but in this work, only rectangular/cuboid domains will be considered.

The repair operator presented in this paragraph, together with the repair operators for handling the minimal step size, as well as deletion of thin and overlapping components, allows for an efficient elimination of unfavorable artifacts appearing in EA-LSM.

5.3.4. Constraint handling

EAs are, in principle, unconstrained optimization methods. On the other hand, industrial cases are usually highly constrained optimization problems. In the vehicle structural optimization context, such constraints involve typically crashworthiness, stiffness, as well as manufacturing requirements. In order to incorporate constraint handling in EA, the constrained optimization problem has to be transformed into an unconstrained one. After that step, the algorithm can be applied directly to the transformed problem. Below, a description of the constraint handling methods used in this work is given.

Inequality and equality constraints

Inequality constraints $g_{Cj}(\mathbf{z})$ as well as equality constraints $h_{Ck}(\mathbf{z})$ are handled with use of the exterior penalty method with static penalties (Coello Coello (2002)), which is the most commonly used approach for constraint handling in evolutionary optimization. The main idea in this approach is to transform the constrained problem into unconstrained by adding penalty value to the objective function being minimized whenever any of the constraints is violated. Unlike interior methods, it does not require a feasible initial design and can explore infeasible designs, e.g. in order to move between disjoint feasible regions in the search space.

In the most general case, the minimized cost function $f(\mathbf{z})$, including the constraint terms, would take the following form:

$$f(\mathbf{z}) = f_{\text{obj}}(\mathbf{z}) + \sum_{j=1}^{n_{\text{ineq}}} P_j \max(0, g_{C_j}(\mathbf{z}))^\delta + \sum_{k=1}^{n_{\text{eq}}} L_k \max(0, |h_{C_k}(\mathbf{z}) - \varepsilon|)^\delta, \quad (5.28)$$

where P_j and L_k are the static penalty factors. In this work, usually $P_j = L_k = P$, with P being a large constant. In order to handle equality constraints, which are very rare in crashworthiness optimization, they are transformed to inequality constraints ($|h_{C_k}(\mathbf{z}) - \varepsilon|$), where ε is a small tolerance value. The exponent δ is normally equal to 1 or 2. Without loss of generality, only inequality constraints will be considered later on in this work. The objective function $f_{\text{obj}}(\mathbf{z})$, as well as constraints $g_{C_j}(\mathbf{z})$, are always normalized using the values calculated for the reference design (Section 5.3.5).

Although more sophisticated constraint handling methods for EA exist (e.g. adaptive penalty methods, hybrid methods, multiobjective optimization techniques), the performance of different approaches is usually similar (Coello Coello (2002)), which justifies the choice of the simpler approach in this work.

The constraint handling method presented above does not impose limitations on the number of constraints that can be considered in the optimization. This is a clear advantage over state-of-the-art TO methods such as HCA, where dealing with more than one constraint causes difficulties (Zeng (2018)).

Bounds on design variables

In order to enforce the bounds on the design variables (Equation (5.12)), a transformation as described by Bäck et al. (2013) is applied individually to each of the n components of the vector \mathbf{z} :

$$z_i = z_i^L + (z_i^U - z_i^L) \frac{2}{\pi} \arcsin \left(\left| \sin \left(\frac{\pi (z_i - z_i^L)}{2 (z_i^U - z_i^L)} \right) \right| \right). \quad (5.29)$$

The transformation is introduced to perform reflection at the bounds, which prohibits the design variables to take values out of the specified range $[z_i^L, z_i^U]$. Imposing the bounds can be treated as optional, e.g. to restrict the range of thicknesses of the MMCs in the optimized structures, and is not used in numerical experiments presented in this work.

5.3.5. Normalization of design variables

Typically, numerical optimization methods aim for transforming the optimization problem in such a way that the unit change of each design variable has a similar effect on the objective function (Arora (2012)). In particular, in case of ESs, often separate step sizes for each of the design variables are used to achieve that (Bäck and Schwefel (1993)). However, if the influence of different

design variables on the optimized cost function is radically different, this is associated with higher learning effort for the optimizer, and reduces its convergence velocity.

As a result, to simplify the optimization problem already at the beginning, a common standard is to normalize the design variables (Arora (2012)). Depending on the type of the parameters, we propose to use the following transformation of the original variables parametrizing the MMCs, to the design variables z'_i manipulated by the optimizer (genotype):

$$z'_i = \frac{z_i - z_i^0 + z_i^s}{2z_i^s} \text{ for } z_i \text{ corresponding to position of the MMCs: } x_{0i}, y_{0i}, z_{0i}, \quad (5.30)$$

$$z'_i = \frac{z_i - z_i^L}{z_i^U - z_i^L} \text{ for } z_i \text{ corresponding to shape and orientation of MMCs: } l_i, t_i, h_i, \theta_i, \alpha_i, \beta_i, \gamma_i, \quad (5.31)$$

where z_i^s is a suitable scaling parameter for the design variable i , and z_i^0 is the initial value of the unscaled design variable in the reference design, respectively. As defined already before, z_i^L and z_i^U , are the lower and upper bounds for the parameters. Please note that here they are used for normalization only, which does not imply that z_i cannot take values outside of the $[z_i^L, z_i^U]$ range. This can be imposed only by using the technique described in Section 5.3.4.

Equation (5.31) transforms the parameters so that $z'_i = 0$ corresponds to z_i^L and $z'_i = 1$ to z_i^U . In case of Equation (5.30), $z'_i = 0$ implies $z_i = z_i^0 - z_i^s$, $z'_i = 1$ results in $z_i = z_i^0 + z_i^s$, and $z'_i = 0.5$ corresponds to $z_i = z_i^0$. As a result, scaling factors are easier to interpret for a human. For lengths, thicknesses and rotation angles, it is usually more convenient to express the scaling in terms of the maximum and minimum values, corresponding to values of 0 and 1 for the normalized design variables z'_i . For the positional variables, one can specify the scaling factor z_i^s , which is the distance from the initial position z_i^0 at which z'_i reaches values of 0 or 1. The value of z_i^s has to be chosen in such a way that the unit change of the scaled positional variables has similar effect as the unit change of the normalized rotation angles, where usually $z_i^L = 0$ and $z_i^U = \pi$, due to the symmetry of MMCs, would be chosen.

The backward transformation takes the following form:

$$z_i = 2z_i^s z'_i + z_i^0 - z_i^s \text{ for } z_i \text{ corresponding to: } x_{0i}, y_{0i}, z_{0i}, \quad (5.32)$$

$$z_i = (z_i^U - z_i^L) z'_i + z_i^L \text{ for } z_i \text{ corresponding to: } l_i, t_i, h_i, \theta_i, \alpha_i, \beta_i, \gamma_i. \quad (5.33)$$

Based on the initial experiments, the following heuristic for defining the scaling factors was found to work well on multiple types of problems. For positional parameters, z_i^s is determined as one third of the size of the design space in a given direction, i.e. w_{dd} for x_{0i} , h_{dd} for y_{0i} , and d_{dd} for z_{0i} .

For rotation angles θ_i , α_i , β_i , and γ_i , $z_i^L = 0$ and $z_i^U = \pi$ is used. Finally, for the shape parameters l_i , t_i , and h_i , the lower bound is 0 and the upper bound is three times the initial value of the parameter from the reference design.

Of course, for specific problems, more suitable scaling factors can be found. However, satisfactory results are already obtained when no considerable differences in the influence of design variables can be observed, which is usually the case for the heuristic used in this work. The optimization algorithm can efficiently handle the rest by adapting individual step sizes for each of the design variables.

5.3.6. Symmetries

Rozvany (2011) widely discussed the problem of symmetry in structural TO. Based on the analysis of known analytical solutions and theoretical considerations, he argues that there exists at least one optimal structural layout that is symmetrical, provided that support and load conditions are symmetric. In some special cases, there might be an infinite number of other symmetric and/or nonsymmetrical optimal topologies. Although it is not clear if the same conclusions apply to crash TO, symmetry conditions can reduce the number of design variables significantly, which is even more important for evolutionary optimization methods, where the convergence velocity strongly depends on the dimensionality of the optimization problem. With this in mind, we will usually enforce symmetry conditions on the representation level in this work. One should note that this is not equivalent to using symmetry conditions in the FE simulation, which we strongly avoid. The main reason is the nonsymmetrical physical behavior of many nonlinear mechanical systems, e.g. in case of buckling. Imposing symmetry conditions in simulation would lead to an improper modeling of such physical systems.

5.4. Summary

This chapter presents a novel TO approach based on a suitable low-dimensional level-set representation and EAs – EA-LSM. Additionally, problem-specific repair operators are proposed to enhance the quality of the obtained structures as well as the convergence velocity of the algorithm. The development of the method is motivated mainly by the applications in crashworthiness, where analytical sensitivities are not available, but it has a generic character and can be potentially applied to various problems in structural mechanics.

Compared to the state-of-the-art methods for crash TO, EA-LSM does not use any heuristic assumptions or simplifications, and evaluates the performance of the design based on high-fidelity nonlinear FE crash simulations. As a result, arbitrary quantifiable optimization criteria can be taken into account, which is an important factor for the real-world applications, where frequently, e.g. injury criteria are considered as constraints or objectives (Fang et al. (2016)). In principle, any

number of constraints can be handled by the method, which is a clear advantage compared to some of the state-of-the-art approaches, such as HCA.

The main disadvantage of the proposed method is a relatively high computational cost of evolutionary optimization, leading to a considerably higher number of FE evaluations than in case of ESL or HCA approaches. Hence, whenever the assumptions of these approaches are satisfied for the problem at hand, these methods should be used in the first place. However, EA-LSM is a general alternative to these approaches. On the other hand, thanks to very good scalability of evolutionary optimizers on parallel computer architectures, the performance of EA-LSM in terms of optimization iterations (generations), representing the total time required to perform the optimization, can be still comparable to state-of-the-art methods. This is very promising in the context of growing power of the computer hardware, allowing potentially for large-scale industrial applications in the future. Nevertheless, nowadays, a single high-fidelity crash simulation can require itself a lot of parallel resources, which motivates the development of improvements of the proposed approach aiming for reduction of the total number of the FE evaluations. This important problem is therefore addressed in the following chapter.

Chapter 6

Machine learning-based enhancements of EA-LSM

Although the EA-LSM presented in Chapter 5 offers great flexibility in terms of objectives and constraints it can address, it is associated with very high computational costs compared to the state-of-the-art methods. The following chapter targets this problem by incorporating enhancements to EA-LSM, using methods from the domain of ML. In particular, this chapter proposes three approaches to achieve that, all based on an improved utilization of the information collected prior to, as well as during the optimization process:

- Based on the knowledge of the effects of parameter variations: Use an approximate gradient information of the objective function to guide the search of the evolutionary optimizer. The gradient approximations can be obtained from a simplified model of a physical system or learned from the previously sampled data.
- Based on the knowledge of design quality: Utilize surrogate models of the objective functions and constraints, which approximate computationally costly finite element simulations.
- Based on the knowledge of the effects of structural modifications: Introduce an adaptive representation of the topology, with number of design variables rising during the optimization process. The adaptation process can be based on random modifications of the design representation or a pre-trained model predicting favorable topological variations.

The first two concepts are described already in the literature, however, in contexts different to the one of this work. In contrast, the last approach represents very novel and challenging field of research.

Depending on the characteristics of the considered problem, e.g. type of objectives and constraints, number of design variables, desired complexity of the target topology, different methods can be used to improve the performance of EA-LSM. Each of the approaches mentioned above can be seen as a module that can be used separately or combined together, leading potentially to a further reduction of computational costs.

The structure of the chapter follows the order of the methods described above. Section 6.1 discusses the methods for approximation and exploitation of the gradient information in hybrid EAs. In Section 6.2, the possibilities of using surrogate models in evolutionary search are investigated. Finally, Section 6.3 proposes the Adaptive Evolutionary Level Set Method (A-EA-LSM) with learning-based topology variations, and Section 6.4 summarizes the chapter.

6.1. Exploiting approximate gradient information

Computationally efficient state-of-the-art methods for crash TO yield very good results for some of the problems. This concerns particularly the objectives involving very limited deformations of the structures, where the assumptions made by the ESL methods or HCA techniques seem to be appropriate. For instance, both ESL and HCA have demonstrated their good performance on intrusion minimization problems with low levels of plastic deformation (Bandi et al. (2013); Bujny et al. (2017a)). As a result, in such cases, one can benefit from the efficiency of the gradient-based or heuristic optimization methods, requiring considerably less FE evaluations than the evolutionary approach proposed in Chapter 5.

The availability of very efficient methods for selected problems inspired the development of hybrid optimization algorithms (Michalewicz (1996)), also frequently referred to as the memetic algorithms (Neri and Cotta (2012)), which combine EAs with a domain-specific knowledge. As shown in Figure 6.1, such a combination can be very effective, since it frequently benefits from both, the good performance of EA on a wide range of optimization problems, as well as exceptionally good properties of specialized methods for selected problems (Michalewicz (1996)). Furthermore, often the combined approach not only performs as good as the best component, but also outperforms each of the individual methods composing it (Michalewicz (1996); Bujny et al. (2016c)). On the contrary, the utilization of domain-specific knowledge creates the risk of biasing the optimization process towards certain designs, limiting the ability of the method to produce novel solutions (Eiben and Smith (2003)). As a result, a good balance between generality and performance of such a hybrid method should be found.

In the context of TO, especially the idea of combining EA with gradient-based approaches seems to be very promising (Bujny (2015); Bujny et al. (2016c, 2017b)). Gradient-based methods are the mainstream in structural TO and rely mostly on analytical derivation of the sensitivity information, which provides a formal framework for solving large-scale optimization problems. As such,

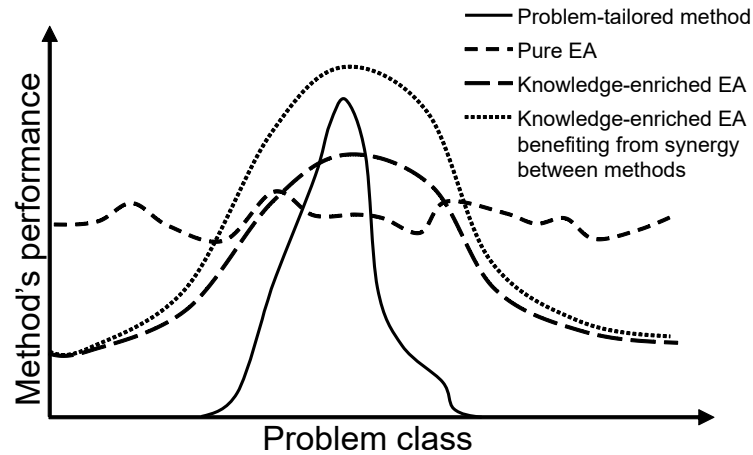


Figure 6.1 Illustration of the potential benefits coming from using hybridization techniques for optimization after Michalewicz (1996). In special cases, analytical or heuristic methods offer very good performance. On the other hand, EAs perform well in a wide range of problems. Hybridization of EA with specialized methods allows for exploitation of the benefits of both approaches. In some situations, a hybrid approach can offer a better performance than each of the methods separately.

these methods should be used in the first place whenever the analytical sensitivity information is available (Sigmund (2011)). Additionally, the availability of the analytical sensitivity information for several important problem categories in structural optimization, as well as strong focus of the research community on gradient-based approaches, justifies the choice of these methods for hybridization with EA.

As discussed above, for crash TO, in many cases, one can use ESL approach to obtain an approximation of the gradient information. However, even for the problems suitable for ESL, this could be only seen as a rough estimation of the gradient for the underlying crash problem, e.g. intrusion minimization. As a result, the hybrid approach has to take into account the uncertainty associated with the estimation of the gradient information and use the gradient information only when it is accurate enough.

For the crash problems where the gradient information cannot be reliably estimated using any of the established methods, e.g. involving large plastic deformations, one can use ML methods to learn the sensitivity information from the collected data. Such an approach for TO was first proposed by Aulig and Olhofer (2014b) and further developed later on (Aulig (2017)). In this work, we extend this approach to the level set representation relying on the concept of MMCs (Guo (2014)).

The methods described in this section are based mainly on the previous works of the author (Bujny (2015); Bujny et al. (2016c, 2017b)) as well as on the master's thesis of Lukas Krischer (Krischer (2018)) supervised by the author of this dissertation, together with Prof. Fabian Duddeck (Technical University of Munich).

6.1.1. Sensitivity analysis for estimation of approximate gradients

Let us start with a derivation of sensitivities for the compliance minimization problem, which is a basis for the approximations discussed in Sections 6.1.2 and 6.1.3. The approach described

below follows, to great extent, the analysis presented by Guo (2014). However, we propose an alternative method for numerical computation of derivatives, which turned out to give better results in the previous experiments (Bujny (2015); Bujny et al. (2016c); Krischer (2018)). Below a formal description of the method is presented.

Analytical derivation of sensitivities

Compliance minimization problem, neglecting body forces, can be defined as follows:

$$\min_{\mathbf{u} \in \mathbf{U}, \mathbf{E}} C(\mathbf{u}) = \min_{\mathbf{u} \in \mathbf{U}, \mathbf{E}} \int_{\Gamma_t} \mathbf{t} \mathbf{u} d\Gamma, \quad (6.1)$$

where \mathbf{t} denotes the boundary tractions applied on the surface Γ_t (Figure 6.2), \mathbf{U} is the space of kinematically admissible displacements \mathbf{u} , and \mathbf{E} is used to denote the stiffness tensor.

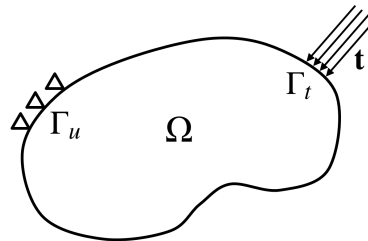


Figure 6.2 Illustration of the static problem solved in structural mechanics (Bujny et al. (2016c)).

Using the adjoint method (Bendsøe and Sigmund (2004)), together with the chain rule (Apostol (1974)), a partial derivative of compliance w.r.t. the LSF can be obtained (Zhang et al. (2017b)):

$$\frac{\partial C}{\partial \Phi} = - \iint_D \delta(\Phi(\mathbf{x})) \boldsymbol{\varepsilon}^T(\mathbf{x}) \mathbf{E}^0 \boldsymbol{\varepsilon}(\mathbf{x}) d\Omega, \quad (6.2)$$

where $\boldsymbol{\varepsilon}$ denotes the elastic strain, while $\delta(x)$ is the Dirac delta function¹ (Dirac (1930)):

$$\delta(x) = \begin{cases} +\infty, & \text{if } x = 0 \\ 0, & \text{if } x \neq 0, \end{cases} \quad (6.3)$$

satisfying the following property:

$$\int_{-\infty}^{+\infty} \delta(x) dx = 1. \quad (6.4)$$

¹ Dirac delta is not a function in the classical sense, since no function defined on the real number has the properties given below (Dirac (1930)). Formally, Dirac delta can be defined as a distribution (generalized function) (Hörmander (2003)).

The derivatives of the global LSF Φ w.r.t. any of the design variables, i.e. x_{0i} , y_{0i} , θ_i , l_i , t_i , can be expressed in the following form:

$$\frac{\partial \Phi}{\partial p_i}(\mathbf{x}) = \begin{cases} \frac{\partial \phi_i}{\partial p_i}(\mathbf{x}), & \text{if } \Phi(\mathbf{x}) = \phi_i(\mathbf{x}) \\ 0, & \text{otherwise,} \end{cases} \quad (6.5)$$

where p_i , for generality, is used to denote any type of design variables listed above. The partial derivative $\frac{\partial \phi_i}{\partial p_i}$ is easy to derive analytically based on the Equation (5.4):

$$\begin{aligned} \frac{\partial \phi_i}{\partial x_{0i}} = & q \left(\frac{\cos \theta_i (x - x_{0i}) + \sin \theta_i (y - y_{0i})}{l_i/2} \right)^{q-1} \frac{\cos \theta_i}{l_i/2} \\ & + q \left(\frac{-\sin \theta_i (x - x_{0i}) + \cos \theta_i (y - y_{0i})}{t_i/2} \right)^{q-1} \frac{-\sin \theta_i}{t_i/2}, \end{aligned} \quad (6.6)$$

$$\begin{aligned} \frac{\partial \phi_i}{\partial y_{0i}} = & q \left(\frac{\cos \theta_i (x - x_{0i}) + \sin \theta_i (y - y_{0i})}{l_i/2} \right)^{q-1} \frac{\sin \theta_i}{l_i/2} \\ & + q \left(\frac{-\sin \theta_i (x - x_{0i}) + \cos \theta_i (y - y_{0i})}{t_i/2} \right)^{q-1} \frac{\cos \theta_i}{t_i/2}, \end{aligned} \quad (6.7)$$

$$\begin{aligned} \frac{\partial \phi_i}{\partial \theta_i} = & q \left(\frac{\cos \theta_i (x - x_{0i}) + \sin \theta_i (y - y_{0i})}{l_i/2} \right)^{q-1} \left(\frac{\sin \theta_i (x - x_{0i}) - \cos \theta_i (y - y_{0i})}{l_i/2} \right) \\ & + q \left(\frac{-\sin \theta_i (x - x_{0i}) + \cos \theta_i (y - y_{0i})}{t_i/2} \right)^{q-1} \left(\frac{\cos \theta_i (x - x_{0i}) + \sin \theta_i (y - y_{0i})}{t_i/2} \right), \end{aligned} \quad (6.8)$$

$$\frac{\partial \phi_i}{\partial l_i} = \frac{1}{2} q \left(\frac{\cos \theta_i (x - x_{0i}) + \sin \theta_i (y - y_{0i})}{l_i/2} \right)^{q-1} \frac{\cos \theta_i (x - x_{0i}) + \sin \theta_i (y - y_{0i})}{(l_i/2)^2}, \quad (6.9)$$

$$\frac{\partial \phi_i}{\partial t_i} = \frac{1}{2} q \left(\frac{-\sin \theta_i (x - x_{0i}) + \cos \theta_i (y - y_{0i})}{t_i/2} \right)^{q-1} \frac{-\sin \theta_i (x - x_{0i}) + \cos \theta_i (y - y_{0i})}{(t_i/2)^2}, \quad (6.10)$$

Figure 6.3 shows the plots of Equations (6.6)–(6.10) normalized with the Euclidean norm of the gradient of the LSF, which, as shown later in this section, have a very intuitive interpretation as

weighting factors for the element strain energies. This property will be also used in Section 6.1.3 for constructing features for the ML models.

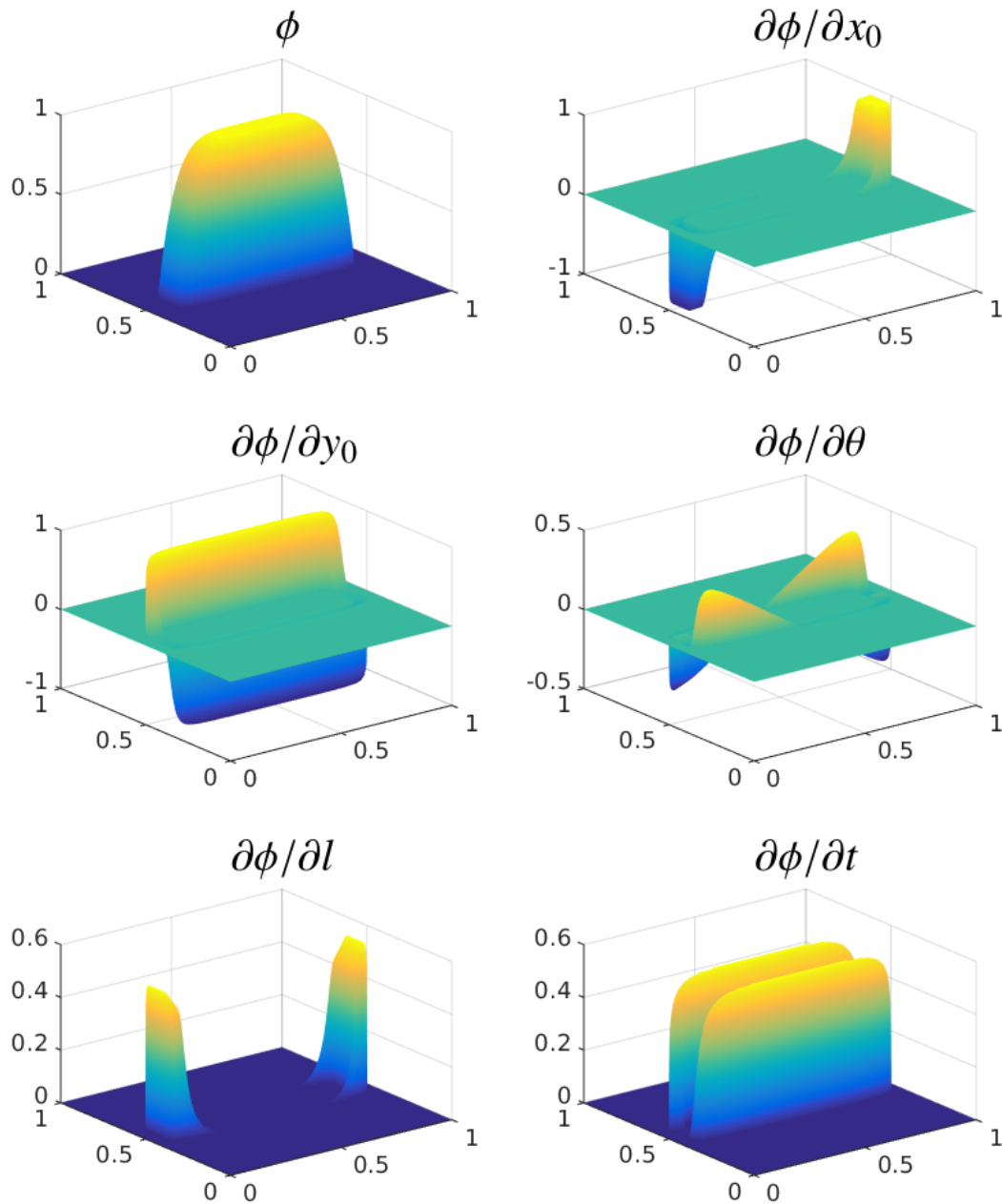


Figure 6.3 Local LSF ϕ and its derivatives w.r.t. the design variables (Bujny (2015); Bujny et al. (2016c)). Only the values in a close neighborhood of the material interface are shown (the rest is set to zero). The partial derivatives are divided by the Euclidean norm of the gradient of the LSF ($\nabla\phi = \left[\frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y} \right]$).

Finally, the analytical sensitivity of the compliance w.r.t. any of the parameters of the i^{th} MMC is

given by:

$$\frac{\partial C}{\partial p_i} = - \iint_D \delta(\Phi(\mathbf{x})) \frac{\partial \Phi}{\partial p_i}(\mathbf{x}) \boldsymbol{\varepsilon}^T(\mathbf{x}) \mathbf{E}^0 \boldsymbol{\varepsilon}(\mathbf{x}) d\Omega. \quad (6.11)$$

Similarly to the approach described above, one can derive the analytical sensitivities of the other criteria taken as objectives or constraints in the optimization problem. In particular, volume constraint frequently plays an important role in TO. The total volume of the material phase in the design domain D for a 2D case is given by:

$$V = \iint_D H(\Phi(\mathbf{x})) d\Omega. \quad (6.12)$$

Consequently, the sensitivity of the volume w.r.t. design variable p_i is equal to:

$$\frac{\partial V}{\partial p_i} = \iint_D \delta(\Phi(\mathbf{x})) \frac{\partial \Phi}{\partial p_i}(\mathbf{x}) d\Omega. \quad (6.13)$$

Numerical computation of sensitivities

Due to the discrete nature of the formulations used in the computer simulations, e.g. with FEM, the equations derived above cannot be used directly for the computation of sensitivities. Instead, one has to approximate them based on discrete mesh of nodal displacements, used to calculate other quantities, such as stresses or element strain energies.

Due to the presence of the Dirac delta in the equations above, the numerical computation of the integrals is challenging. The easiest approach relies on the introduction of an approximate Heaviside function with continuous derivative, which can be treated as an approximation of the Dirac delta (Belytschko et al. (2003)). However, this approach turned out to result in low accuracy of gradient approximations for the level-set parametrization used in this work (Bujny (2015); Bujny et al. (2016c); Krischer (2018)). As a result, we propose an alternative approach for numerical estimation of the gradients, which is easy and offers good accuracy according to the previously conducted experiments (Bujny (2015); Krischer (2018)).

Analytical sensitivities derived above have the following form²:

$$\xi = \iint_{\mathbb{R}^2} \delta(\Phi(x,y)) \mathcal{F}(x,y) d\Omega \quad (6.14)$$

where $\mathbf{x} = [x, y]^T$ denotes the spatial location, while \mathcal{F} is an arbitrary function of those variables.

² Please note that the derivations are done for a 2D problem formulation, but can be easily generalized to 3D cases.

After change of variables, the integral ξ can be rewritten as (Hörmander (2003)):

$$\xi = \int_0^\infty \int_{K:\Phi=t} \frac{\delta(t)}{|\nabla\Phi(x(l,t),y(l,t))|} \mathcal{F}(x(l,t),y(l,t)) dl dt. \quad (6.15)$$

Since the Dirac delta term takes nonzero values only for $\Phi = t = 0$, Equation (6.15) can be rewritten as follows:

$$\xi = \int_{K:\Phi=0} \frac{\mathcal{F}(x(l),y(l))}{|\nabla\Phi(x(l),y(l))|} dl. \quad (6.16)$$

Technically, the integral above can be numerically estimated by finding all of the elements crossed by the 0th iso-contour of the LSF. After finding those elements, the expression $\mathcal{F}(x,y)/|\nabla\Phi(x,y)|$ can be calculated in the center of each element and multiplied by an approximate length of the material-void interface in the element. A summation of these values yields an efficient approximation of Equation (6.14):

$$\xi \approx \sum_{e \in G} \frac{\mathcal{F}(x_{C_e}, y_{C_e})}{|\nabla\Phi(x_{C_e}, y_{C_e})|} l_e, \quad (6.17)$$

where G denotes the set of indices of the elements crossed by the 0th level-set iso-contour, while (x_{C_e}, y_{C_e}) are the coordinates of the element's center and l_e denotes the length of the material-void interface inside the element.

Equation (6.17) can be directly used to compute the analytical sensitivities derived in the previous subsection (Equations (6.11) and (6.13)).

As mentioned before, one can easily note that the term $\frac{\partial\Phi}{\partial p_i}(x,y)/|\nabla\Phi(x,y)|$, where p_i denotes a design variable parametrizing an MMC (e.g. x_{0_i}, l_i), has a very intuitive interpretation. A comparison of Equations (6.11) and (6.17), shows that $\frac{\partial\Phi}{\partial p_i}(x,y)/|\nabla\Phi(x,y)|$ acts as a weighting factor in the summation of the element strain energies ($\frac{1}{2}\boldsymbol{\epsilon}^T \mathbf{E}^0 \boldsymbol{\epsilon}$). For example, when numerically calculating the derivative of compliance w.r.t. the parameter x_{0_i} , one would multiply strain energies on the left end of the MMC by -1 , while multiplying the energies on the right-hand side by a weighting factor equal to 1 (Figure 6.3). As a consequence, the sign of the derivative $\frac{\partial C}{\partial x_{0_i}}$ will depend on the ratio of magnitudes of strain energies at the MMC's ends. One should note that the strain energies of the elements located at the part of the 0th iso-contour parallel to the x -axis do not contribute to the partial derivative $\frac{\partial C}{\partial x_{0_i}}$. In contrast, for the partial derivative $\frac{\partial C}{\partial y_{0_i}}$, the situation is exactly the opposite.

The simplifications in the numerical estimation of Equation (6.15) described above do not guarantee high accuracy of gradient approximation, especially for very coarse meshes. However, the

experimental evaluation of the gradient estimates carried out before (Bujny (2015)) has confirmed satisfactory gradient estimation accuracy of this method for different mesh resolutions. In particular, in this work, the sensitivities of the compliance are used exclusively as an approximation of the gradients for crashworthiness objectives, e.g. intrusion, energy absorption, so the accuracy of the compliance sensitivity itself becomes a secondary factor.

6.1.2. Gradient approximation via global equivalent static loads method

The global ESL method, sometimes referred also to as replacement loading (Wehrle et al. (2015)), belongs to the oldest approaches used in crashworthiness TO (Cavazzuti et al. (2010); Christensen et al. (2012)), and is often used also in the engineering practice. The method assumes that the dynamic crash loads can be replaced with static loads, which do not change over time. As a consequence, one can use well-established gradient-based methods for TO of linear elastic problems (Bendsøe and Sigmund (2004)). Global ESL is therefore appropriate for optimization problems involving limited plastic deformations, e.g. design of safety cells (Duddeck et al. (2016)). Consequently, ESL can be seen as a specialized approach, which can be used for enhancing the evolutionary search. One of the potential realizations of this idea is described below.

In a first step, the equivalent static case consistent with the dynamic crash scenario has to be defined. To a great extent, this relies on expert knowledge and is defined manually. Usually, the load definition covers a wide area of the vehicle design space and represents roughly the time-dependent loading conditions. As an example, consider the 2D test case used in this work (Figure 6.4(a)) and its equivalent static case (Figure 6.4(b)). The equivalent case represents roughly the crash load just with a single point load. Of course, more detailed equivalent static modeling, accounting for contact as well as large deformations and changing loading conditions in the crash scenario, would be also possible in this case. However, the modeling accuracy is not of the main interest in this work, which targets evaluation of feasibility of using evolutionary optimization enhanced with an approximate gradient information in crash TO.

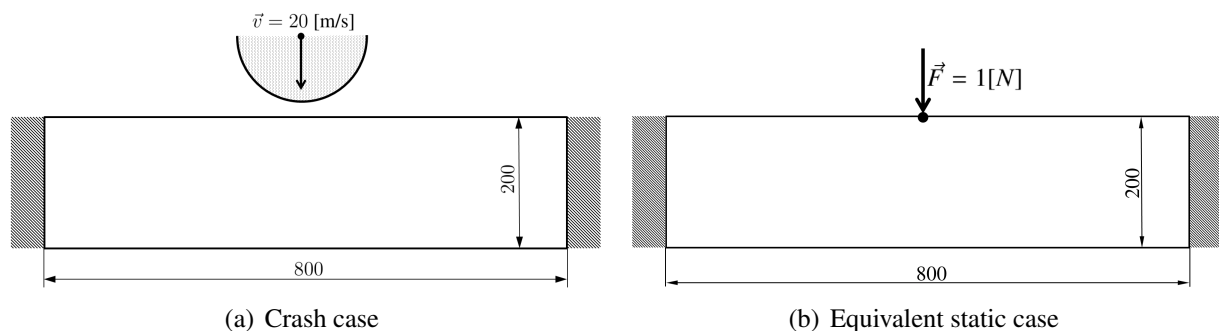


Figure 6.4 Example of a crash case and the corresponding equivalent static case.

Once the equivalent static case is defined, the gradient information can be obtained by performing a static FE analysis and using the approach for numerical computation of sensitivities described in Section 6.1.1. The gradient of compliance and volume can be used then directly in the approach

presented in Section 6.1.4 as an approximation of sensitivities related to crash performance, e.g. intrusion or energy absorption.

6.1.3. Gradient approximation by predicting sensitivities

Estimation of sensitivities described in Section 6.1.2 is limited only to the cases when the optimization objective exhibits some similarity to the compliance. A potential solution to overcome this problem is to use ML to model the relationships between the structural state of the design under crash loading conditions and the sensitivities of arbitrary responses. This idea was first proposed by Aulig and Olhofer (2014b) and used for the case of the standard density-based TO of linear elastic structures. Later on, this approach was also evaluated in TO of crash problems (Aulig (2017)). This section extends this idea to TO using the MMC framework. Furthermore, in this work, we do not rely on the correctness of the sensitivity models, but use it as an approximate gradient information in the hybrid approach proposed in Section 6.1.4.

Similarly to the approach introduced by Aulig and Olhofer (2014b), the method described in this work relies on the assumption that the gradient of the cost function ∇f can be predicted based on the structural state of the design, evaluated using FEM. In particular, Aulig and Olhofer (2014b) used a standard grid representation of the topology and assumed that the sensitivity of the objective function w.r.t. the element density ρ_e can be predicted exclusively based on the state features of the considered element, e.g. its strain energy, stress or nodal displacements. For that reason, the features used for building sensitivity models by Aulig and Olhofer (2014b) were referred to as the local state features.

Following a similar logic, in case of the MMC-based representation used in this work, one could assume that the sensitivity of the objective function w.r.t. parameters describing an MMC, i.e. x -position x_0 , length l , rotation angle θ , etc., can be predicted with use of the state features related to the finite elements located spatially within the considered MMC. Indeed, in the special case of compliance being the objective function for an optimization problem involving linear elastic cases, the analytical sensitivities depend only on the strain energies of elements located within a given MMC. However, instead of using the local state features to model the sensitivities w.r.t. element densities and finding relationships between them and the sensitivities w.r.t. parameters of MMCs, in this work, we construct compound Component State Features (CSFs) (Krischer (2018)) related directly to the state of the entire MMC.

By observing the similarity of the term $\boldsymbol{\varepsilon}^T(\mathbf{x}) \mathbf{E}^0 \boldsymbol{\varepsilon}(\mathbf{x})$ in the Equation (6.11) to the compliance sensitivities $\left(\frac{\partial C}{\partial \rho_e}\right)$ in the density-based TO methods (Bendsøe and Sigmund (2004)), this work assumes that the sensitivity of any objective function w.r.t. the parameters of an MMC can be modeled with use of CSFs of the following form:

$$CSF = - \iint_D \delta(\Phi(\mathbf{x})) \frac{\partial \Phi}{\partial p_i}(\mathbf{x}) S(\mathbf{x}) d\Omega, \quad (6.18)$$

where $S(\mathbf{x})$ can be an expression combining any quantities describing the state of the structure at location \mathbf{x} (local state features). In particular, $S(\mathbf{x}) = \boldsymbol{\varepsilon}^T(\mathbf{x}) \mathbf{E}^0 \boldsymbol{\varepsilon}(\mathbf{x})$ would yield a CSF identical to the analytical sensitivities of compliance, given by Equation (6.11).

Following the derivation of the numerical scheme for estimation of sensitivities of the form given by Equation (6.14), described in Section 6.1.1, CSFs can be computed numerically as follows:

$$CSF \approx \sum_{e \in G} \frac{\frac{\partial \Phi}{\partial p_i}(x_{C_e}, y_{C_e}) S_e}{|\nabla \Phi(x_{C_e}, y_{C_e})|} l_e, \quad (6.19)$$

where S_e refers to the local state features at the finite element e .

In general, S_e can be any combination of local state features describing the state of a given finite element, which gives a possibility to handcraft features most suitable for a given problem. However, hand-crafting the features either requires certain prior knowledge about the function being modeled or involves a cumbersome trial-and-error process to obtain the best model. From this perspective, nodal displacements seem to be the most general features, whose transformations can yield other quantities. Due to the generic character of such features, this work considers CSFs exclusively based on displacements and their interactions. As a result, CSFs of the following forms will be considered in this work:

$$CSF \approx \sum_{e \in G} \frac{\frac{\partial \Phi}{\partial p_i}(x_{C_e}, y_{C_e}) u_{e,j}}{|\nabla \Phi(x_{C_e}, y_{C_e})|} l_e, \quad (6.20)$$

and:

$$CSF \approx \sum_{e \in G} \frac{\frac{\partial \Phi}{\partial p_i}(x_{C_e}, y_{C_e}) u_{e,j} u_{e,k}}{|\nabla \Phi(x_{C_e}, y_{C_e})|} l_e, \quad (6.21)$$

where $u_{e,j}$ is the j^{th} nodal displacement of the e^{th} finite element, as shown in Figure 6.5. For 2D cases using the standard four-node shell elements or eight-node solid elements (assuming the same displacement magnitudes of the corresponding degrees of freedom), we have the node index $j = 1, 2, \dots, 8$ and $k = 1, 2, \dots, 8$.

Finally, the CSFs are used as the independent variables in the models of the partial derivatives of the considered objective function. In contrast to the work of Aulig and Olhofer (2014b), we propose to use independent models for each of the parameters defining an MMC:

$$\frac{\partial f}{\partial p_i} \approx \left(\frac{\partial f}{\partial p} \right)_{\boldsymbol{\Theta}_p} (\mathbf{CSF}_i), \quad (6.22)$$

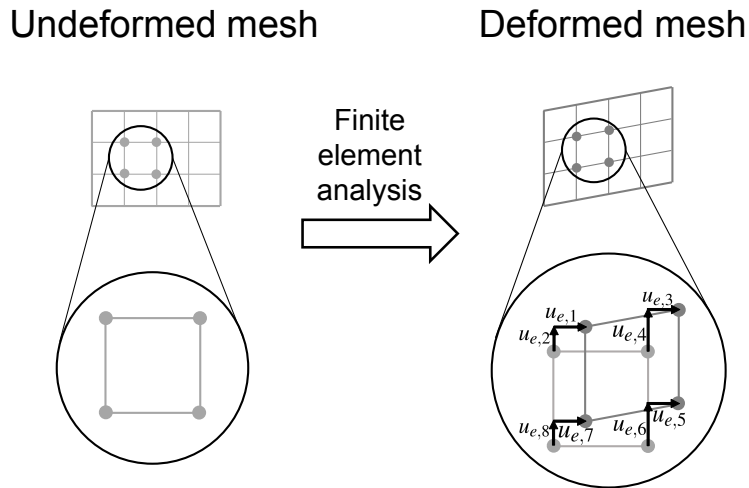


Figure 6.5 Naming convention of the displacements in four-node shell elements or eight-node solid elements (top view) used for computation of the CSFs.

where \mathbf{CSF}_i is a vector of different CSFs computed for the i^{th} MMC and used in the sensitivity model $\left(\frac{\partial f}{\partial p}\right)_{\Theta_p}$, with a vector of parameters Θ_p . For a two-dimensional MMC parametrization, this would correspond to five independent models for $p = x_0, y_0, \theta, l$, and t .

In general, different ML regression models (Bishop (2007)), e.g. Linear Regression (LR), Support Vector Regression (SVR), Gaussian Processes (GP), Artificial Neural Networks (ANN), can be used to model the sensitivities. The initial studies (Krischer (2018)) showed that, even for the nonlinear crash cases, a satisfactory prediction accuracy can be obtained already with simple LR models. Especially when the number of training samples is limited, LR is a good choice, due to a relatively low number of trainable parameters, which helps to avoid overfitting (Bishop (2007)). Therefore, in this work, we narrow down the discussion to the linear models only.

The training samples for the ML models are obtained via numerical estimation of sensitivities using the Finite Difference (FD) method (Iserles (2008)). In particular, we focus on the central finite differences, offering higher accuracy than forward or backward differences (Iserles (2008)). For a set of different designs, which can be generated through Design of Experiments (DoE) or obtained from intermediate steps during an optimization, each of the design variables is varied by a small value ζ . The formula for estimation of the sensitivity of the objective function w.r.t. a design variable z_j takes then the following form:

$$\frac{\partial f}{\partial z_j} \approx \frac{f(\mathbf{z}_{j+}) - f(\mathbf{z}_{j-})}{2\zeta}, \quad (6.23)$$

with:

$$\mathbf{z}_{j+} = [z_1, z_2, \dots, z_j + \zeta, \dots, z_n]^T, \quad (6.24)$$

and:

$$\mathbf{z}_{j-} = [z_1, z_2, \dots, z_j - \zeta, \dots, z_n]^T. \quad (6.25)$$

Since each design variable is one of the five parameters describing an MMC (assuming a 2D case), computing finite differences for all of the design variables in a single design results in $\frac{n}{5} = M$ samples of the sensitivities for each MMC parameter p .

In order to better illustrate the sampling process, let us consider modeling a sensitivity of a function f w.r.t. the rotation angle θ , for a design with 16 MMCs. For the i^{th} MMC, the objective function is evaluated for a rotation angle $\theta_i + \zeta$ and $\theta_i - \zeta$, with all the other design variables remaining unchanged. Based on that, the central finite difference is calculated according to Equation (6.23). This yields the first training sample for the sensitivity model $\left(\frac{\widetilde{\partial f}}{\partial \theta}\right)_{\Theta_\theta}$. This process is carried out for all $i = 1, \dots, 16$ MMCs, yielding 16 samples for training the model $\left(\frac{\widetilde{\partial f}}{\partial \theta}\right)_{\Theta_\theta}$. By repeating that process for the new designs, involving different configurations of MMCs, sufficient number of samples for training a regression model of $\frac{\partial f}{\partial \theta}$ can be collected. Similarly, data samples for the models of $\frac{\partial f}{\partial x_0}$, $\frac{\partial f}{\partial y_0}$, $\frac{\partial f}{\partial t}$, and $\frac{\partial f}{\partial l}$ can be gathered. Finally, five ML models: $\left(\frac{\widetilde{\partial f}}{\partial x_0}\right)_{\Theta_{x_0}}$, $\left(\frac{\widetilde{\partial f}}{\partial y_0}\right)_{\Theta_{y_0}}$, $\left(\frac{\widetilde{\partial f}}{\partial \theta}\right)_{\Theta_\theta}$, $\left(\frac{\widetilde{\partial f}}{\partial l}\right)_{\Theta_l}$, $\left(\frac{\widetilde{\partial f}}{\partial t}\right)_{\Theta_t}$, can be trained based on the created data sets, resulting in a gradient approximation:

$$\begin{aligned} \widetilde{\nabla} f(\mathbf{z}) = & \left[\left(\frac{\widetilde{\partial f}}{\partial x_0}\right)_{\Theta_{x_0}} (\mathbf{CSF}_1), \left(\frac{\widetilde{\partial f}}{\partial y_0}\right)_{\Theta_{y_0}} (\mathbf{CSF}_1), \left(\frac{\widetilde{\partial f}}{\partial \theta}\right)_{\Theta_\theta} (\mathbf{CSF}_1), \left(\frac{\widetilde{\partial f}}{\partial l}\right)_{\Theta_l} (\mathbf{CSF}_1), \right. \\ & \left. \left(\frac{\widetilde{\partial f}}{\partial t}\right)_{\Theta_t} (\mathbf{CSF}_1), \dots, \left(\frac{\widetilde{\partial f}}{\partial x_0}\right)_{\Theta_{x_0}} (\mathbf{CSF}_M), \left(\frac{\widetilde{\partial f}}{\partial y_0}\right)_{\Theta_{y_0}} (\mathbf{CSF}_M), \right. \\ & \left. \left(\frac{\widetilde{\partial f}}{\partial \theta}\right)_{\Theta_\theta} (\mathbf{CSF}_M), \left(\frac{\widetilde{\partial f}}{\partial l}\right)_{\Theta_l} (\mathbf{CSF}_M), \left(\frac{\widetilde{\partial f}}{\partial t}\right)_{\Theta_t} (\mathbf{CSF}_M) \right]^T \end{aligned} \quad (6.26)$$

The entire process is summarized in Figure 6.6.

The obtained models could be, in principle, used directly in a gradient-based optimization, as in the work of Aulig and Olhofer (2014b). In fact, using the finite differences as training samples for a regression model rather than utilizing them directly in a gradient-based algorithm, allows for reduction of the problem of the high noisiness of such numerical estimates, appearing usually in crash simulations (Duddeck (2008); Wehrle (2015)). Taking into account limited accuracy of such models, similarly to the approach described in Section 6.1.2, we consider them rather as an approximation to enhance the hybrid methods discussed in the next section.

To summarize, the key advantage of the method discussed in this section, compared to the ap-

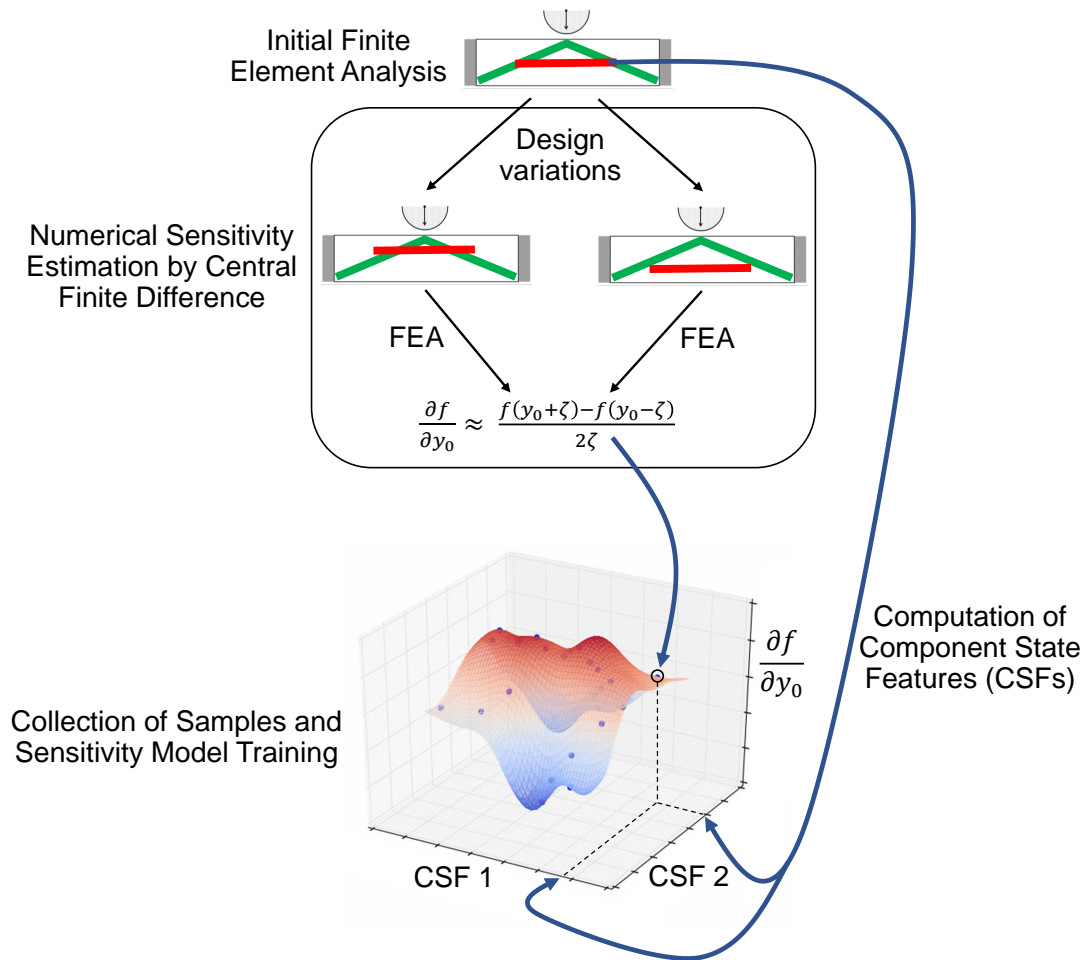


Figure 6.6 Illustration of the sampling process via finite-differencing to generate a dataset for training ML models of sensitivities.

proaches described previously, is its generic character. Potentially, the same method can be used to learn sensitivities of arbitrary quantifiable objectives and constraints. Although the initial cost of creating the data set of sensitivities via finite differencing can be high, once the model is trained on a sufficiently diverse set of designs, it can be used multiple times in different optimization cases.

6.1.4. Hybrid evolutionary algorithms

In the previous works (Bujny (2015); Bujny et al. (2016c, 2017b)) we proposed a set of hybrid approaches, relying on different concepts for combining evolutionary optimization with gradient-based search:

- Global Hybrid Evolution Strategy with Gradient-based Local Search (GHES), updating a part of the offspring population using gradient information.
- Concurrent Hybrid Evolution Strategy with Gradient Individual (CHES), applying gradient-based optimization to the best individual, while optimizing the rest of the population using standard ES.
- Hybrid CMA-ES Method (HCMA-ES), updating the mean of the normal distribution used in

the mutation operator of CMA-ES according to the gradient information.

Among the methods mentioned above, GHES turned out to be the most efficient approach for the cases when the gradient information was inaccurate (Bujny (2015); Bujny et al. (2017b)), as it is often in case of crashworthiness optimization (Duddeck (2008)). In these cases, the method is able to preserve convergence velocity of the ES composing it, while going beyond the performance of gradient-based or evolutionary search for the cases when the sensitivity information is accurate enough (Bujny (2015); Bujny et al. (2016c, 2017b)). On the other hand, CHES performs only as good as its best component, while any significant inaccuracies of the gradient information tend to deteriorate the search of HCMA-ES. As a result, in this work, the discussion is narrowed down exclusively to GHES. From now on, the TO method using the level set representation discussed in Chapter 5 together with the GHES optimizer will be referred to as the Hybrid Evolutionary Level Set Method (H-EA-LSM).

The H-EA-LSM algorithm can be summarized as follows (Algorithm 2):

```

iter := 0;
initialize  $\mathcal{P}(0) := \{\mathbf{a}_1(0), \dots, \mathbf{a}_\mu(0)\} \in \mathbb{I}^\mu$ ;
evaluate  $\mathcal{P}(0) : \{f(\mathbf{a}_1(0)), \dots, f(\mathbf{a}_\mu(0))\}$ ;
while iter < itermax do
    recombine:  $\mathcal{P}'(iter) := r_{\Theta_r}(\mathcal{P}(iter))$ ;
    mutate:  $\mathcal{P}''(iter) := m_{\Theta_m}(\mathcal{P}'(iter))$ ;
    evaluate  $\mathcal{P}''(iter) : \{f(\mathbf{a}_1''(iter)), \dots, f(\mathbf{a}_\lambda''(iter))\}$ ;
    improve:  $\mathcal{P}'''(iter) := i_{\Theta_i}(\mathcal{P}''(iter))$ ;
    evaluate  $\mathcal{P}'''(iter) : \{f(\mathbf{a}_1'''(iter)), \dots, f(\mathbf{a}_\lambda'''(iter))\}$ ;
    select:  $\mathcal{P}(iter+1) := s_{\Theta_s}(\mathcal{P}'''(iter))$ ;
    iter := iter + 1;
end

```

Algorithm 2: Pseudocode of the gradient-enhanced H-EA-LSM used in this work.

The naming convention used in Algorithm 2 follows the one introduced in Chapter 3. The main difference between Algorithm 2 and the standard ES (Chapter 3) (Bäck and Schwefel (1993)) is the introduction of the improvement operator $i_{\Theta_i} : \mathbb{I}^\lambda \rightarrow \mathbb{I}^\lambda$ with a set of parameters Θ_i , inspired by the work of Eiben and Smith (2003).

In the improvement step, a part of the offspring population is selected for improvement with an approximate gradient information obtained using one of the methods described in Sections 6.1.2–6.1.3. Usually, only a small fraction (10%–30%) of the offspring population is sufficient for obtaining good results (Bujny (2015); Bujny et al. (2016c, 2017b)). In this work, the selection of the

individuals for the improvement is done randomly. In fact, other selection rules for the improvement, e.g. certain number of the best individuals, did not yield better results (Bujny (2015)). The improvement is realized as a single Steepest Descent (SD) step (Rao (2009)):

$$\mathbf{z} := \mathbf{z} - s\widetilde{\nabla}f(\mathbf{z}), \quad (6.27)$$

where s is the gradient step size, which can be a fixed value or determined at each iteration by means of the line search, e.g. using the quadratic interpolation method (Arora (2012)). We use $\widetilde{\nabla}f(\mathbf{z})$ to denote the gradient approximation of the cost function w.r.t. the vector of design variables \mathbf{z} .

After the improvement step, the algorithm continues by evaluating the improved individuals and selecting the best individuals out of the offspring population, including both improved and not improved individuals, to form the next parent population (non-elitist selection (Bäck (1996))). As a result, the algorithm is able to reject the improved individuals when the gradient approximation is not accurate enough, using exclusively the standard selection operator. This does not deteriorate the performance of the algorithm, unless the fraction of the individuals influenced by the wrong gradient prediction is considerable, which would reduce the effective offspring population size. On the other hand, it turns out that frequently the application of the mutation operator prior to the improvement step can considerably increase the convergence velocity due to the fact that the mutation provides better starting points for the gradient improvements. Taking into account the facts mentioned above, it becomes clear that the random strategy for selection of the individuals for improvement can lead to better results, since initially worse individuals can be much better after the joint mutation-improvement step. At the same time, a part of the best individuals after mutation would still remain in the population and would not be deteriorated in case of inaccurate gradient approximation.

After the selection step, the entire process is repeated until the maximal number of generations $iter_{max}$ is reached.

6.2. Utilizing surrogate modeling techniques

Surrogate modeling techniques (Forrester et al. (2008)) have proven to be a valuable tool particularly in optimization problems involving costly simulations (Fang et al. (2005); Lee and Kang (2007); Duddeck (2008); Yoshimura et al. (2016)). Especially when the number of design variables is low, the objectives and constraints can be efficiently approximated based on relatively low number of samples, where each sample involves one costly simulation. For optimization problems of higher dimensionality, the use of surrogate modeling techniques, in particular Kriging (Krige (1951)) within the EGO approach (Jones et al. (1998)), becomes prohibitive due the curse of dimensionality (Bishop (2007); Forrester et al. (2008)). Because of the high dimensionality of

structural TO problems using standard grid representation, surrogate modeling techniques have not been applied in this field before. Thanks to the utilization of the low-dimensional level set representation presented in Chapter 5, this became feasible also for structural TO (Raponi (2017); Raponi et al. (2017, 2019a)), establishing a novel non-gradient TO approach. However, even with this representation, one should note that the number of design variables should be kept on a low level, to use the method efficiently. Therefore, the method discussed in this section is meant to be applied mainly in the situations when the desired complexity of the target design is low, while the finite element simulations are very costly.

The approach described in this section was developed within a master's thesis of Elena Raponi at the Honda Research Institute Europe in collaboration with the Technical University of Munich and University of Camerino (Raponi (2017)) and presented in research papers (Raponi et al. (2017, 2019a)). The master's thesis was supervised by the author of this dissertation, together with Dr. Markus Olhofer, Prof. Fabian Duddeck, and Prof. Simonetta Boria.

6.2.1. Kriging-guided level set method (KG-LSM)

Given a level-set parametrization of the design as described in Chapter 5, Section 5.2, the proposed optimization method follows the EGO (Jones et al. (1998)) approach with modifications crafted for solving efficiently structural TO problems. For consistency of the description of the proposed method, we introduce elements of the EGO approach in this section, instead of doing that in the prior chapters. We follow a usual EGO (Forrester et al. (2008)), using Kriging (Krige (1951); Sacks et al. (1989)) as the underlying surrogate model, which showed also the highest accuracy in our initial numerical experiments (Raponi (2017)). As in the standard EGO, the method starts with a Design of Experiments (DoE), where the design vectors are sampled and evaluated based on a high-fidelity FE simulation model. Among different DoE methods, Optimal Latin Hypercube Sampling (OLHS) has demonstrated its good properties as an initial sampling for EGO (Forrester et al. (2008)) and is used in this work. OLHS distributes the points in the design space in such a way that projection of the points onto an axis of each design variable is uniform (as in the standard Latin Hypercube Sampling (LHS)), while the distances between the points in the design space are maximized (Forrester et al. (2008)). As a result, even if one of the design variables has very little influence on the objective functions or constraints, the information from the generated samples can be used to efficiently model the relationships between the considered responses and the other design variables. As illustrated in Figure 6.7, this information would be wasted in a Full Factorial Sampling (FFS), with computational loss exponential in the number of non-influential design variables.

First of all, a set of n_s vectors of design variables $\mathbf{Z} = [\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots, \mathbf{z}^{(n_s)}]^T$ is generated using OLHS. Subsequently, for each of the vectors $\mathbf{z}^{(i)}$ an FE model is created and evaluated in a simulation. Based on the data from the simulation, corresponding sets of responses, being objectives and constraints, for $\mathbf{r}_k = [r_k^{(1)}, r_k^{(2)}, \dots, r_k^{(n_s)}]^T$, $k = 1, \dots, n_{ineq} + 1$ are collected, where n_{ineq} is the

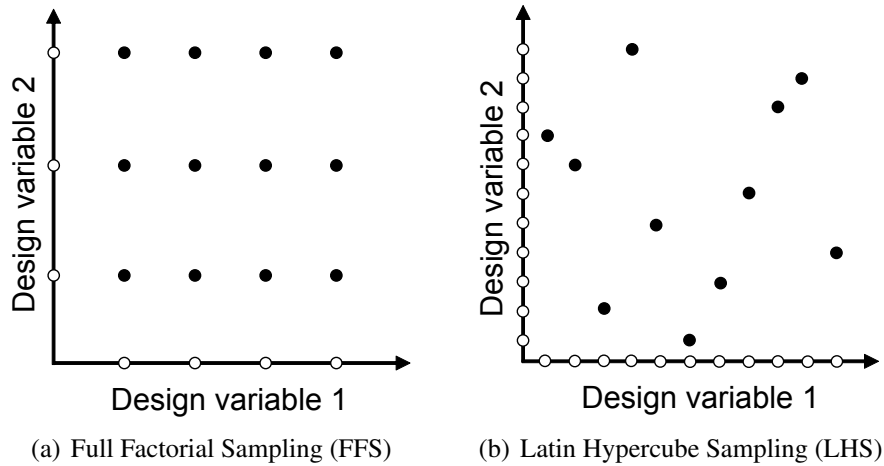


Figure 6.7 Comparison between the FFS and LHS. Even if one design variable has very low influence on the objective function, thanks to the uniform projections onto each axis of the design space, LHS provides more information for the influential dimensions than the FFS.

number of constraints³. To approximate each of the responses, $n_{ineq} + 1$ Kriging models (Krige (1951); Sacks et al. (1989); Forrester et al. (2008)) are trained based on the provided input (\mathbf{Z}) and target (\mathbf{r}_k) data.

Each Kriging model treats the samples of the responses (\mathbf{r}_k) as a result of a stochastic process. This can be described by using a set of random vectors:

$$\mathbf{R}_k = \begin{bmatrix} R_k(\mathbf{z}^{(1)}) \\ \vdots \\ R_k(\mathbf{z}^{(n_s)}) \end{bmatrix}, \quad (6.28)$$

with a mean $\mathbf{1}\mu_{s_k}$, $\mathbf{1}$ being an n_s by 1 vector of ones. The random variables are related to each other by basis functions of the form:

$$\text{cor}[R_k(\mathbf{z}^{(i)}), R_k(\mathbf{z}^{(l)})] = \exp\left(-\sum_{j=1}^{n_s} \eta_{kj} |z_j^{(i)} - z_j^{(l)}|^{d_{kj}}\right), \quad (6.29)$$

where the parameters η_{kj} and d_{kj} control the width of the basis function and smoothness of the approximation in the neighborhood of the given samples, respectively.

³ Without loss of generality, a single-objective optimization problem with inequality constraints only is assumed here.

Based on Equation (6.29), the following correlation matrix can be defined:

$$\mathbf{\Psi}_k = \begin{bmatrix} \text{cor}[R_k(\mathbf{z}^{(1)}), R_k(\mathbf{z}^{(1)})] & \cdots & \text{cor}[R_k(\mathbf{z}^{(1)}), R_k(\mathbf{z}^{(n_s)})] \\ \vdots & \ddots & \vdots \\ \text{cor}[R_k(\mathbf{z}^{(n_s)}), R_k(\mathbf{z}^{(1)})] & \cdots & \text{cor}[R_k(\mathbf{z}^{(n_s)}), R_k(\mathbf{z}^{(n_s)})] \end{bmatrix}, \quad (6.30)$$

which is related to the covariance matrix by:

$$\text{Cov}(\mathbf{R}_k, \mathbf{R}_k) = \sigma_{s_k}^2 \mathbf{\Psi}_k. \quad (6.31)$$

The parameter vectors $\boldsymbol{\eta}_k = [\eta_{k1}, \eta_{k2}, \dots, \eta_{kn_s}]^T$ and $\mathbf{d}_k = [d_{k1}, d_{k2}, \dots, d_{kn_s}]^T$ are estimated by maximizing the likelihood of the data \mathbf{r}_k in the hope of minimizing the generalization error:

$$\begin{aligned} L(\mathbf{R}_k^{(1)}, \dots, \mathbf{R}_k^{(n_s)} | \mu_{s_k}, \sigma_{s_k}^2) &= \frac{1}{(2\pi\sigma_{s_k}^2)^{n_s/2}} \exp\left(-\frac{\sum_{i=1}^{n_s} (\mathbf{R}_k^{(i)} - \mu_{s_k})^2}{2\sigma_{s_k}^2}\right) \\ &= \frac{1}{(2\pi\sigma_{s_k}^2)^{n_s/2} |\mathbf{\Psi}_k|^{1/2}} \exp\left(-\frac{(\mathbf{r}_k - \mathbf{1}\mu_{s_k})^T \mathbf{\Psi}_k^{-1} (\mathbf{r}_k - \mathbf{1}\mu_{s_k})}{2\sigma_{s_k}^2}\right). \end{aligned} \quad (6.32)$$

By maximizing the natural logarithm of Equation (6.32) via setting its derivatives to zero, one obtains the maximum likelihood estimates of the mean μ_{s_k} and variance $\sigma_{s_k}^2$:

$$\hat{\mu}_{s_k} = \frac{\mathbf{1}^T \mathbf{\Psi}_k^{-1} \mathbf{r}_k}{\mathbf{1}^T \mathbf{\Psi}_k^{-1} \mathbf{1}}, \quad (6.33)$$

$$\hat{\sigma}_{s_k}^2 = \frac{(\mathbf{r}_k - \mathbf{1}\hat{\mu}_{s_k})^T \mathbf{\Psi}_k^{-1} (\mathbf{r}_k - \mathbf{1}\hat{\mu}_{s_k})}{n_s}. \quad (6.34)$$

Since μ_{s_k} and $\sigma_{s_k}^2$ depend on the parameter vectors $\boldsymbol{\eta}_k$ and \mathbf{d}_k , the maximization of the likelihood is achieved by inserting Equations (6.33) and (6.34) to Equation (6.32) and maximizing it numerically. Once the values of the $\boldsymbol{\eta}_k$ and \mathbf{d}_k parameter vectors are determined, they can be used for prediction of the responses for new input parameters \mathbf{z} , based on already collected samples. The predictions from the surrogate models have to be consistent with the previously collected data as well as the correlation parameters. Therefore, they should maximize the likelihood of the already observed data together with the prediction itself. This is achieved by augmenting previously

collected data by the new input and the corresponding outputs for the approximated responses:

$$\tilde{\mathbf{r}}_k = [\mathbf{r}_k, \hat{r}_k]^T, \quad (6.35)$$

to obtain an augmented correlation matrix:

$$\tilde{\Psi}_k = \begin{bmatrix} \mathbf{\Psi}_k & \boldsymbol{\psi}_k \\ \boldsymbol{\psi}_k^T & 1 \end{bmatrix}, \quad (6.36)$$

where $\boldsymbol{\psi}_k$ is a correlation vector between the observed data and the predictions:

$$\boldsymbol{\psi}_k = \begin{bmatrix} \text{cor} [R_k(\mathbf{z}^{(1)}), R_k(\mathbf{z})] \\ \vdots \\ \text{cor} [R_k(\mathbf{z}^{(n_s)}), R_k(\mathbf{z})] \end{bmatrix}, \quad (6.37)$$

The maximization of the likelihood of the augmented data yields:

$$\hat{r}_k = \hat{\mu}_{s_k} + \boldsymbol{\psi}_k^T \mathbf{\Psi}_k^{-1} (\mathbf{r}_k - \mathbf{1} \hat{\mu}_{s_k}), \quad (6.38)$$

and

$$\hat{\sigma}_k^2 = \hat{\sigma}_{s_k}^2 \left[1 - \boldsymbol{\psi}_k^T \mathbf{\Psi}_k^{-1} \boldsymbol{\psi}_k + \frac{1 - \mathbf{1}^T \mathbf{\Psi}_k^{-1} \boldsymbol{\psi}_k}{\mathbf{1}^T \mathbf{\Psi}_k^{-1} \mathbf{1}} \right]. \quad (6.39)$$

Finally, the surrogate model of the k^{th} response is given by:

$$R_k(\mathbf{z}) = N(\hat{r}_k(\mathbf{z}), \hat{\sigma}_k^2), \quad (6.40)$$

where the right-hand side stands for the normal distribution with a mean $\hat{r}_k(\mathbf{z})$ and a standard deviation $\hat{\sigma}_k^2$.

Once the surrogate models of the responses are constructed according to the approach presented above, based on the samples created during the initial DoE, the models can be updated by maximizing the Expected Improvement (EI) acquisition function (Forrester et al. (2008)):

$$E[I(\mathbf{z})] = \begin{cases} (r_{1_{\min}} - \hat{r}_1(\mathbf{z})) \Phi_G \left(\frac{r_{1_{\min}} - \hat{r}_1(\mathbf{z})}{\hat{\sigma}_1(\mathbf{z})} \right) + \hat{\sigma}_1(\mathbf{z}) \phi_G \left(\frac{r_{1_{\min}} - \hat{r}_1(\mathbf{z})}{\hat{\sigma}_1(\mathbf{z})} \right) & \text{if } \hat{\sigma}_1(\mathbf{z}) > 0 \\ 0 & \text{if } \hat{\sigma}_1(\mathbf{z}) = 0 \end{cases}, \quad (6.41)$$

where Φ_G stands for the cumulative distribution function and ϕ_G for probability density function of a Gaussian distribution. Since in this work only single-objective optimization problems are considered, the response r_k for $k = 1$, denoted as r_1 , is assumed to be the objective function.

The maximization of EI is usually carried out with use of non-gradient global optimization methods such as EAs, e.g. Differential Evolution (DE) (Storn and Price (1997)), which is used in this work. Since the evaluation of EI is computationally cheap compared to the FE simulations, the choice of a particular method is not of the highest importance⁴.

In case of TO problems, during the optimization of EI, many designs would be infeasible from the structural point of view, as illustrated in Figure 6.8. This is mainly due to the fact that the maximization of EI leads to sampling of the points in the areas of high uncertainty of the surrogate model, resulting in a much more global search than in case of EA-LSM.

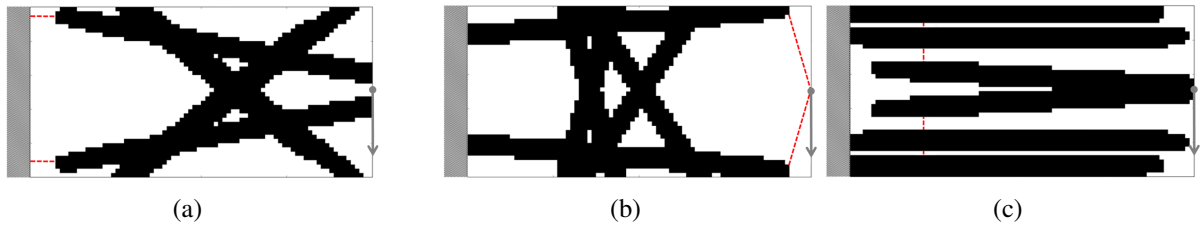


Figure 6.8 Illustration of the possible violations of structural feasibility for a cantilever beam problem (Raponi et al. (2019a)). Figure 6.8(a) shows a loss of the connection between the structure and the support, in Figure 6.8(b) the structure loses the connection to the load, while Figure 6.8(c) presents an example of internally disconnected structure. Red dashed lines in the plots correspond to the distances used to calculate the levels of violation of the structural feasibility constraints.

To eliminate such designs before performing costly FE evaluations, the EI criterion (Equation (6.41)) is modified to penalize infeasible structures:

$$\overline{EI}(\mathbf{z}) = \begin{cases} E[I(\mathbf{z})] & \text{if } \mathbf{z} \text{ is connected,} \\ -B(\mathbf{z}) & \text{if } \mathbf{z} \text{ is disconnected,} \end{cases} \quad (6.42)$$

where $B(\mathbf{z})$ is a function proportional to the violation of the connectivity constraint measured by the minimal distance between the MMCs necessary to build a structurally feasible design:

$$B(\mathbf{z}) = \kappa (B_1(\mathbf{z}) + B_2(\mathbf{z}) + B_3(\mathbf{z})), \quad (6.43)$$

with κ being a large constant playing the role of a penalty factor and $B_1(\mathbf{z})$, $B_2(\mathbf{z})$, and $B_3(\mathbf{z})$ are levels of violation in the first, second, and the third type of infeasibility (Figure 6.8), respectively.

⁴ For high-dimensional optimization problems, the performance of the algorithm used for the optimization of EI could become an important factor. For these cases, both the efficiency of the implementation as well as the properties of the optimization method should be examined carefully. In this work, however, we assume that, in general, the cost of a crash simulation is considerably higher than the numerical cost of the optimization of the acquisition function.

The levels of violation are calculated as the normalized distances required to create a structurally feasible design, being:

- Shortest distance from the material to the support (Figure 6.8(a)).
- Shortest distance from the material to the load (Figure 6.8(b)).
- Shortest distance between disjoint parts of the design (Figure 6.8(c)).

For more information regarding the computation of the levels of violations, please refer to the work by Raponi et al. (2019a).

The modification of EI in the form given by the Equation (6.42) is necessary to avoid plateaus in the optimization landscape, where the gradient of the objective function equals $\mathbf{0}$, which would be the case if the structurally infeasible designs implied constant values of EI. Such plateaus are usually difficult to deal with for evolutionary optimizers, since the selection operator is unable to distinguish between the infeasible designs, which have exactly the same fitness. In particular, if the entire population of individuals happens to be in the plateau, the optimization changes to a random walk (Pearson (1905)). On the other hand, evaluating disconnected designs would either cause problems with the FE simulations or lead to many high-fidelity simulations of inferior designs, which would be computationally inefficient.

In fact, the problem of disconnected structures occurs also in case of optimization using EA-LSM (Chapter 5), but is not so severe since EA-LSM performs mainly a search based on the mutation operator and starts the optimization from a population of individuals normally distributed around the reference design. As a result, it progresses by making relatively small modifications of the previous designs and all of the structurally infeasible solutions are rejected during the selection step. In principle, the penalization scheme could be helpful also in case of EA-LSM, but is not used in this work in order to guarantee a high level of generality of the method. In more complex cases, defining all the connections to the supports and loads prior to the optimization might be difficult and too restrictive, leading to inferior designs. Therefore, penalization of structurally infeasible designs can be considered as optional for EA-LSM, while it is a must for KG-LSM due to a much more global character of the EI-driven search.

In case of constrained optimization problems, instead of maximizing EI (Equation (6.41)), so-called Constrained Expected Improvement (CEI) is maximized:

$$CEI(\mathbf{z}) = E[I(\mathbf{z})] \prod_{i=1}^{n_{ineq}} P[F_i(\mathbf{z})], \quad (6.44)$$

where $P[F_i(\mathbf{x})]$ is the probability of feasibility of the i^{th} constraint, defined as:

$$P[F_i(\mathbf{z})] = \frac{1}{\hat{\sigma}_{s_{i+1}}^2(\mathbf{z})\sqrt{2\pi}} \int_0^\infty \exp\left(-\frac{(F_i - \hat{r}_{i+1}(\mathbf{z}))^2}{2\hat{\sigma}_{s_{i+1}}^2(\mathbf{z})}\right) dR_{i+1}, \quad (6.45)$$

and $F_i = R_{i+1}(\mathbf{z}) - r_{i+1_{min}}$, is the measure of feasibility of the i^{th} constraint, with $r_{i+1_{min}}$ being the limit value.

The product of probabilities in Equation (6.44) is justified by the assumption about the independence of the random variables.

The penalized version of CEI (Equation (6.44)) is given by the following formula:

$$\overline{CEI}(\mathbf{z}) = \begin{cases} CEI(\mathbf{z}) & \text{if } \mathbf{z} \text{ is connected,} \\ -B(\mathbf{z}) & \text{if } \mathbf{z} \text{ is disconnected.} \end{cases} \quad (6.46)$$

Similarly to the unconstrained optimization problem, Equation (6.46) is maximized numerically using DE. The obtained design vector $\mathbf{z}_{infill} = \arg \max_{\mathbf{z}} (\overline{CEI})$ satisfies all of the constraints according to the meta-models as well as guarantees structural feasibility of the design.

The design found by maximization of Equation (6.46), or alternatively, Equation (6.42) in case of unconstrained optimization problem, is evaluated based on high-fidelity static or crash FE simulation. The responses from the simulation are used to enhance the meta-models by adding new samples to the datasets and updating all of the models. This procedure is repeated until convergence, as outlined in the pseudocode of KG-LSM (Algorithm 3). In Algorithm 3, $iter$ is the iteration counter and $iter_{max}$ is the maximal number of iterations, set accordingly to the available budget of FE evaluations. To denote the best design found by the algorithm, symbol \mathbf{z}_{opt} is used, with the corresponding value of the objective function $f_{obj_{opt}}$. The rest of the symbols used in Algorithm 3 have been defined earlier in this section.

Please note that in the initial DoE, the structural feasibility criterion is not integrated into the sample generation process. Instead, after the vectors of design variables are generated via DoE, the vectors corresponding to the structurally infeasible designs are deleted and not evaluated using FE simulations. As a result, the initial surrogate models are trained using exclusively structurally feasible designs.

The approach described above can be extended to address problems with more than one objective function by using multi-objective EGO approaches (Forrester et al. (2008); Jeong and Obayashi (2005)). Furthermore, to improve the scalability of the algorithm, multi-modal optimization techniques (Fender et al. (2016)) can be used to find multiple optima of \overline{EI} or \overline{CEI} , allowing for evaluation of several infill points in parallel. Similarly, multi-objective optimization techniques can be used to find a Pareto front of solutions balancing exploration and exploitation via weighting the terms in Equation (6.41) (Grobler et al. (2018)), which can be then evaluated in parallel.

Another interesting extension is hybridization of KG-LSM with EA-LSM based on the observation that KG-LSM converges very fast at the beginning of the optimization, while EA-LSM is much

```

/* Design of Experiments                                     */
Generate  $n_s$  input samples using OLHS:  $\mathbf{Z} = [\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots, \mathbf{z}^{(n_s)}]^T$ ;
/* Evaluation of the responses                               */
for  $i = 1$  to  $n_s$  do
    Generate FE model based on parameters  $\mathbf{z}^{(i)}$ ;
    for  $k = 1$  to  $n_{ineq} + 1$  do
        Evaluate response  $r_k^{(i)}$  using FEM;
        Append  $r_k^{(i)}$  to the vector  $\mathbf{r}_k$  collecting the responses;
    end
end
/* Elimination of the infeasible designs                     */
for  $i = 1$  to  $n_s$  do
    if  $\mathbf{z}^{(i)}$  is infeasible then
         $(\mathbf{Z}, \mathbf{r}_k) := (\mathbf{Z}, \mathbf{r}_k) \setminus \{\mathbf{z}^{(i)}, \mathbf{r}_k^{(i)}\}$ ;
    end
end
/* Main optimization loop                                    */
while  $iter < iter_{max}$  do
    for  $k = 1$  to  $n_{ineq} + 1$  do
        Construct Kriging model for  $k^{\text{th}}$  response based on collected data  $(\mathbf{Z}, \mathbf{r}_k)$ ;
    end
    Find a feasible infill point using DE:  $\mathbf{z}_{infill} := \arg \max_{\mathbf{z}} (\overline{CEI})$ ;
    Update the set of input samples:  $\mathbf{Z} := \mathbf{Z} \cup \mathbf{z}_{infill}$ ;
    for  $k = 1$  to  $n_{ineq} + 1$  do
        Evaluate response  $r_k(\mathbf{z}_{infill})$  using FEM;
        Update the set of responses:  $\mathbf{r}_k := \mathbf{r}_k \cup r_k(\mathbf{z}_{infill})$ ;
    end
end
Find the objective value:  $f_{obj_{opt}} := \min(\mathbf{r}_1)$ ;
Find the best design:  $\mathbf{z}_{opt} = \mathbf{z} \in \mathbf{Z} : r_1(\mathbf{z}) = f_{obj_{opt}}$ ;

```

Algorithm 3: Pseudocode of the KG-LSM algorithm (Raponi et al. (2019a)).

more efficient in the final exploitation phase. This idea was used in the approach proposed by Raponi et al. (2019b), where KG-LSM switches to EA-LSM once a given convergence criterion is satisfied. This turned out to be a very efficient approach in reducing the number of costly FE evaluations and could be further improved by initializing the covariance matrix used by CMA-ES in EA-LSM with use of the data points collected during the optimization using KG-LSM (Mohammadi et al. (2015)).

Finally, the latest developments of surrogate-assisted optimization techniques for large datasets (van Stein et al. (2017)) and high-dimensional problems (Kyriacou et al. (2014); Wang et al. (2016); Bouhlel et al. (2016, 2018)) seem to be very promising from the perspective of applying KG-LSM to real-world TO tasks. Some of those techniques can be combined also with a

gradient information to enhance the surrogate models (Bouhleb and Martins (2019)), which is very interesting in the context of the methods for gradient approximation described in Section 6.1.

6.3. Adaptive evolutionary level set method (A-EA-LSM)

This section follows a different strategy to mitigate relatively high computational costs of EA-LSM compared to the state-of-the-art methods. Instead of improving the efficiency of the optimization algorithm itself, it reduces the dimensionality of the search space by relying on the concept of adaptive encoding (Olhofer et al. (2001)). Intuitively, this resembles the manual design process, where the complexity of a part is gradually increased by adding new geometric features. The main idea in the proposed approach is to take advantage of the fact that the convergence velocity of EA increases when the number of design variables is decreased (Bäck (1996)). As a result, the computation time can be strongly reduced by limiting the dimensionality of the search space. In case of ESs, used in this work, estimated convergence speed scales as $\mathcal{O}(n)$ to $\mathcal{O}(n^2)$, depending on the type of the objective function (Hansen and Ostermeier (2001)). Most of the adaptive approaches start with a low-dimensional representation of the design to quickly find an optimum in the low-dimensional space. When the optimization converges, the representation is extended to fine-tune the solution. This idea is illustrated in Figure 6.9. The key assumption here is that the optimum found in a low-dimensional space is a good approximation of the optimum when the representation is extended. The adaptation process is repeated until a satisfactory resolution of the representation is obtained. Still, one of the fundamental questions to be answered is how to extend the representation in a way that would offer the best convergence velocity, while not limiting considerably the ability of the method to find the global optimum.

In general, the idea of using adaptive representations in TO is not new and has proven to be a useful way of reducing the computational costs (Ortmann and Schumacher (2013); Ortmann (2015); Gilbert and Tyas (2003); Hagishita and Ohsaki (2009)) in optimization of truss and thin-walled structures. This work introduces an adaptive representation consistent with the MMC framework, which can be used together with ESs. To achieve that, the method uses the idea of speciation of a variety of topologies in niches (Stanley and Miikkulainen (2002)), while allowing for much more diverse topology modifications than in the previously mentioned approaches. However, the main contribution here is an introduction of a learning-based topology variation method (Bujny et al. (2018)), which is a fundamental difference compared to the heuristics used by the existing approaches (Ortmann and Schumacher (2013)). As a result, the method solves the problem of how to efficiently extend the representation using ML models, trained exclusively based on the data from numerical experiments. The main assumption is that there are certain rules, reflected for instance in the form of existing engineering intuition, which can be learned from the numerical simulations. The existence of such rules can be also justified by the conclusions of the research carried out by Schumacher and Ortmann (2013). From this perspective, the method presented in this work can

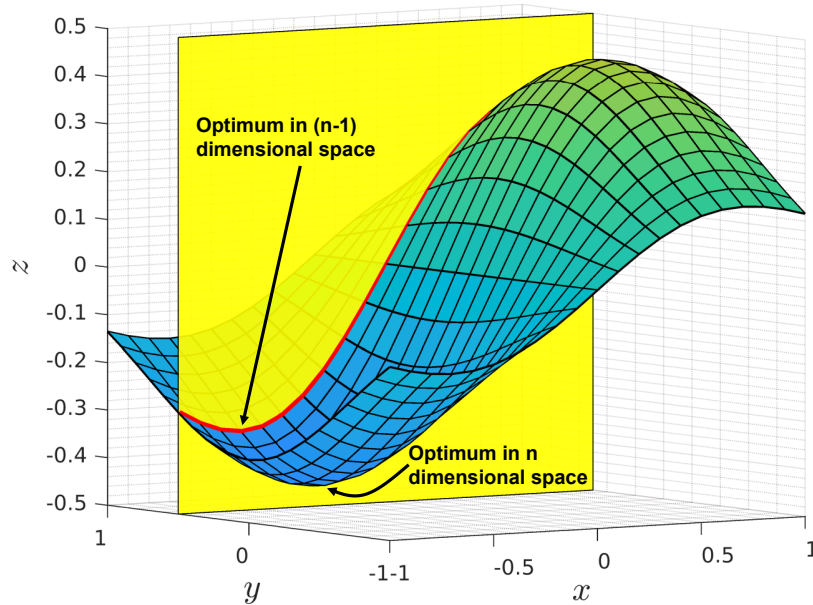


Figure 6.9 Simplified illustration of the changes in the landscape of the cost function due to the introduction of new variables in the parametrization after Olhofer et al. (2001). The simplification assumes that the introduction of new parameters does not change the landscape of the initial, $(n - 1)$ -dimensional optimization problem (the shape of the solid line remains the same after extension of the search space).

be seen as an alternative, data-driven approach for discovering engineering rules, potentially useful also in other optimization methods, or even as a standalone AI assistance tool in engineering software, capable of suggesting favorable design changes.

6.3.1. Graph-based parametrization

In the standard EA-LSM (Chapter 5), the parameters defining each of the MMCs $(x_{0i}, y_{0i}, \theta_i, l_i, t_i)$ ⁵, are used as design variables for the optimization algorithm. This type of representation, however, is not convenient for an adaptive encoding, where the number of MMCs is kept low at the beginning and increased gradually as the optimization progresses. The main problem appearing in such a case is schematically presented in Figure 6.10. Assume that the starting topology, in case of a cantilever beam design problem, is composed of two MMCs connected at the right end. This design can be represented in form of a graph having two edges and a single node (design A). As far as this low-dimensional design representation is considered, this topology corresponds to the global optimum. However, once an additional connection in the graph is introduced (design B), the modified design, consisting of four MMCs, would be rather a local than the global optimum (design C) in the extended parameter space. Now, in a standard MMC representation, the optimizer would need to realize a correlated change of 12 design parameters to go from the local to the global optimum. Considering mutation as the dominating operator in ESs (Bäck (1996)), this would be very unlikely, and, most probably, the optimization would converge to the design A by reducing the thickness of the new connection. An alternative approach would be to use the positions of the nodes and the thicknesses of the edges of the graph corresponding to the structure as the

⁵ Without loss of generality, the discussion in this section is limited to 2D optimization problems.

design variables. Using a graph-based representation, the transition from the design B to design C would require a correlated mutation of only two parameters, which define the position of the newly inserted node.

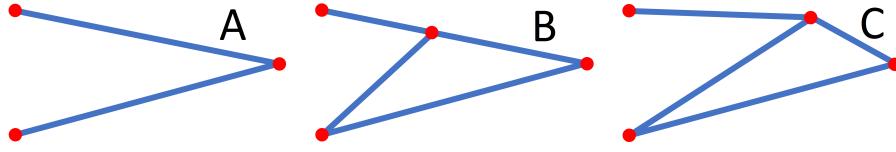


Figure 6.10 Starting topology (A), being a global optimum in a low-dimensional search space, topology modified by introducing a topology variation (B) and the global optimum in the higher-dimensional search space (C) (Bujny et al. (2018)).

The situation discussed above is a standard problem occurring when using adaptive representations and justifies using graph-based parametrization in such cases. Moreover, by assuming certain connectivity between the components, a graph representation yields less design variables for the same number of MMCs. This, however, is associated with a less general parametrization, where the topological changes can no longer be realized by means of mutation and recombination, but need a specialized adaptation operator. On the other hand, the reduction of design variables can lead to a considerable increase of the convergence velocity of the optimization algorithm. The arguments mentioned above support the idea of using graph-based parametrization, together with a suitable mapping to the MMC representation, in the A-EA-LSM, discussed in this section.

The mapping of the graph representation of the design to the standard MMC-based parametrization used in the EA-LSM takes the following form in a 2D case:

$$(x_{0i}, y_{0i}, \theta_i, l_i, t_i) = \left(\frac{x_{1i} + x_{2i}}{2}, \frac{y_{1i} + y_{2i}}{2}, \arctan \left(\frac{y_{2i} - y_{1i}}{x_{2i} - x_{1i}} \right), \sqrt{(x_{1i} - x_{2i})^2 + (y_{1i} - y_{2i})^2}, t_{1,2i} \right), \quad (6.47)$$

with (x_{1i}, y_{1i}) and (x_{2i}, y_{2i}) being the coordinates of the first and the second node of the graph edge with index i . The corresponding thickness of the edge is defined by the parameter $t_{1,2i}$. All of the parameters defining each of the graph's edges ($i = 1, \dots, M$), i.e. $(x_{1i}, y_{1i}, x_{2i}, y_{2i}, t_{1,2i})$, compose the vector of design variables, which are modified by the optimizer. Two examples of the graph representations of a design, together with a mapping to a material distribution via the MMC framework, are shown in Figure 6.11.

6.3.2. Optimization algorithm

This section introduces modifications of the standard ES (Bäck (1996)), which enable efficient optimization with the adaptive representation discussed above. The changes include:

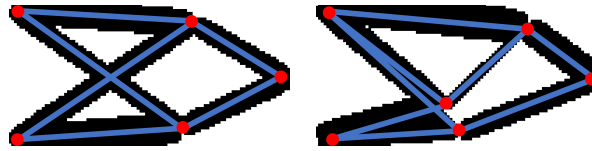


Figure 6.11 Exemplary material distributions with the corresponding graph representations (Bujny et al. (2018)). The designs correspond to the global (left) and a local (right) optimum in a compliance minimization problem of a cantilever beam.

- Modified recombination operator.
- Additional topology variation operator.
- Introduction of the speciation of individuals.
- Derandomized step size adaptation.

The modifications are described in detail in the sections below.

Recombination

The simplest variant of EA-LSM uses standard ESs, where the dimensionality of the optimization problem is assumed to remain constant. In case of an adaptive representation used in this section, this is not true and requires a modification of the recombination operator. In ESs, recombination plays a secondary role, due to the fact that the search is mostly based on random mutations. It turns out that ESs can work without the recombination on design variables, but the recombination on strategy parameters is necessary (Bäck (1996)). Therefore, for the sake of simplicity, the optimization approach presented in this section completely eliminates the recombination on the vectors of design variables.

After the modification of the recombination operator, the optimization can be carried out with individuals of different dimensionalities existing in the same population. The accelerating effect of the adaptive representation can be fully exploited when the optimization starts from a design of possibly lowest structural complexity and the representation is gradually extended when the optimization converges to local optima in the lower-dimensional search spaces. The indication of the convergence of the algorithm can be based for instance on the step size σ_{best} of the best individual. When σ_{best} drops below a predefined threshold σ_{th} , the topology of the best individual is extended.

Topology variation

In this work, two different approaches for introducing topological modifications are considered. The first method relies on a random choice of a pair (node, edge) out of all of the possible combinations of already existing nodes and edges in the graph, excluding the cases when the node

belongs to the edge. Once a (node, edge) pair is randomly selected, a new connection is created between the node and the edge. As illustrated in Figure 6.16, we would always consider only the topological variations introducing new graph edges from one of the already existing nodes (red) to a new node (green) added by splitting an existing graph edge. The second method is based on a pre-trained neural network classifier, which predicts potentially favorable topological variations for a given objective, on the basis of the structural state of the design. This approach is described in detail in Section 6.3.3.

After either random or model-based selection of the (node, edge) pair, it is very important to identify possibly the best position of the newly inserted graph edge, to go away from the local optimum found by the optimization in the lower-dimensional search space. Otherwise, even good topology variations would be removed in most cases. We propose to accomplish that by performing an intermediate, fast optimization in a two-dimensional search space by sampling different configurations via Optimal Latin Hypercube Sampling (OLHS) (Forrester et al. (2008)), as shown in Figure 6.12. The proposed algorithm, using exclusively the objective function of the main optimization problem, attempts to find the best elongation of the added edge as well as the position of the inserted node along the edge, which it splits. The position of the node along the edge is varied arbitrarily between the ends of the edge, while the elongation can change within a predefined range – in this work, corresponding to 20% of the height of the design domain. All of the other nodes and edges of the graph remain unchanged. Finally, after finding the best position and elongation, the thicknesses of all the graph edges are scaled so that the modified design respects the volume constraint. At the end, the modified individual is added to the population of the parent individuals.

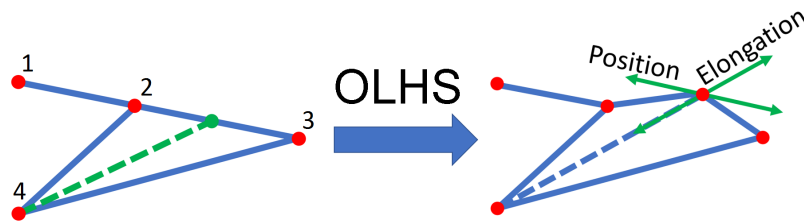


Figure 6.12 Addition of a new connection (depicted with a green dashed line) at the (node, edge) pair (4, (2,3)) and the parametrization of the new edge used to evaluate the improvement potential associated with the topology variation (Bujny et al. (2018)). Different configurations are explored via OLHS on the elongation of the new edge and the position of the new node along the edge, which it splits.

Speciation

Although the design after a topology variation is preliminary optimized, often, it cannot compete with the simpler designs present in the parent population, which are already very close to the local optimum in a lower-dimensional search space. Once a topology variation is introduced, all of the design variables defining the modified structure have to be adapted to take advantage of the extended representation. What is more, the optimization in a higher-dimensional search space is slower, as well. Therefore, a protection of the extended representation is commonly recommended

(Olhofer et al. (2001); Stanley and Miikkulainen (2002)) to allow the modified designs to improve to a level allowing it to compete with the simpler structures. One of the easiest approaches is to evolve the individual with an extended representation for a certain number of generations in a completely separate population, and then to add it to the population of individuals of different dimensionalities. However, this approach can be computationally costly and requires defining another set of parameters for the sub-optimization beforehand. Alternatively, one could protect the modified individuals by using niching techniques (Stanley and Miikkulainen (2002)), which allow the individuals with modified topologies to compete in a niche, while still being a part of the larger population. As a result, the designs in the niches have enough time to adapt and to be able to compete with the simpler topologies. This concept is used in this work, where the niches are created by modifying the cost function f_k of the k^{th} individual as follows:

$$f'_k = f_k s_k, \quad (6.48)$$

with s_k being the total number of individuals belonging to the same species as the k^{th} design, i.e. with the same number of graph edges (or MMCs).

Step size adaptation

Finally, it turns out that the standard self-adaptation used in ESs (Schwefel (1987)) is not well-suited for the optimization with the adaptive representations. The reason is that the main assumption of the self-adaptation, that the designs with low cost function values result from utilizing the most appropriate step sizes for a given landscape of the optimization problem (Schwefel (1987)), is frequently violated due to the introduction of the topology variation operator. More precisely, topology variations often result in large improvements in terms of the cost function, which is completely independent from the particular choices of the step sizes, and leads to a very unstable behavior of the step sizes adapted with use of the self-adaptation mechanism. In order to solve that problem, this work proposes to use a derandomization of the step size adaptation proposed by Hansen and Ostermeier (2001):

$$\sigma_k^{(g+1)} = \sigma_k^{(g)} \exp\left(\frac{\|\mathbf{u}_k\| - E[\|N(\mathbf{0}, \mathbf{I})\|]}{d_{ad}\sqrt{n_k}}\right), \quad (6.49)$$

$$\mathbf{z}_k^{(g+1)} = \mathbf{z}_k^{(g)} + \sigma_k^{(g)} \mathbf{u}_k, \quad (6.50)$$

with $\sigma_k^{(g)}$ and $\sigma_k^{(g+1)}$ being the step sizes corresponding to the k^{th} offspring individual for generations g and $g+1$, respectively. The vector \mathbf{u}_k is drawn from a multivariate normal distribution $N(\mathbf{0}, \mathbf{I})$, where n_k stands for the dimensionality of the parametrization of the k^{th} individual, defined by a vector of design variables \mathbf{z}_k . Finally, an adaptation damping parameter d_{ad} is introduced,

as well. The A-EA-LSM optimization algorithm is summarized in form of a flowchart in Figure 6.13.

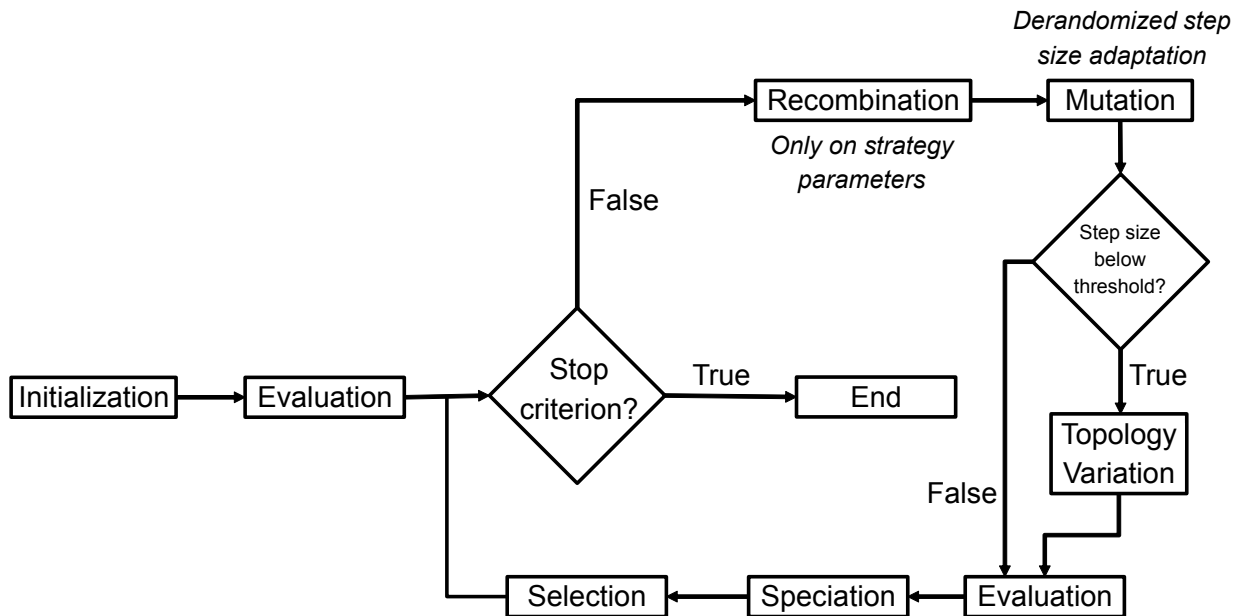


Figure 6.13 Flowchart of TO using A-EA-LSM.

6.3.3. Learning-based topology variation

As an alternative to a purely random way of selecting (node, edge) pairs for the introduction of topological variations, in this section, an ML-based approach is proposed. In fact, A-EA-LSM can give already satisfactory results with a random approach due to its ability to reject unfavorable topology modifications. However, since using heuristic rules for topology variations turned out to be a promising strategy for crash TO (Ortmann and Schumacher (2013)), one could assume that a discovery of such rules based on the collected data should be possible. The data-driven approach has clear advantages compared to the heuristic rules relying on the expert knowledge:

- The proposed method has a generic character. New rules for different optimization criteria can be easily learned using the fully automated process, whereas the standard approach would require an organizationally cumbersome process to acquire expert knowledge.
- Instead of using general expert rules for crashworthiness design, specific rules for particular objectives can be learned from the data.
- The proposed approach gives the possibility to learn novel rules, not used in engineering practice before.

In this section, an automated approach for learning rules for topological variations in static and crash cases is discussed. Exemplary data generation methods as well as the feature engineering process are explained. More precisely, the description below covers the following aspects of the data generation process:

- Generation of a diverse set of topologies.
- Sampling of different structural variations for each of the generated topologies.
- Feature generation based on the geometry and structural state of each topology.
- Transformation of the features for training of a classifier performing a pairwise comparison between different candidate topology variations.

The last part of this section proposes an approach for building neural networks for the prediction of favorable topological modifications, based on the geometric and structural state features of the design, e.g. stresses or displacements.

Sampling

The process of automated rule learning starts with a generation of a dataset of different designs. In particular, these designs could be already existing engineering structures, collected in the past design cycles, or result from a TO process. In those cases, a corresponding graph representation of a topology has to be found, which could be achieved e.g. using automated skeletonization techniques (Kresslein et al. (2018)). For simplicity, in this work, only limited number of designs parametrized using the graph representation is considered for sampling. For each of the generated structures, several different boundary conditions are used to obtain sufficient variability in the dataset of scenarios.

An example of a group of parametrized graph structures, used in this work for the generation of the design database for static and crash cases, is shown in Figure 6.14. The green vectors indicate the degrees of freedom determining the positions of the red nodes, which are modified to apply both shape as well as topological modifications. The positions of the nodes are varied according to a uniform random distribution, within ranges corresponding to the relevant dimensions⁶ of the design space. The topological modifications occur when the variations of the positions of the nodes result in a creation of new intersections in the topology (yellow nodes). The modifications of the supporting and loading conditions for the static cases are realized by a random selection among one of the 3 options shown in Figure 6.15. For crash cases, only the situation when the design domain is fixed at both ends and the structure is impacted in the middle by a cylindrical pole, as in Figure 6.4(a), is considered. As a result of this process, N_T different topologies with diverse boundary conditions are generated and simulated using FEM. Please note that due to the possible topological modifications of the base topologies resulting from that process (new intersections), the underlying graph representation has to be identified again to include all of the newly created nodes and edges. In the next step, for each of the generated cases, all (node, edge) pairs, excluding nodes belonging

⁶ For example, in the samplings performed in this thesis, the vertical positions of the red nodes on the boundary of the design space were allowed to take any values in a range $(0, h_{dd})$, corresponding to the bottom and top edge of the design domain. Similarly, the positions of the red nodes along the graph edges (e.g. the graph in the first row and the second column in Figure 6.14) were allowed to vary arbitrarily between both ends of the intersected edge.

to a given edge, are listed as potential candidates for introducing topology variations. Those cases will be later on referred to as the feasible topology variations, since they include exclusively the cases when the topology, and not only shape or size, is modified. Subsequently, for each of the listed (node, edge) pairs, a new connection is introduced by connecting the existing node with a new node created at the edge. As an example, for the design B (Figure 6.10), all feasible topology variations are illustrated in Figure 6.16. In order to estimate the improvement potential (in terms of the considered objective function) offered by each of the topological variations, the exact position of the new node along the edge is determined based on the procedure described in Section 6.3.2 (Figure 6.12). Finally, the objective's improvement values for each topology variation are listed along with the structural features describing the corresponding (edge, node) pair. These features are extracted directly from the FE simulation results and are described in the next section.

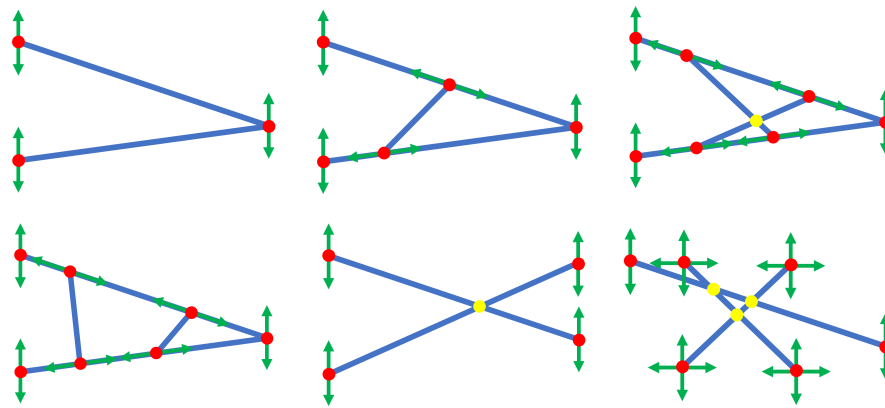


Figure 6.14 Parametrized topologies for generation of samples for training topology variation prediction model (Bujny et al. (2018)). Green vectors depict the degrees of freedom varied to generate the dataset of topologies. Yellow points correspond to the new nodes created at the intersections of the edges as a result of the random variations of the parameters.



Figure 6.15 Three types of boundary conditions used in the sampling process (Bujny et al. (2018)). The nodes in the FE mesh can be fixed on the left, right or on both sides. In all of the three cases, a single point load is applied in a random direction and at a random position within the design space.

Feature extraction

In this work, we assume that, for a given design, the choice of a particular (node, edge) pair for introducing topology variation can be made exclusively based on the structural features describing the (node, edge) pair. A similar assumption was used for defining the heuristic rules for topology

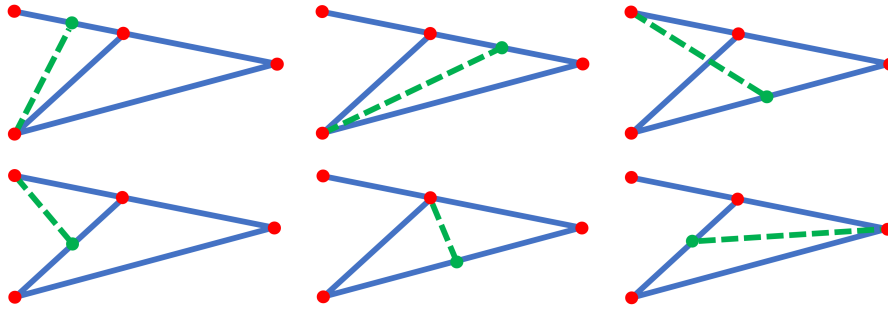


Figure 6.16 All feasible topology variations for a reference topology (solid blue lines) (Bujny et al. (2018)). Topology variations are realized always by connecting an existing graph node (red) with a new node (green) by adding a new graph edge (dashed green lines). In the sampling process, always all feasible topology variations are taken into account.

modifications (Ortmann and Schumacher (2013)), where, for instance, always the fastest deforming walls are supported. The structural features will be used later on to predict the ranking of the topology variations according to their improvement potential.

The structural features characterizing a given (node, edge) pair can be categorized as:

- Geometric features – encoding the information about the relative spatial position of the node w.r.t. the edge as well as their geometry.
- State features – specifying the structural state of the node and the edge using such quantities as displacements, equivalent stresses or strain energy density.

As an example, the geometric and the state features for linear elastic static cases, used in our previous publications (Bujny et al. (2018)) and in this work, are defined in Table 6.1 and Table 6.2. Altogether, 13 different features encoding the information about the structural state of the topology are considered. The features can consist of entire distributions of the quantities along the graph edges, as well. For crash cases, we take into account only the maximal values of the state quantities over time. The state features used for the crash cases are specified in Appendix A.1. The same geometric features as in the linear elastic case (Table 6.1) are used for crash.

The overall process for generating the vectors of structural features describing each (node, edge) pair, and the potential objective improvements associated with a topology modification by creating the new connection between the considered node and edge, is depicted in Figure 6.17.

For convenience of building a model for predicting the ranking of topological variations, one has to transform the original structural features describing (node, edge) pairs as well as the improvements of objective resulting from topological modifications (targets). This work proposes to use a pairwise transform of the feature and target vectors for each of N_{NE_i} (node, edge) pairs, using the

Table 6.1 Geometric features used for prediction of favorable topological variations in linear elastic static cases (Bujny et al. (2018)) as well as in nonlinear crash cases. Geometric features are normalized with the characteristic dimension of the design space, e.g. length of the diagonal.

ID	Feature
1	Minimal distance between the node and the straight line the edge lies on.
2	Distance between the node and the middle of the edge.
3	Shortest path along the graph from the node to the middle of the edge.
4	Length of the edge.

Table 6.2 Structural state features used for prediction of favorable topological variations in linear elastic static cases (Bujny et al. (2018)). Structural features are normalized using the maximal value of each field (e.g. von Mises stress, strain energy, displacement).

ID	Feature
5	Maximal resultant displacement in the neighborhood of radius 1 around the graph node.
6	Maximal strain energy in the neighborhood of radius 1 around the graph node.
7	Maximal von Mises stress in the neighborhood of radius 1 around the graph node.
8	Maximal resultant displacement along the graph edge.
9	Maximal strain energy along the graph edge.
10	Maximal von Mises stress along the graph edge.
11	Maximal resultant displacement in the middle of the edge.
12	Maximal strain energy in the middle of the edge.
13	Maximal von Mises stress in the middle of the edge.

following formula:

$$(\mathbf{X}'_k, \mathbf{Y}'_k) = (\mathbf{X}_i - \mathbf{X}_j, \text{sgn}(\mathbf{Y}_i - \mathbf{Y}_j)), \quad (6.51)$$

with \mathbf{X}'_k and \mathbf{Y}'_k being the k^{th} ($k = 1, \dots, (N_{NEt})^2$) feature and target vectors after pairwise transformation, respectively. The vectors of features describing the i^{th} ($i = 1, \dots, N_{NEt}$) and j^{th} ($j = 1, \dots, N_{NEt}$) (node, edge) pair, for a given reference topology, are denoted by \mathbf{X}_i and \mathbf{X}_j . Finally,

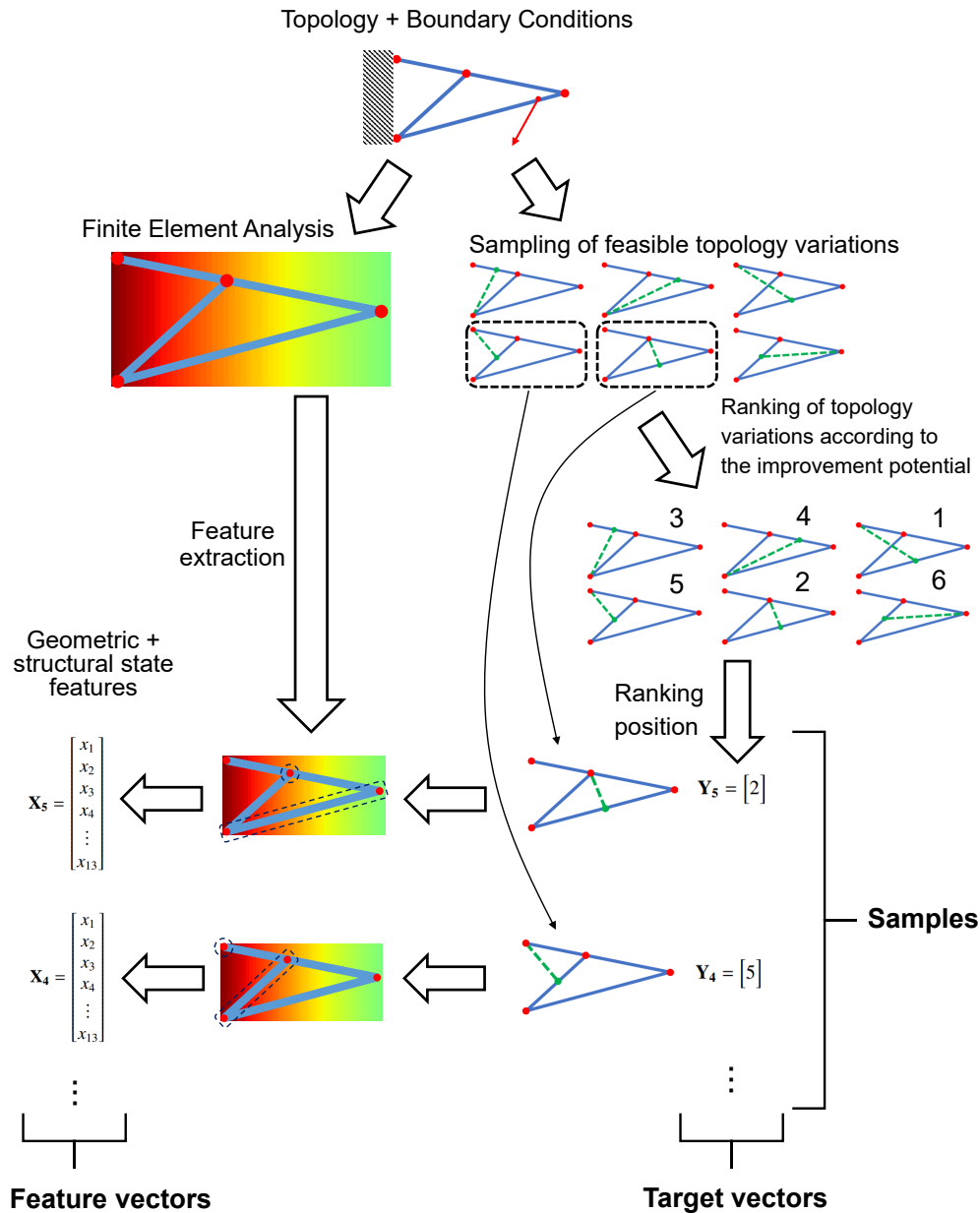
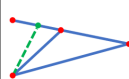
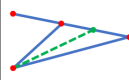
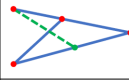
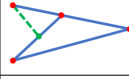
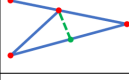
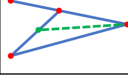


Figure 6.17 Illustration of the feature generation process based on the structural state features.

Y_i and Y_j are the ranking positions of those pairs, where the higher values correspond to better performance. An example of the pairwise transform for the sampling shown in Figure 6.16 is presented in Figure 6.18. Each of the 6 sampled topologies is compared with the others, leading to generation of 36 transformed feature and target vectors. Pairwise transformation of the samples from N_T different cases results in a total number of $N_S = \sum_{t=1}^{N_T} (N_{NEt})^2$ samples, used for training of an ML model.

Original feature and target vectors

	(X_1, Y_1)
	(X_2, Y_2)
	(X_3, Y_3)
	(X_4, Y_4)
	(X_5, Y_5)
	(X_6, Y_6)



Transformed feature and target vectors

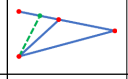
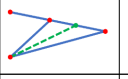
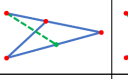
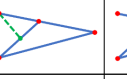
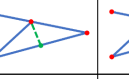
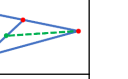

						
(X'_1, Y'_1)	(X'_2, Y'_2)	(X'_3, Y'_3)	(X'_4, Y'_4)	(X'_5, Y'_5)	(X'_6, Y'_6)	(X'_6, Y'_6)
(X'_7, Y'_7)	(X'_8, Y'_8)	(X'_9, Y'_9)	(X'_{10}, Y'_{10})	(X'_{11}, Y'_{11})	(X'_{12}, Y'_{12})	(X'_{12}, Y'_{12})
(X'_{13}, Y'_{13})	(X'_{14}, Y'_{14})	(X'_{15}, Y'_{15})	(X'_{16}, Y'_{16})	(X'_{17}, Y'_{17})	(X'_{18}, Y'_{18})	(X'_{18}, Y'_{18})
(X'_{19}, Y'_{19})	(X'_{20}, Y'_{20})	(X'_{21}, Y'_{21})	(X'_{22}, Y'_{22})	(X'_{23}, Y'_{23})	(X'_{24}, Y'_{24})	(X'_{24}, Y'_{24})
(X'_{25}, Y'_{25})	(X'_{26}, Y'_{26})	(X'_{27}, Y'_{27})	(X'_{28}, Y'_{28})	(X'_{29}, Y'_{29})	(X'_{30}, Y'_{30})	(X'_{30}, Y'_{30})
(X'_{31}, Y'_{31})	(X'_{32}, Y'_{32})	(X'_{33}, Y'_{33})	(X'_{34}, Y'_{34})	(X'_{35}, Y'_{35})	(X'_{36}, Y'_{36})	(X'_{36}, Y'_{36})

Figure 6.18 Transformation of the feature and target vectors according to Equation (6.51).

Training

The transformed samples are used for training a classifier returning one of three values:

- +1 if the improvement potential of topology variation i is higher than for the variation j .
- 0 if the improvement potentials of topology variations i and j are equal.
- -1 if the improvement potential of topology variation i is lower than for the variation j .

By comparing each candidate topology variation with all the other feasible topology variations based on the predictions of the classifier, the ranking of the most promising topology modifications can be constructed.

In order to solve the classification problem, different types of methods, e.g. Logistic Regression, Support Vector Machine (SVM), k-Nearest Neighbors (k-NN), can be considered. In this work, however, a Multilayer Perceptron (MLP) Artificial Neural Network (ANN) is used, due to its superior test accuracy in the prior comparative studies. As outlined above, the classifier is used to predict which of the two candidate topology variations offers higher improvement potential. The process of feature transformation and classifier training is summarized in Figure 6.19. Finally, the best topology variation according to the ranking constructed based on the pairwise comparisons is used in A-EA-LSM.

The approach described in this section has a generic character and can be used to learn topology variation rules for different objective functions. To improve the prediction accuracy for different types of problems, further feature engineering might be required. As soon as appropriate features are identified, the approach could be possibly used in different structural optimization problems.

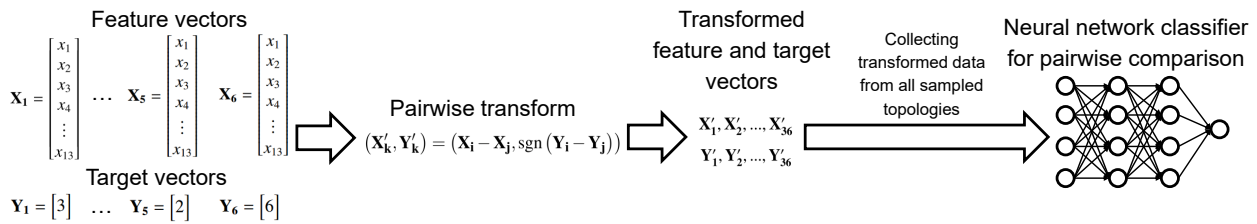


Figure 6.19 Illustration of the pairwise transformation of the original feature and target vectors and the training of a classifier used for prediction of favorable topological variations.

Similarly, one could also use a comparable approach with different parametrizations, e.g. as an alternative to the hand-engineered heuristic rules in the method proposed by Ortmann and Schumacher (2013).

6.4. Summary

This chapter proposes three methods for improving the efficiency of the EA-LSM, discussed in Chapter 5. All of them take advantage of ML techniques to address different aspects of TO, i.e. modeling of sensitivities, approximating objectives and constraints, or learning rules for introducing topology modifications. As such, the improvements can be treated as separate modules, which could be potentially combined together.

In the first approach, H-EA-LSM, an approximate gradient information is used to accelerate the evolutionary search. Depending on the optimization problem, the approximate gradient can be obtained from physical or mathematical surrogate models. The usefulness of the method depends on the accuracy of the gradient information for a given optimization problem. However, due to the ability of the algorithm to reject the individuals affected by inaccurate gradient predictions, no divergent behavior was observed in such cases.

The second method, KG-LSM, uses an adapted EGO approach to address TO problems. This is a novel approach for TO, taking advantage of the low-dimensional level-set representation used in this work. Due to the well-known limitations on the dimensionality of the surrogate-assisted optimization problems (Lim et al. (2010); Jin (2011)), especially for Kriging models used within the EGO approach (Wang et al. (2016)), KG-LSM is mainly applicable to optimization relying on a 2D representation. This does not exclude, of course, optimization of extrusions (Patel et al. (2009); Nutwell et al. (2017)), arbitrarily loaded plates (Goetz et al. (2012); Raeisi et al. (2019)), or topometry optimization of thin-walled structures (Liu et al. (2015); Han et al. (2015)), where a mapping between 2D MMCs and a 3D structure is explicitly specified. Thanks to very good convergence speed of KG-LSM in terms of FE evaluations, the method is applicable to very expensive TO problems, provided that the coarse, low-dimensional representation of the design is sufficient

for the problem. On the other hand, the applicability of the method to higher-dimensional problems looks very promising in the light of the recent advances of the research on high-dimensional surrogate-assisted optimization (Kyriacou et al. (2014); Wang et al. (2016); Bouhlel et al. (2016, 2018)).

Finally, the last approach, A-EA-LSM, utilizes the concept of an adaptive representation (Olhofer et al. (2001)) to reduce the computational costs. The representation of the design is grown during the optimization process, reducing the effective number of design variables, which directly influences the convergence velocity of the evolutionary search. Instead of using predefined rules for extending the representation (Ortmann and Schumacher (2013)), the rules are learned from the sampled data, which is a novel approach, interesting also from the perspective of assisting design process in the engineering software.

Unlike KG-LSM, both H-EA-LSM and A-EA-LSM have the potential to be applied also to optimization of 3D TO problems, but this work focuses on evaluation of performance of those methods in 2D. In case of A-EA-LSM, the large number of possible combinations for topology variations in 3D problems could make the approach less effective. The main reason for that is the accumulation of errors from pairwise comparisons using the predictions of the trained classifier to construct the ranking of topology variations.

In the following chapters, the proposed enhancements of EA-LSM will be evaluated on suitable static and crash TO problems.

Chapter 7

Validation by linear elastic cases

Validation is a crucial step in development of any new TO algorithm. One should be able to show that the new method is capable of providing solutions at least qualitatively similar to the broadly accepted reference designs for selected benchmark problems. Especially in the crashworthiness community, methods are frequently criticized for the lack of validation on representative cases. The variety of test cases used in crash TO makes the comparison between methods very difficult. In fact, even with consistent test cases, due to the heuristic assumptions made by the state-of-the-art methods, it is difficult to consider the resulting topologies as reliable reference solutions.

In the ideal case, one would like to validate first the proposed method on benchmark cases that can be solved with use of well-established gradient-based methods. Optimization of linear elastic structures seems to be a perfect candidate for addressing this task. In particular, for such problems as compliance minimization with a volume constraint, analytical sensitivities are available and well-known reference solutions exist.

Compared to crash problems, linear elastic cases are associated with lower computational costs, as well. This makes the structural-mechanical considerations and statistical evaluation of the investigated methods significantly easier. What is more, existence of the simplified engineering models for linear elastic problems allows for a much better understanding of the resulting topologies, giving an additional insight into the optimized designs as well as the optimization method itself. As a result, carrying out TO on linear elastic cases seems to be a very good solution for demonstrating the correctness of the methods proposed in this work. Although the efficiency of the proposed evolutionary-based algorithms is considerably lower than the established methods, the main focus

of this chapter is to evaluate the properties of the proposed approaches and check if they are able to reproduce the solutions obtained with gradient-based optimizers. Therefore, the analysis presented in this chapter is just an intermediate step on the way towards applying those methods in optimization of demanding crash problems.

The chapter is divided into two sections. Section 7.1 evaluates EA-LSM and its extensions based on various 2D linear elastic problems. To investigate the properties of EA-LSM, a comprehensive parameter sensitivity study is carried out. Finally, Section 7.2 addresses the question of feasibility of using the 3D version of EA-LSM based on a high-dimensional cantilever beam optimization problem.

7.1. Two-dimensional topology optimization for minimum compliance

The most common benchmark case in structural TO, used usually to validate new methods, is an optimization of a linear elastic cantilever beam. Typically, the optimization problem for cantilever beam involves minimization of the compliance, C , subject to a constraint on the total volume (or mass) of the structure. In such a case, the analytical sensitivity information is available and gradient-based methods can be used efficiently. Moreover, as a test scenario accepted in the TO community, reference solutions for this problem with different methods exist. Taking the arguments mentioned above into account, we choose it as the first test case used to validate the methods proposed in this work. Formally, the optimization problem addressed in this chapter can be described as follows:

$$\begin{aligned} \min_{\mathbf{z}} (C(\mathbf{z}) = \mathbf{f}\mathbf{u}(\mathbf{z})), \mathbf{z} \in \mathbb{R}^n; \\ \text{s.t. } \mathbf{K}(\mathbf{z})\mathbf{u} = \mathbf{f}; \\ V(\mathbf{z}) \leq V_{req}, \end{aligned} \quad (7.1)$$

where \mathbf{f} is a vector of nodal forces and \mathbf{K} is the stiffness matrix in a discretized FE model. V_{req} is the required volume ($V(\mathbf{z})$) of the structure. The vector of nodal displacements is denoted by \mathbf{u} .

In this chapter, cantilever beam cases as depicted in Figure 7.1 are used. In order to investigate the properties of EA-LSM, not only the standard 1:2 aspect ratio case with 50% volume fraction is used, but also the cases of 1:1 and 1:3 aspect ratio. Each of those cases is considered with 30%, 50%, and 70% volume fraction constraint.

In addition to the cantilever beam case, we introduce a transverse bending scenario, being a linear-elastic counterpart of the two-dimensional test cases used later on in TO for crashworthiness criteria in Chapter 8. The test case is depicted in Figure 7.2. As in case of the cantilever beam, the objective function minimized in this case is the compliance of the structure. However, only the

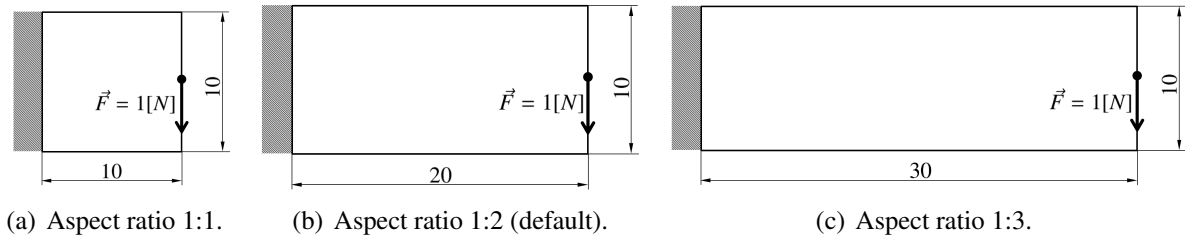


Figure 7.1 Design domain (different aspect ratios) and boundary conditions for the linear elastic cantilever beam test cases used for validation of the EA-LSM and its extensions. Dimensioning in mm.

50% volume fraction constraint is considered here. The exact configuration of all of the test cases considered in this chapter is given in Table 7.1.

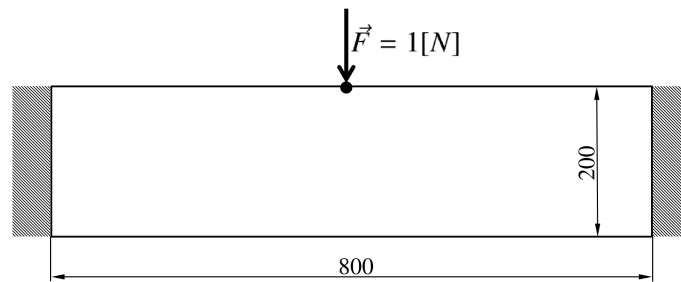


Figure 7.2 Design domain and boundary conditions for the linear elastic transverse bending test case used for validation of the EA-LSM and its extensions. Dimensioning in mm.

7.1.1. Evolutionary level set method (EA-LSM)

In this section, validation of the EA-LSM, as described in Chapter 5, is carried out. First of all, the ability of the method to reproduce the results of the state-of-the-art gradient-based methods is studied. Secondly, the study of the sensitivity of the algorithm with respect to the initial design is presented. Thirdly, the influence of the mesh resolution and the number of basis functions on the final solutions is investigated. Finally, we focus on the analysis of computational costs of the method, which is the starting point for Sections 7.1.2, 7.1.3, and 7.1.4, addressing the issue of mitigation of computational costs with methods proposed in Chapter 6.

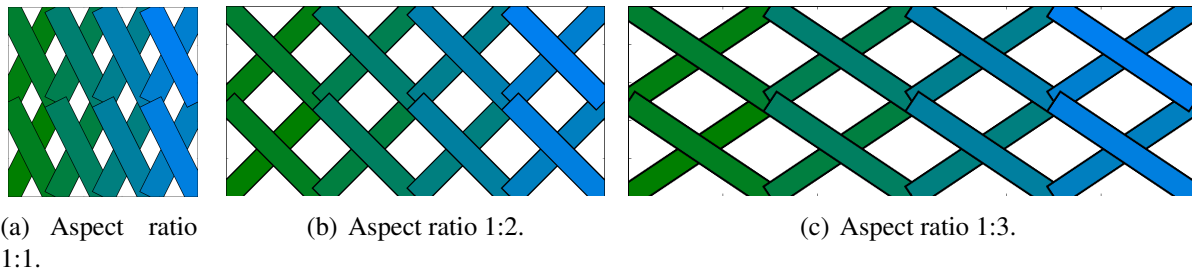
Validation of the method

The validation of the method in this section is based on the cantilever beam and the transverse bending test case presented above. In order to allow for a statistical evaluation of the results, for each of those nine cases, 30 optimization runs are performed. For each of the optimization runs, the initial population is created based on random variations (according to a normal distribution) around the diagonal reference design, by following the procedure described in Section 5.3.2.

In all of the cases considered in this section, the EA-LSM algorithm with 16 basis functions is used. As we show later, increasing the number of basis functions does not lead to a considerable

Table 7.1 Configuration of the static test cases used for validation.

Property	Symbol	Value	Unit
Young's modulus	E	$2.1 \cdot 10^5$	MPa
Poisson's ratio	ν	0.3	-
Load	F	1	N
Required volume fraction	V_f	30%, 50%, 70%	-
Mesh resolution cantilever, 1:1 aspect ratio	-	50 x 50	-
Mesh resolution cantilever, 1:2 aspect ratio	-	100 x 50	-
Mesh resolution cantilever, 1:3 aspect ratio	-	150 x 50	-
Mesh resolution transverse bending	-	160 x 40	-
Solver	-	CalculiX 2.9 (Dhondt (2004))	-
Element type	-	Four-node shell element (S4R)	-


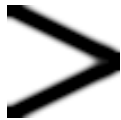



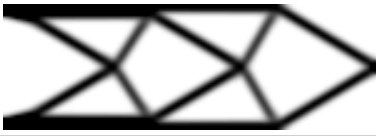









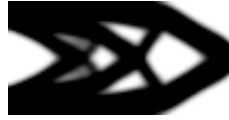
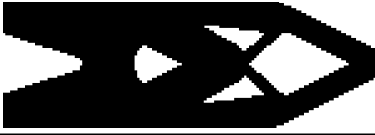
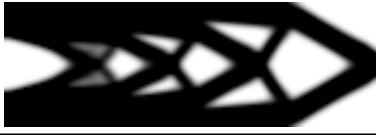
**Figure 7.3** Reference designs used for generation of the initial population of individuals for the cantilever beam test cases with different aspect ratios.

improvement in the objective function for the cases considered in this chapter, and is sufficient to represent topologically complex designs. Thanks to the use of symmetry conditions, this resulted in a 40-dimensional optimization problem, solved with CMA-ES(7,15). The exact setup of the optimization algorithm as well as its justification is given in Appendix B.2. The reference structures used for generation of the initial population of individuals for the cantilever beam cases are shown in Figure 7.3. For the transverse bending case, an initial layout as shown in Figure 5.9(a) is used. In all of the cases, the initial step size for CMA-ES and the standard deviation used to distribute the initial population around the reference design is set to $\sigma_{init} = 0.025$. For comparison, a state-of-the-art gradient-based SIMP method¹, as described by Andreassen et al. (2011), was used. The

¹ More precisely, Solid Isotropic Material with Penalization (SIMP) (Bendsøe and Sigmund (2004)) itself is a material

setup of the used SIMP method is defined in Appendix B.1.

Table 7.2 Comparison of the best topologies obtained with EA-LSM and SIMP (Andreassen et al. (2011)) for the cantilever beam design problem.

Volume fraction	Aspect ratio	Optimized topology	
		EA-LSM	SIMP
30%	1:1		
30%	1:2		
30%	1:3		
50%	1:1		
50%	1:2		
50%	1:3		
70%	1:1		
70%	1:2		
70%	1:3		

interpolation scheme, which penalizes the intermediate densities using a power function. As such, SIMP is used in different density-based methods, including also non-gradient approaches such as HCA (Patel (2007)). However, it is a common practice to refer to the gradient-based approaches using SIMP interpolation scheme as to the "SIMP methods" (Rozvany (2008); Dijk et al. (2013)). Those approaches rely on rigorously derived analytical sensitivities, and frequently use Optimality Criteria (OC) as the underlying gradient-based optimizers (Sigmund (2014)). For convenience, we follow the same convention in this work, and refer to the OC-based approach using SIMP interpolation simply as to "SIMP".

Table 7.3 Material distribution, layout of basis functions, and the global level set function corresponding to the best designs for 50% volume fraction, for aspect ratios of 1:1 (top), 1:2 (middle), and 1:3 (bottom). MMCs with zero thickness were deleted from the plots.


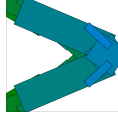



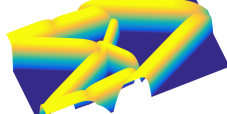

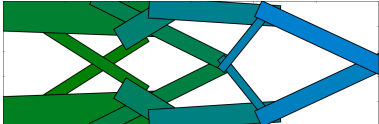
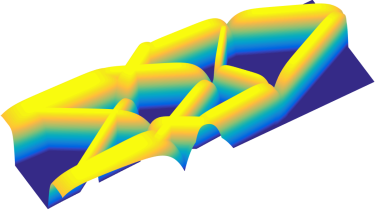
Material distribution	MMC layout	LSF
		
		
		

Table 7.2 shows the best designs obtained in 30 optimization runs for each of the nine cantilever beam cases. Additionally, for the case of 1:2 aspect ratio with 50% volume fraction, the evolution of the best design during the optimization is shown in Appendix C.1. The results show that, in general, EA-LSM produces topologies consistent with the ones obtained with SIMP. With only 16 MMCs, the method is able to develop structures of high complexity, as in case of the 1:3 aspect ratio and 50% volume fraction. However, for some cases, clear differences between the designs obtained with EA-LSM and SIMP are observable. The reasons for those differences are threefold:

- The parametrizations used in EA-LSM and SIMP are completely different. On the one hand, SIMP operates on thousands of design variables specifying densities of finite elements in a continuous fashion. On the other hand, EA-LSM uses a small number of MMCs, limiting the dimensionality of the search space to the order of tens, and restricts the densities to take two discrete values – 0 or 1. It is clear that both optimization problems are different due to the discrepancies in the representations and also that the optima for those two formulations might be different. What is more, EA-LSM might be not able to represent the designs obtained with SIMP, especially for very complex designs, since it is restricted by the number of design variables and the number of shape primitives. This problem is visible, for instance, for the structures with the 1:3 aspect ratio, where more MMCs would be necessary to represent the structures with higher accuracy. In contrast, for 70% volume fraction, SIMP has severe problems with the flexibility of its representation, resulting in a high number of elements with intermediate densities.
- The results obtained with SIMP are highly dependent on the setup of the method, in particular

filtering radius and mesh resolution. For instance, although the designs obtained with EA-LSM and SIMP for 1:3 aspect ratio and 30% volume fraction are qualitatively different; for a different set of hyperparameters, SIMP yields the same type of topology as EA-LSM.

- The optimization problem is nonconvex and local optima are likely to occur (Bendsøe and Sigmund (2004)). Especially the gradient-based methods might be very sensitive to that problem. As a result, there is no guarantee that the designs obtained with SIMP or EA-LSM correspond to the global optimum.

More insight into the topologies from Table 7.2 can be obtained by analyzing the layouts of MMCs and LSF plots. For the case of 50% volume fraction and aspect ratios of 1:1, 1:2, and 1:3, they are depicted in Table 7.3. The plots show how both, less and more complex structures can be represented with 16 basis functions. In case of less complex structures (1:1 aspect ratio), EA-LSM uses the limited number of basis functions to define precisely the shape of the design. In contrast, for more complex structures (1:3 aspect ratio), 16 basis functions are used to give a coarse representation of the design, grasping only the most important geometric features. This shows that the complexity of the structures can be limited explicitly by defining different numbers of MMCs. Unlike in the density-based approaches, where the computational costs depend mostly on the resolution of the FE mesh, since the number of design variables is usually equal to the number of finite elements, for EA-LSM, the number of MMCs is the parameter most strongly influencing the cost of the method. This seems to be a convenient property from the practical point of view, where less complex structures could be obtained with relatively low computational cost, while being still simulated with a high-fidelity FE model. The aspects of complexity control, structural attainability, and dependency of the solutions on the number of MMCs are discussed later in this chapter, as well.

Most of the further investigations in this section are carried out based on the transverse bending case, corresponding to the nonlinear dynamic crash test case considered in Chapter 8. Therefore, it is the second test case analyzed here. Again, the best result for the 30 optimization runs is compared with the design obtained in the SIMP approach and depicted in Figure 7.4. As in case of the cantilever beam, the designs are qualitatively similar, showing the ability of EA-LSM to optimize topologies for different sets of boundary conditions.



Figure 7.4 Comparison of the best topology obtained with EA-LSM and SIMP (Andreassen et al. (2011)) for the transverse bending problem.

Uniqueness of solutions and repeatability

In the previous section, the ability of EA-LSM to reproduce the designs obtained with state-of-the-art gradient-based methods was studied. The methodology we employed assumed running 30 optimizations for each of the cases and choosing the best result. However, in practice, the resources are limited and such a broad analysis of every optimized component would not be possible. Therefore, the repeatability of the solutions with EA-LSM is studied here based on the optimizations from the previous section.

Unlike gradient-based methods, EAs are stochastic optimizers. Since problems addressed in this work are generally nonconvex (Bendsøe and Sigmund (2004)), gradient-based approaches, being deterministic methods, would always lead to the same (local or global) optima for the same starting points, while evolutionary methods can potentially result in different solutions². Therefore, from the practical point of view, the average performance of the method is much more important.

For the sake of simplicity, let us focus exclusively on the cantilever beam cases with 50% volume fraction. Figure 7.5 shows the box plots of the final compliance values for the aspect ratios of 1:1, 1:2, and 1:3. The compliance values are scaled to show relative performance improvement of the designs compared to the median design. The corresponding best, median, and worst designs are depicted in Table 7.4. Additionally, for each aspect ratio, Table 7.5 presents the three most often appearing topology types and the frequency with which they were obtained in 30 optimization runs.

Table 7.4 The best, median (15th), and the worst out of 30 cantilever beam designs obtained for the aspect ratios of 1:1, 1:2, and 1:3 with 50% volume constraint.

		Aspect ratio		
Design	1:1	1:2	1:3	
Best				
Median				
Worst				

² In general, evolutionary methods are considered to be much more explorative and therefore much better in locating the global optimum (Rao (2009)).

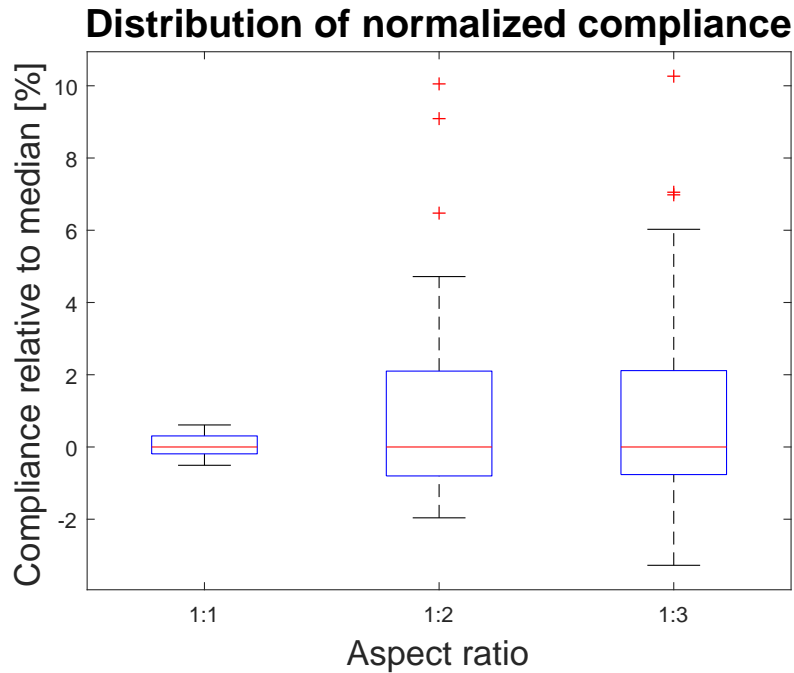











Figure 7.5 Distribution of normalized compliance values for the cantilever beam test case with 50% volume fraction and aspect ratios of 1:1, 1:2, and 1:3. Each of the box plots is based on 30 samples obtained in separate optimization runs. The compliance values were normalized according to the formula: $\frac{C_i - m}{m} \cdot 100\%$, where C_i is the i^{th} compliance sample and m is the median for a given aspect ratio. Therefore, negative values show the percentage improvement of performance, while positive correspond to percentage drop of performance. The line in center of each box indicates the median, while the top and bottom edges of the boxes correspond to the 25th (q_1) and 75th (q_3) percentiles of the normalized compliance values, respectively. The whiskers, represented using dash black lines, extend to the data points of the highest and lowest compliance values, excluding outliers, which are marked using the red '+' symbol. The criterion for classification of a point as an outlier requires it to be greater than $q_3 + 1.5(q_3 - q_1)$ or lower than $q_1 - 1.5(q_3 - q_1)$.

The plots show that the spread of the compliance values and the variety of the obtained topologies grows as the aspect ratio decreases. It seems that if the number of MMCs is low compared to the complexity of the optimal topology, the optimizations end much more frequently in the local optima, leading to the relatively high variance of topologies. Anyway, for all of the cases, over half of the designs exhibit performance differing from the median by less than 2%. Therefore, although the obtained topologies are frequently qualitatively different, they have similar structural performance. It is both a weakness and strength of the proposed approach. On the one hand, the optimization results are not repeatable and each optimization might result in slightly different structural performance. In practical context, however, this is not so important, since in general, one wants just to improve the performance and not to find the global optimum. On the other hand, having a variety of different designs can be very beneficial in engineering practice. TO is meant to be used in the conceptual phase of the part design. At that stage, it is usually very difficult to define all objectives and constraints precisely. Therefore, having multiple designs gives the designer the possibility to choose between different options once the optimization criteria change at the later stages of the product development process.

In general, the idea of generating multiple designs for a given optimization problem points in the

Table 7.5 The most frequent topologies for the cantilever beam case with 50% volume fraction and aspect ratios of 1:1, 1:2, and 1:3. For each aspect ratio, three most often appearing design types are presented and the frequency with which they were obtained is specified.

Aspect ratio					
1:1		1:2		1:3	
Design	Freq.	Design	Freq.	Design	Freq.
	90%		57%		23%
	3%		17%		43%
	7%		7%		7%

direction of set-based approaches, where a set of attractive and distinct designs is derived via the numerical optimization (Fender et al. (2016, 2017)). This contrasts with the idea of point-based design, which does not take into account the fact that optimization targets and constraints themselves are subject to a development process. The set-based approaches, allowing for an exploration of different design concepts and interaction with the designer, are gaining more and more attention recently. With commercial solutions such as Autodesk Generative Design (Autodesk (2018)), the companies are trying to introduce TO methods in the industry by providing engineers with a large variety of concept structures that can be chosen according to their preferences. We believe that the variety of the structures generated by EA-LSM can be used in a similar way, either through the simple multi-start strategy, as demonstrated in this work, or by utilizing more sophisticated approaches from the field of multi-modal (Fender et al. (2016)) and multi-objective optimization (Duddeck (2008); Aulig et al. (2018); Wilson et al. (2001)).

Choice of the initial design

In Section 5.3.2, a procedure for creating the initial population of individuals for EA-LSM was proposed. We argued that the initial distribution of individuals in the search space is very important and can strongly influence the performance of the algorithm as well as the quality of the final solutions. The goal of this section is to confirm those statements and test the proposed initialization method against alternative approaches. Very strong dependency of the optimized design on the

choice of the initial ground structure was one of the main drawbacks of the GSAs (Fredricson et al. (2003); Pedersen (2003); Torstenfelt and Klarbring (2007)), which have some similarities to EA-LSM, and limited significantly their use in practice. Therefore, since a good optimization method should exhibit certain invariance of the final solutions w.r.t. different initial configurations, the robustness of EA-LSM in this context is investigated.

The analysis is based on a comparison of final designs obtained for three different initial layouts of the level set basis functions – diagonal (reference), orthogonal, and a filled one, resulting in a design domain entirely occupied by material. The initial configurations are depicted in Figure 7.6. In contrast to the previous section, only the transverse bending test case is considered here.

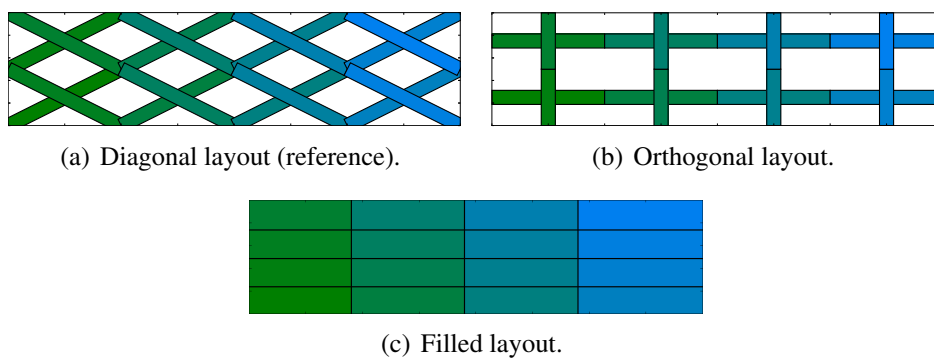




Figure 7.6 Illustration of the three initial layouts of MMCs used for the sensitivity study (Bujny et al. (2017a)).

In order to enable statistical investigations, for each layout, 30 optimization runs are carried out. In all of the cases, CMA-ES(7,15) is used as the optimization algorithm. The performance of the optimized designs is visualized in the box plots shown in Figure 7.7.

Table 7.6 Two main topology types obtained with EA-LSM and the frequency with which they occur in 30 optimization runs for three different initial layout types (Bujny et al. (2017a)). The compliance values were normalized with the initial value of the reference (diagonal) case.

Topology type	Compliance	Frequency of topology types w.r.t. layouts		
		Diagonal	Orthogonal	Filled
	0.4276	90%	47%	3%
	0.4527	10%	27%	7%

From the point of view of structural mechanics, the initial design corresponding to the diagonal (reference) layout of MMCs, is almost nine times stiffer than the one with orthogonal layout, resulting in a considerably better starting point for the optimization. In contrast, although the design initialized with the filled layout has the highest possible stiffness, it violates strongly the

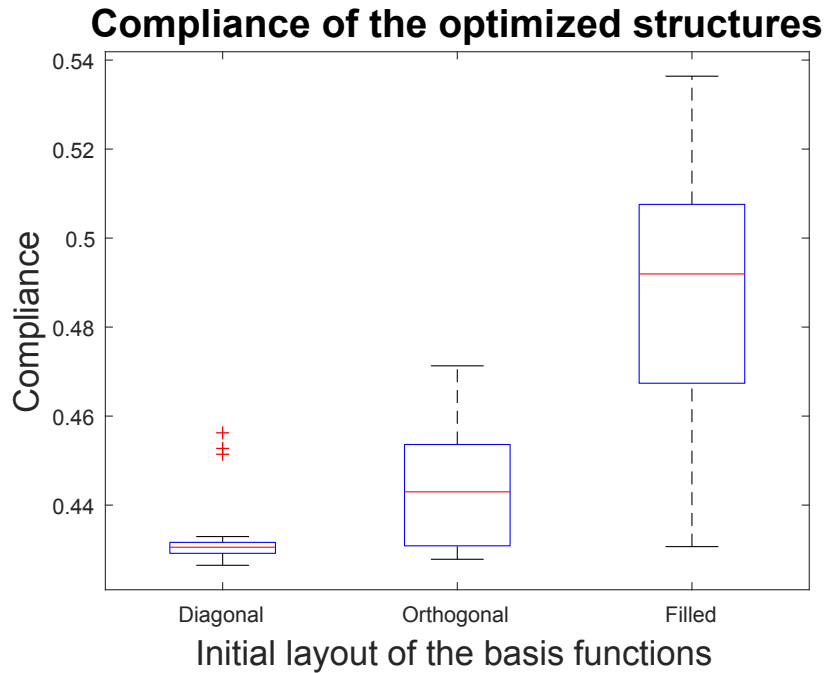


Figure 7.7 Box plots based on 90 optimization runs of the EA-LSM for the linear elastic transverse bending problem (Bujny et al. (2017a)). For each of the three initial layouts of MMCs, the compliance at the end of the optimization was taken to construct the plots. The compliance values were normalized with the initial value of the reference (diagonal) case. The line in center of each box indicates the median, while the top and bottom edges of the boxes correspond to the 25th (q_1) and 75th (q_3) percentiles of the normalized compliance values, respectively. The whiskers, represented using dash black lines, extend to the data points of the highest and lowest compliance values, excluding outliers, which are marked using the red '+' symbol. The criterion for classification of a point as an outlier requires it to be greater than $q_3 + 1.5(q_3 - q_1)$ or lower than $q_1 - 1.5(q_3 - q_1)$.

volume constraint, leading to very high values of the penalty, contributing heavily to the cost function (Equation (5.28)). Consequently, the optimizations initialized with the orthogonal and the filled layouts lead much more often to local optima, as shown in Table 7.6. This explains to great extent why the average performance of the designs generated from the diagonal layout is better³ and illustrates certain sensitivity of EA-LSM w.r.t. the initial configuration. One should note also how the variance of the compliance rises for the orthogonal and filled layout, which speaks again in favor of the diagonal layout.

Although EA-LSM seems to be quite sensitive to the choice of the initial layout of MMCs, in all of the optimizations considered in this section, it never failed to find structures reasonable from the mechanical point of view. In fact, as shown in Figures 7.8, 7.9, and 7.10, for each of the initial layouts, it was able to find the best design type. This shows that, as long as the choice of the initial configuration is not particularly unfavorable (e.g. when all MMCs reside initially in a small part of the design domain), the global optimum can be found provided that the number of optimization runs is sufficiently high.

To summarize, although EA-LSM exhibits some similarities to the GSAs, it allows the MMCs to

³ This statement was verified with the Wilcoxon rank sum test. The null hypothesis of equal medians was rejected at the 1% significance level.

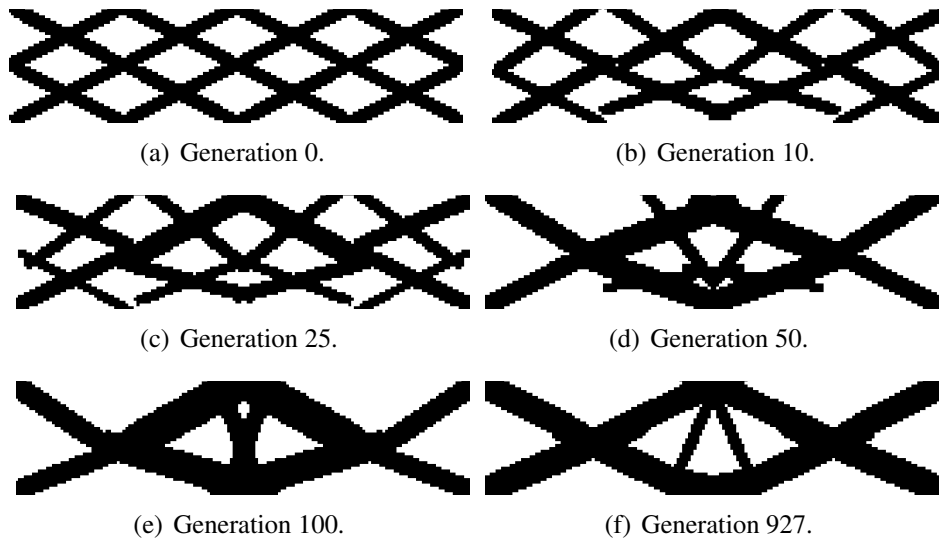


Figure 7.8 Evolution of the best individual during the optimization process for the diagonal initial layout (Bujny et al. (2017a)).

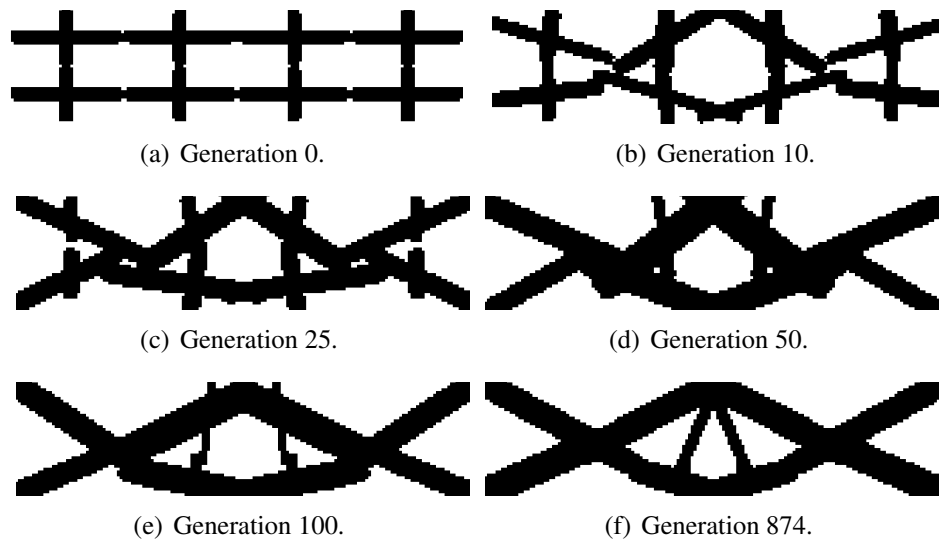


Figure 7.9 Evolution of the best individual during the optimization process for the orthogonal initial layout (Bujny et al. (2017a)).

deform and move inside the design domain. This property, together with the ability of EAs to find global optima, helps to reduce the dependency of the final solution on the initial configuration. Nevertheless, some dependency still exists and the probability of finding global optimum can decrease significantly once an unfavorable initial configuration is used. The initialization based on the diagonal layout, described in Chapter 5, performs the best in the considered scenario and seems to be a reasonable choice in general case.

Effect of the mesh size

The dependency of the final optimization result on the mesh resolution is one of the main problems in the density-based methods, e.g. SIMP (Bendsøe and Sigmund (2004)). They stem from

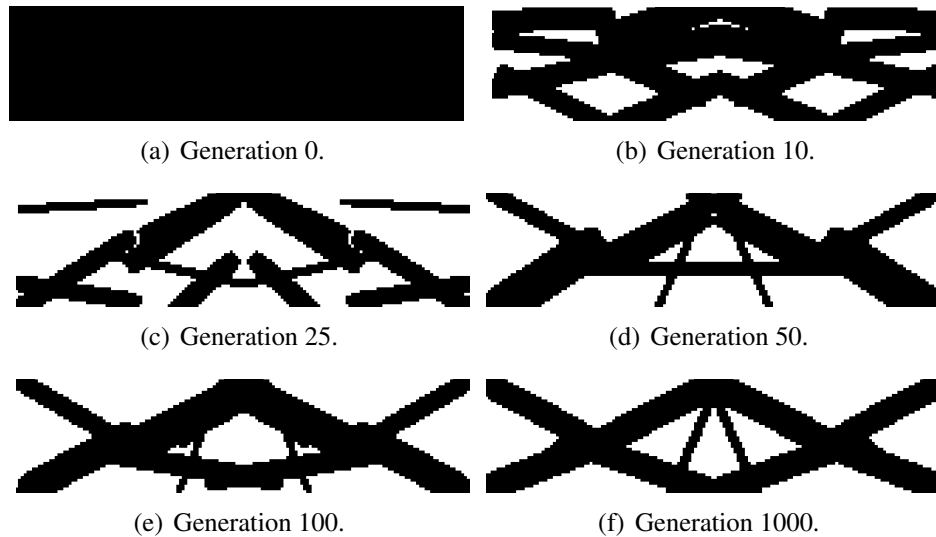


Figure 7.10 Evolution of the best individual during the optimization process for the filled initial layout (Bujny et al. (2017a)).

the direct coupling of the FE discretization with the parametrization of the optimization problem. In order to handle mesh-dependency, density-based methods employ often such techniques as perimeter control and density or sensitivity filtering (Bendsøe and Sigmund (2004)).

Since in EA-LSM the parametrization is decoupled from the discretization used in the simulation, EA-LSM should not exhibit the problem of mesh-dependency. To verify this hypothesis, 30 optimization trails for 3 different mesh sizes were carried out. As in the previous section, we used CMA-ES(7,15) as the optimization algorithm. The best designs out of 30 for each mesh resolution are depicted in Figure 7.11.

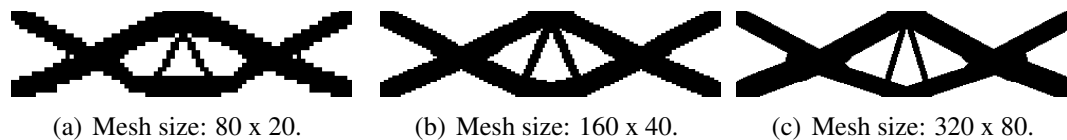


Figure 7.11 The best designs obtained in 30 optimization runs with 3 different FE mesh resolutions (Bujny et al. (2017a)).

The optimized designs show very good invariance w.r.t. the mesh resolution. On the one hand, finer meshes allow for more accurate FE modeling, but do not result in more detailed structures. On the other hand, with a very coarse mesh, the method is still able to come up with a qualitatively similar topology, with small differences resulting from the less accurate FE model. This is a desired behavior of a TO method and is one of the advantages of EA-LSM over density-based methods.

Moreover, due to the macroscopic parametrization with MMCs, the checkerboard problem known from the state-of-the-art density-based methods (Bendsøe and Sigmund (2004)) does not appear. In principle, checkerboard patterns might develop for a very high number of MMCs provided that

the mesh size is kept on a low level. This would make the parametrization to some extent similar to the density-based approaches, with a number of MMCs being close to the number of finite elements. However, due to the high computational effort of EAs, rising significantly with the number of design variables, we assume that EA-LSM should use a low-dimensional representation with sufficiently fine FE discretization. This would allow for both, efficient use of EAs and an accurate modeling of the physical behavior of mechanical structures. Therefore, such considerations are of limited practical interest, and EA-LSM could be assumed not to exhibit any major mesh-dependency problems.

Structural attainability and influence of the number of basis functions

According to Bendsøe and Sigmund (2004), it is well-established that TO problems with 0-1 or SIMP formulations have no solutions in a continuum setting, i.e. for a theoretical non-discretized problem. In general, by introducing more holes into a structure, while keeping the volume fraction constant, one should be able to increase structural performance of the design. In the limiting case, this process would lead typically to a development of anisotropic microstructures that cannot be modeled with the original isotropic design description. In discretized computational implementations, the existence of solutions is trivial due to a finite number of design options (Bendsøe and Sigmund (2004)), but the problem described above embodies itself in a form of a rising complexity of structures once more detailed modeling on the representation level is allowed. In case of SIMP, this would correspond to larger number of holes for finer mesh sizes, as illustrated in Figure 7.12. In contrast, in EA-LSM, the complexity of the structure can be controlled explicitly with the number of MMCs and their minimal thickness, irrespectively of the used FE mesh resolutions. In practical engineering applications, where structures of low complexity are usually preferred, this is a very useful property, since it allows for accurate modeling of physical behavior, while keeping the number of design variables on a low level.

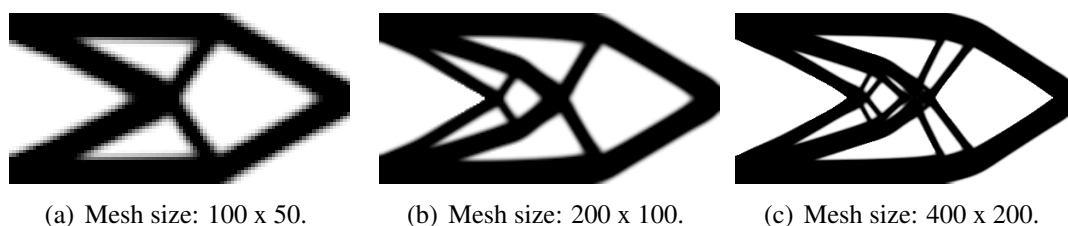


Figure 7.12 Dependency of the final design obtained with SIMP on the mesh resolution. Settings of the SIMP approach as described in Appendix B.1.

Figure 7.13 illustrates how more complex structures for the transverse bending problem can be obtained with use of 64 MMCs. In comparison to the best design obtained with 16 MMCs (Figure 7.4), the best topology modeled with 64 MMCs has ca. 1.65% lower compliance. One should note, however, that even for the problem with 64 MMCs, local optima corresponding to the best design

with 16 MMCs can be found during the optimization process. Therefore, by specifying higher number of MMCs, one can only limit the complexity of the design.

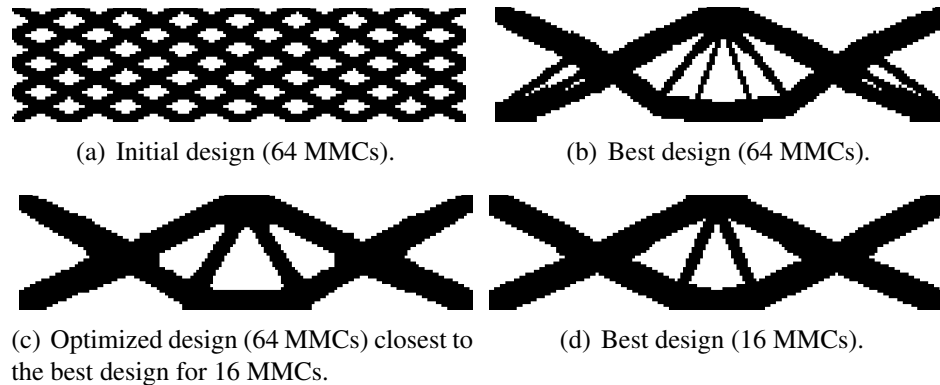


Figure 7.13 Designs obtained in 30 optimization runs with 64 and 16 MMCs (Bujny et al. (2017a)).

The best design obtained with EA-LSM with 64 MMCs no longer matches the reference design obtained with SIMP (Figure 7.4(b)). In order to obtain structures of higher complexity with SIMP, one can increase the resolution of the mesh and change the filter size. Figure 7.14 presents the results of TO with SIMP for a mesh resolution of 320×80 and different filter sizes. The closest EA-LSM design (second best) is depicted in Figure 7.14(d).

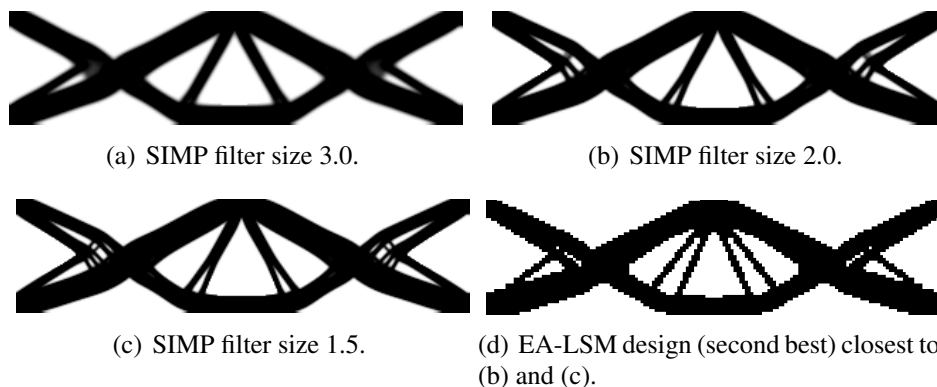



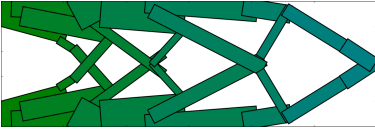
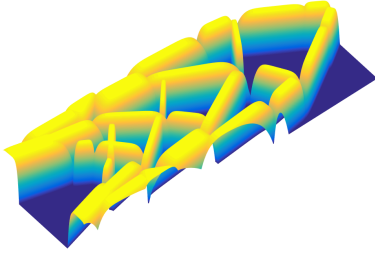

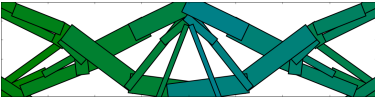
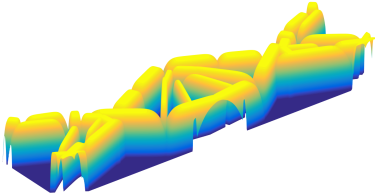
Figure 7.14 Optimization results with SIMP for the transverse bending case and mesh resolution of 320×80 and the closest design obtained with EA-LSM (Bujny et al. (2017a)).

The problem of complexity in structural design is strictly associated with the concept of structural attainability, being the ability of the method to represent the geometric features of a given design. Table 7.7 shows how more complex topologies are represented with use of EA-LSM with 64 MMCs.

Computational costs and scalability of parallel EA-LSM

The previous sections discussed mostly the issues associated with the quality of the designs obtained with EA-LSM. However, one of the most important factors in real-world optimization prob-

Table 7.7 The best designs out of 30 optimization runs for the cantilever beam and transverse bending case showing the topological attainability of EA-LSM.

Material distribution	MMC layout	LSF
		
		

lems is the computational cost of a method. This aspect is particularly important for crashworthiness optimization, where each evaluation can take several hours.

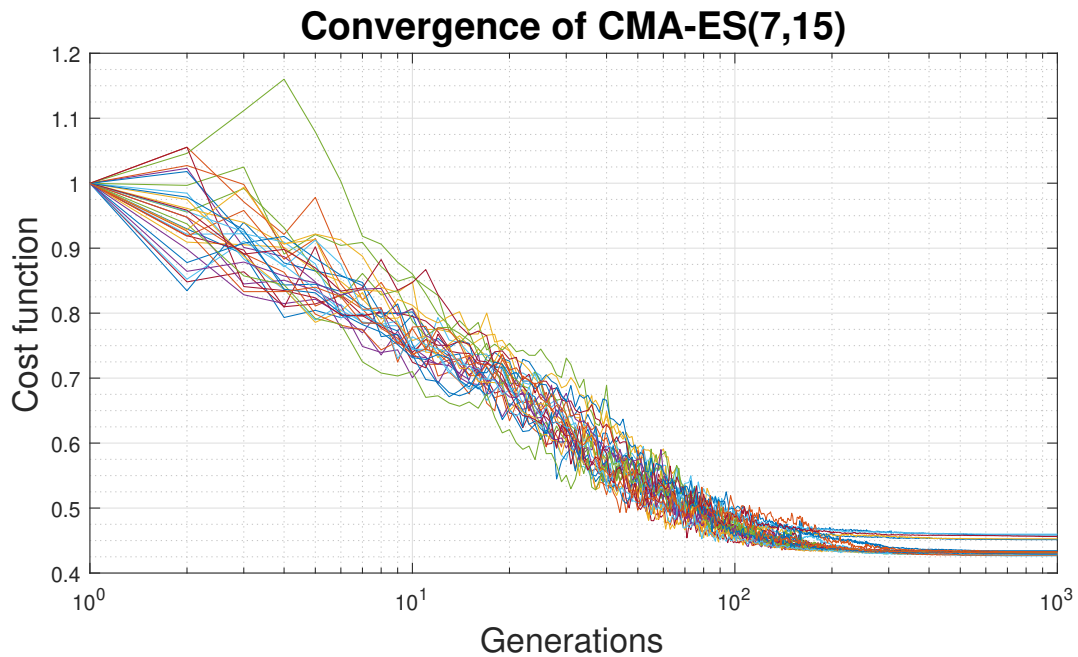


Figure 7.15 Convergence of the cost function for 30 optimization runs of CMA-ES(7,15) for a 40-dimensional transverse bending optimization problem.

Figure 7.15 shows the convergence of the 30 optimization runs for the transverse bending case with 16 MMCs. As the optimization algorithm, CMA-ES(7,15) was used. In most of the cases, the algorithm converges between 100–200 generations, corresponding to 1500–3000 FE evaluations. It is relatively high compared to the gradient-based methods – e.g. SIMP (Andreassen et al. (2011))

needs ca. 40 FE analyses for an optimization problem with the same mesh resolution, resulting in a qualitatively similar design. In the context of crashworthiness TO, also heuristic methods, e.g. HCA or ESO/BESO, are very popular. In terms of computational costs, they require 1–2 orders less FE evaluations than EA-LSM, as well. The significantly higher computational costs of EA-LSM, however, can be justified as follows:

1. EA-LSM is meant to be used for problems where gradient information is not available. This holds especially for inherently non-differentiable problems, e.g. in crashworthiness, which are characterized by high levels of numerical noise, discontinuities, and bifurcations.
2. The method is very general. It can be used with any quantifiable objective functions and constraints. In particular, it could be also used to solve problems in various subdomains of structural mechanics. This contrasts with both gradient-based and heuristic approaches, which are dedicated exclusively for specific cases.
3. EA-LSM is a black-box optimization method. This means that it can be easily used with any commercial software, without access to the source code.
4. In terms of generations, performance of EA-LSM and gradient-based as well as heuristic approaches is very similar. Provided that sufficient resources are available, the computations in each generation can be done in parallel, justifying the comparison of the methods in terms of iterations (generations) and not evaluations. In fact, the convergence velocity of EA-LSM (in terms of generations) can be explicitly controlled by choosing appropriate population sizes.
5. EAs outperform gradient-based methods in finding global optima. Although this is of very limited practical relevance, it can be an important aspect in some cases.

In fact, most of the arguments listed above would be true for any TO method based on EAs. However, it seems that the representation employed in EA-LSM, together with the use of ESs, allows EA-LSM to outperform by far the alternative non-gradient TO methods. For instance, the method proposed by Hamza et al. (2013) needs almost up to one million evaluations to converge using a GA for problems of similar dimensionality (46 and 52), not to mention the methods based on grid parametrizations.

So far, we assumed to deal with a 40-dimensional optimization problem, corresponding to a parametrization with 16 basis function and a one symmetry condition. As shown in the previous sections, such parametrization is usually sufficient to grasp the complexity of a wide spectrum of 2D structures. However, if better-performing structures of higher complexity are targeted, one would have to increase the number of MMCs. Similarly, to obtain a coarser concept structure, the number of MMCs can be decreased. As pointed out earlier, those changes on the representation level would influence the dimensionality of the optimization problem and consequently, will affect the cost of using EA-LSM. Figure 7.16 shows the average number of evaluations required for CMA-ES (with default population sizes (Appendix B.2)) to converge for parametrizations with 4, 16, and 64 MMCs, corresponding to problem dimensionalities of 10, 40, and 160, respectively.

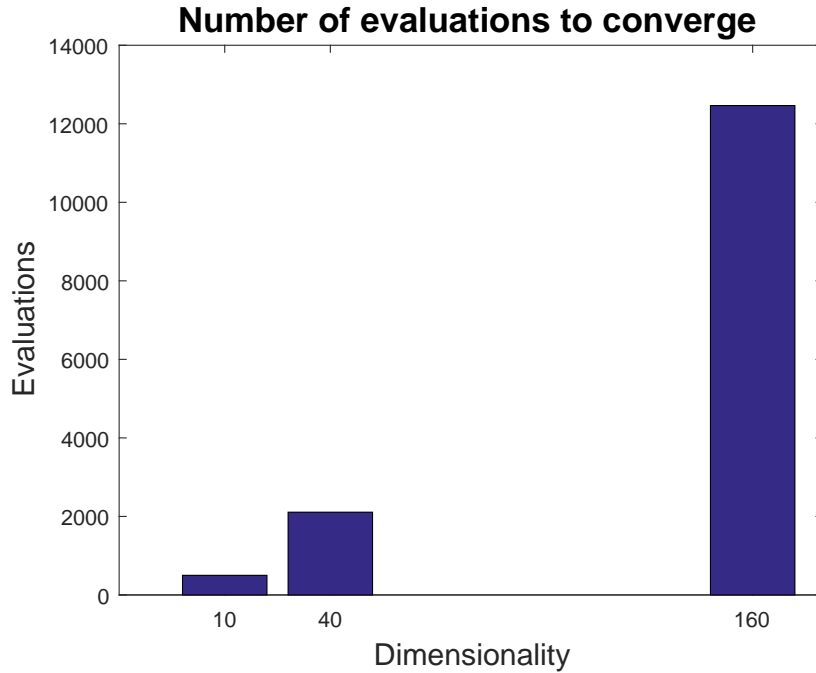


Figure 7.16 Average number of evaluations necessary for EA-LSM to converge for different dimensionalities of the optimization problem, calculated based on 30 independent optimization runs for each case. In all of the cases, CMA-ES(μ, λ), with population size determined according to the rule specified in Appendix B.2, was used as the underlying evolutionary optimization algorithm.

One can note that the computational cost of the method grows significantly as the problem dimensionality rises. In many situations, it can already be prohibitive to carry out TO with EA-LSM with such a detailed representation. However, as mentioned in the point 4, one can compensate the growth of computational costs due to the problem dimensionality by choosing an appropriate population size and parallelizing the computations.

A rule of thumb for standard ESs states that the convergence velocity of the algorithm depends as $c \sim \mu \ln\left(\frac{\lambda}{\mu}\right)$ on the number of parents μ and offspring λ (Bäck (2014)). Provided that the whole offspring population is evaluated in parallel, this allows usually for compensation of the drop in the convergence velocity, which, for a fixed population size, depends on the problem dimensionality n as $c \sim \frac{1}{n}$. Taking into account virtually linear scalability of EAs due to the very low communication effort⁴, it should be possible to compensate the doubling of the problem dimensionality by increasing both μ and λ by a factor of 2. Similarly, it should be possible to scale linearly the convergence velocity of EA-LSM by scaling μ and λ and keeping the problem dimensionality constant.

All in all, although the computational costs of EA-LSM are relatively high, it offers a great generality and can deal with the problems the other methods cannot. On the other hand, in the era of parallel computing, almost perfect scalability of EAs gives EA-LSM a huge advantage, making its performance in terms of computational time comparable with the gradient-based methods when sufficient computational resources are available. Currently, however, this property can be utilized

⁴ For ESs, only the values of the cost function and design variables have to be communicated between the processors.

only to limited extent in the real-world scenarios. As a result, alternative solutions to decrease the computational effort of EA-LSM should be found. In the next sections, this problem is therefore addressed with use of the methods proposed in Chapter 6.

7.1.2. Hybrid evolutionary level set method (H-EA-LSM)

In this section, the idea for improving the performance of EA-LSM via utilization of approximate gradient information, as described in Section 6.1, is evaluated based on numerical experiments. In case of linear elastic problems, considered in this chapter, the analytical sensitivities of objectives and constraints are available. Therefore, this section aims for evaluation of the benefits coming from the hybridization of EA-LSM with gradient-based search in an ideal case, when the gradients can be computed exactly. Similarly, in the second part of this section, the ability of the ML-based approach for gradient approximation to learn the gradient model from the sampled data is verified.

Evaluation of H-EA-LSM using analytical sensitivities

To evaluate the effectiveness of the proposed approach for gradient-enhanced evolutionary search, its performance is compared with EA-LSM using standard ES(20,100) and CMA-ES(8,17), as well as a simple gradient-based approach – Steepest Descent (SD) method (Rao (2009)). All of the algorithms used in this section are summarized in Table 7.8. For the purposes of a statistical evaluation, 30 optimization runs were performed for each of the listed algorithms except from SD, which yields always the same results due to its deterministic character.

Table 7.8 Optimization algorithms used for evaluation of H-EA-LSM.

Method	Description
SD	Steepest Descent optimization method (reference).
EA-LSM (ES)	Standard ES(20,100) with 1 step size.
EA-LSM (CMA-ES)	Covariance Matrix Adaptation (8,17) ES.
H-EA-LSM	Hybrid ES(20,100) with 1 step size and 10% of improved individuals.

As a test case, the cantilever beam problem with an aspect ratio of 1:1 (Figure 7.1(b)) is used. The reference structure used as a starting point for SD as well as for generation of the initial population of individuals in EA-LSM and H-EA-LSM is shown in Figure 7.3(b).

To allow for a more convenient comparison of the methods, both gradient-based and evolutionary

approaches target minimization of the following cost (Lagrangian) function:

$$f(\mathbf{z}) = C(\mathbf{z}) + \lambda_V V(\mathbf{z}), \quad (7.2)$$

where $C(\mathbf{z})$ stands for compliance, $V(\mathbf{z})$ is the total volume of the structure, λ_V is a Lagrange multiplier for volume, and \mathbf{z} denotes the vector of design variables. Please note that in this case the optimization problem is unconstrained and the volume fraction is enforced by incorporating a penalty factor $\lambda_V V(\mathbf{z})$ directly into the cost function. As a result, the constraint-handling techniques do not have to be used in this case, which makes the comparison between different optimization algorithms, usually utilizing different methods for enforcing constraints, considerably simpler. The exact values of the parameters used for the optimization are given in Table 7.9.

Table 7.9 Configuration of the optimization algorithms used in the evaluation of the gradient-enhanced evolutionary methods.

Parameter	Symbol	Value
Lagrange multiplier for volume	λ_V	0.02
Step length in the line search	s	0.2
Initial step size in the mutation operator	σ_{init}	0.1

Figure 7.17 presents an averaged convergence behavior of the algorithms based on 30 optimization runs. The distribution of the values of the cost function after 10, 100, and 1000 optimization iterations is shown in Figure 7.18.

The plots show that H-EA-LSM has superior convergence properties in terms of optimization iterations w.r.t. both of the methods composing it, namely SD and ES. On average, H-EA-LSM converges faster at the beginning, which is very important in practical applications, and reaches lower values of the cost function. Obviously, SD requires less cost function evaluations, where each of them involves a static FE analysis, but H-EA-LSM is meant to be used in situations when an exact gradient information is not available and SD is not applicable. Although H-EA-LSM requires 20% more FE evaluations than EA-LSM (ES) in order to find the optimal step length for the gradient improvement by the quadratic interpolation method (Arora (2012)), it is still more efficient than EA-LSM in terms of evaluations. What is more, as shown in Table 7.10, H-EA-LSM much more frequently results in the best topological concept and shows much lower variance of the cost function values for the 30 optimization runs (Figure 7.18). To summarize, H-EA-LSM does not only perform as good as its best component, but actually benefits from the synergy between ES and SD. By applying the mutation prior to the gradient improvement step, the algorithm allows for exploration of multiple starting points for the gradient descent. As a consequence, the

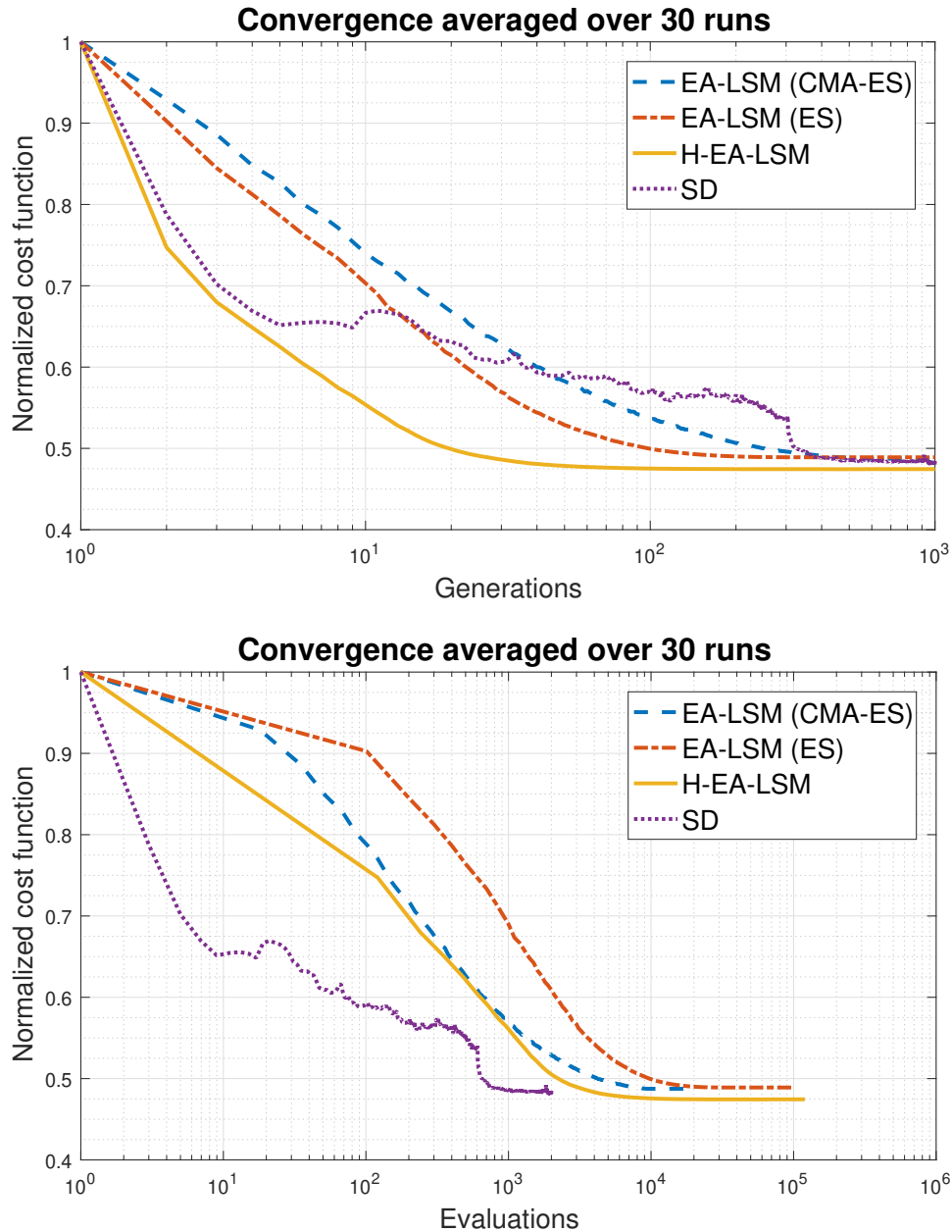
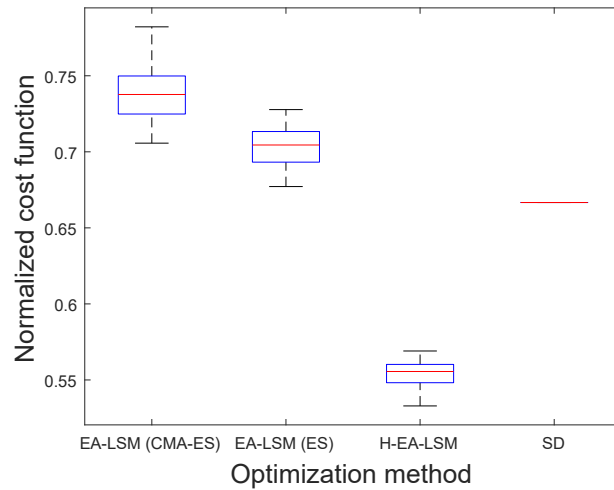
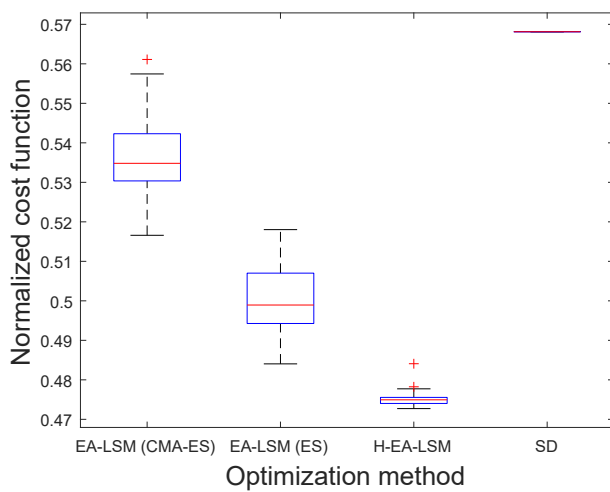


Figure 7.17 Convergence of the cost function in terms of optimization generations/iterations (top) and FE evaluations (bottom) for EA-LSM using ES and CMA-ES, H-EA-LSM, and SD. The cost function is normalized with the initial value. For all the methods except SD, the results are averaged over 30 optimization runs, which results in monotonic curves. SD run is not repeated due to a deterministic character of the method. Since SD is not a population-based method, generations in this case should be interpreted as the optimization iterations.

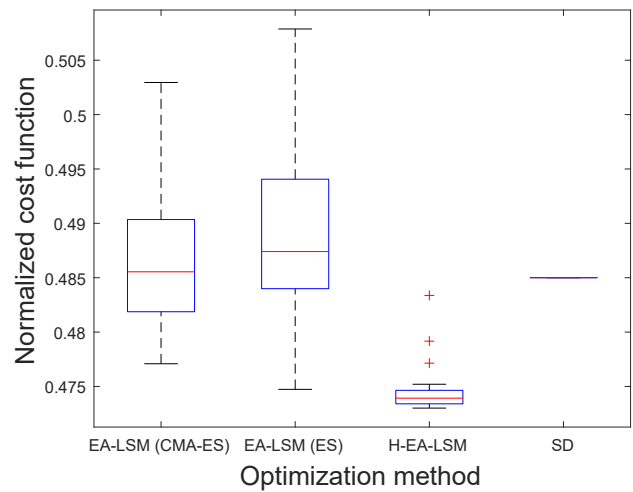
selection operator can choose between multiple individuals after a joint mutation-improvement step and the non-improved individuals. This allows for finding the best gradient-improved individuals or rejecting them in favor of the standard individuals whenever the gradient information is inaccurate. Therefore, the method offers additional speedup in the situations when the gradient information is helpful, and can still work well in the situations when it cannot rely on the gradient approximation.



(a) After 10 generations.



(b) After 100 generations.



(c) After 1000 generations.





Figure 7.18 Distribution of the cost function values after 10, 100, and 1000 iterations/generations for different evolutionary optimization algorithms and SD. The cost function is normalized with the initial value. The line in center of each box indicates the median, while the top and bottom edges of the boxes correspond to the 25th (q_1) and 75th (q_3) percentiles of the normalized compliance values, respectively. The whiskers, represented using dash black lines, extend to the data points of the highest and lowest compliance values, excluding outliers, which are marked using the red '+' symbol. The criterion for classification of a point as an outlier requires it to be greater than $q_3 + 1.5(q_3 - q_1)$ or lower than $q_1 - 1.5(q_3 - q_1)$.



Figure 7.19 Topologies optimized with use of the SIMP approach (Sigmund (2014)) (left) and a standard LSM (Challis (2010)) (right).

Finally, as shown in Figure 7.19, for the same optimization problem, the state-of-the-art methods for TO of linear elastic structures, SIMP (Sigmund (2014)) and a standard LSM (Challis (2010)), result in qualitatively similar topologies as the best design presented in Table 7.10.

Table 7.10 Four dominant topology types obtained with different optimization algorithms and the frequency with which they occur in 30 optimization runs (Bujny et al. (2016c)). The topologies are ordered starting from the best performing designs on the top to the worst ones at the bottom.

Topology type	Method			
	EA-LSM (ES)	EA-LSM (CMA-ES)	H-EA-LSM	SD
	7%	0%	80%	100%
	27%	63%	0%	0%
	10%	10%	17%	0%
	3%	0%	3%	0%

Gradient approximation by predicting sensitivities

In the situations when the gradient information cannot be estimated based on physical surrogate models, such as ESL cases for the crash problems, an ML-based approach proposed in Section 6.1.3 can be used. This section demonstrates how a simple linear regression model can be trained based on the collected gradient samples for the minimum compliance problem.

First of all, the samples of the gradients were collected for training the models. In case of linear regression models considered in this chapter, usually a limited number of samples is sufficient to obtain good prediction accuracy. Therefore, the gradient samples were collected during 80 iterations of a single run of the SD algorithm, based on finite-difference approximation of sensitivities, for the optimization problem considered in the previous section (Equation (7.2)). For a design parametrized with 16 MMCs, this resulted in a total number of $16 \cdot 80 = 1280$ samples for training of each model $\left(\frac{\partial f}{\partial p}\right)_{\Theta_p}$, where $p = x_0, y_0, \theta, l$, and t .

Taking into account only the features of the form given by Equation (6.21), five independent first order models of the following form were trained based on the collected data:

$$\frac{\partial f}{\partial p_i} \approx \left(\frac{\partial f}{\partial p}\right)_{\Theta_p} (\text{CSF}) = c_0 + c_1 \text{CSF}_1 + c_2 \text{CSF}_2 + \dots + c_{36} \text{CSF}_{36}, \quad (7.3)$$

where c_i , $i = 0, \dots, 36$ are the coefficients determined by solving the linear regression problem (Bishop (2007)). Please note that the total number of features is 36, since, for 8 degrees of freedom of the considered 4-node shell elements, there are exactly 28 different terms of the form $u_{e,i}u_{e,j}$ and 8 terms $u_{e,i}^2$. The fitting of the models resulted in R^2 values presented in Table 7.11.

Table 7.11 R-squared values calculated on the test data for the sensitivity models of compliance.

	Model				
	$\widetilde{\frac{\partial f}{\partial x_0}}$	$\widetilde{\frac{\partial f}{\partial y_0}}$	$\widetilde{\frac{\partial f}{\partial \theta}}$	$\widetilde{\frac{\partial f}{\partial l}}$	$\widetilde{\frac{\partial f}{\partial t}}$
R^2	1.0	1.0	1.0	1.0	1.0

In case of all of the models, a perfect accuracy was obtained. In fact, it can be shown (Krischer (2018)) that a linear combination of the features used in the models proposed in this section can exactly represent the analytical gradients for the compliance minimization problem; such a result is not surprising and shows the ability of the proposed method to reproduce the exact gradients based exclusively on the sampled data. Obviously, an SD optimization based on the gradient models would yield an identical topology to the one presented in the first row of Table 7.10. Similarly, an optimization using H-EA-LSM based on the gradient models would also result in the same convergence characteristics as shown in Figure 7.17.

Of course, in general case, for arbitrary objective functions, a linear combination of the features of the form given by Equation (6.21) might be not sufficient to obtain good prediction accuracy. In these cases, by collecting sufficient amount of training data and fitting more nonlinear regression models, such as SVR or ANNs, higher gradient prediction accuracy could be potentially achieved. If the amount of the training data is limited, also appropriate feature engineering for the linear models can be an effective approach to achieve high prediction accuracy. Anyway, in this work, we limit the investigations to the features of the form given by Equations (6.20) and (6.21), and use them also for modeling of the gradients for the nonlinear dynamic crash cases (Section 8.1.2).

7.1.3. Kriging-guided level set method (KG-LSM)

In this section, an alternative concept for non-gradient level-set TO, based on problem-specific version of the EGO technique, as described in Section 6.2, is evaluated on the cantilever beam problem. Since it is well-known that the performance of EGO highly decreases with the rising dimensionality of the optimization problem (Wang et al. (2016)), three test cases of different structural complexity are considered to evaluate the potential and limits of the proposed approach:

1. Representation using 2 MMCs with all parameters but the y -position (y_0) and the rotation angle (θ) fixed. Together with a symmetry condition, this results in a two-dimensional optimization

problem. Since the thickness of the MMCs remains constant, the volume constraint does not have to be imposed, and an unconstrained optimization problem can be considered.

2. Representation using 6 MMCs, with the positional parameters (x_0, y_0, θ) treated as design variables and the shape parameters (l, t) fixed. Together with a symmetry condition, this results in a total number of 9 design variables.
3. Representation using 6 MMCs, with a single symmetry condition, and all parameters subject to optimization, resulting in a total number of 15 design variables.

In order to provide statistically significant results, the performance of all the used methods is compared based on 30 optimization runs. To be able to compare different methods in a consistent way, for an optimization problem of a given dimensionality, we generate first 30 different sets of samples using DoE techniques and use them as a starting point for the considered optimization techniques. As a result, we provide the same initial conditions for all of the algorithms.

The results presented in this section are based on the master's thesis of Raponi (2017), supervised by the author of this dissertation, and the following conference and journal publications (Raponi et al. (2017, 2019a)).

Optimization problem with 2 design variables

At first, the efficiency of the KG-LSM is evaluated on a problem with 2 design variables. For simplicity, the structural feasibility constraint, enforced usually by the penalization of the EI (Section 6.2), is not used here. As a result, the performance of a pure EGO-based search can be directly compared with an unconstrained optimization with CMA-ES.

The reference design used for initialization of EA-LSM based on CMA-ES is shown in Figure 7.20. EA-LSM, as outlined in Section 5.3.2, creates the initial population of individuals by distributing the individuals around the reference design, using a normal random distribution. For a two-dimensional optimization problem, according to the rule specified in Appendix B.2, a population size of $\mu = 3$ parent and $\lambda = 6$ offspring individuals is used for the optimization.

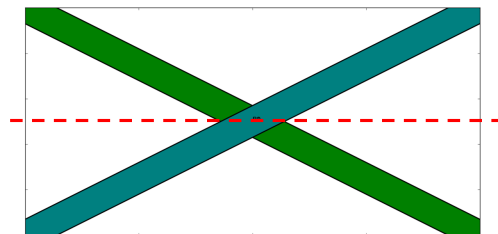


Figure 7.20 Reference design for the cantilever beam problem with 2 design variables (Raponi (2017); Raponi et al. (2019a)).

In contrast to EA-LSM, KG-LSM samples the points uniformly in the search space with use of the OLHS technique. For the first experiment, according to the recommendations of Forrester et al. (2008), a relatively low number of 20 designs is generated using the sampling plan, for a total budget of 500 FE evaluations, where the rest of the designs are created based on the maximization of the EI criterion.

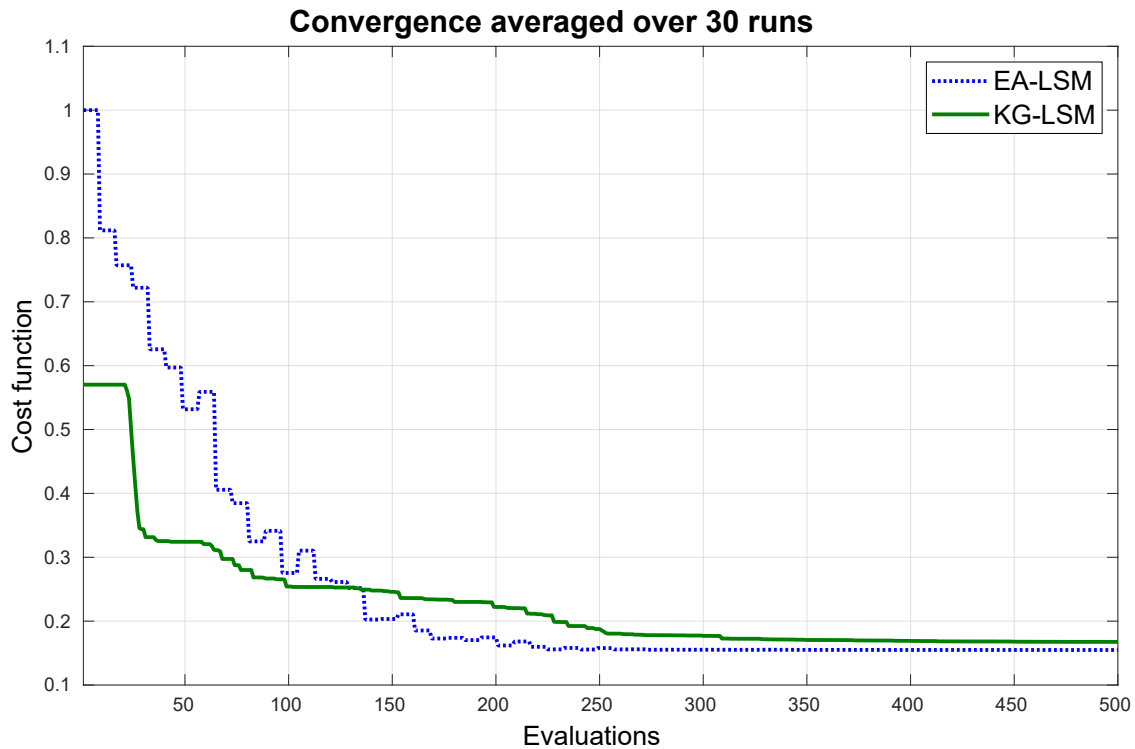


Figure 7.21 Comparison of the convergence behavior of KG-LSM and EA-LSM based on an optimization problem with 2 design variables, averaged over 30 optimization runs (Raponi et al. (2019a)).

The comparison of the convergence behavior of KG-LSM and EA-LSM averaged over 30 runs is presented in Figure 7.21. Due to the unfavorable reference design used for the initialization of EA-LSM, the initial value of the cost function corresponding to the best design resulting from the DoE in KG-LSM is already considerably better than the one for EA-LSM. After the DoE phase, the convergence rate of KG-LSM is significantly higher than the one of EA-LSM, as well. However, EA-LSM shows much better performance in the final stage of the optimization, when it is able to fine-tune the already highly-optimized design. This property has been used in our later work on hybrid strategies (Raponi et al. (2019b)), taking advantage of both, the high convergence velocity of KG-LSM in the initial phase of the optimization, and the superior exploitation performance of EA-LSM. Anyway, a comparison of the final designs obtained for KG-LSM and EA-LSM (Figure 7.22) does not show significant differences between the best topologies found with both approaches.

Please note that the EA-LSM version compared here follows exactly the description from Chapter 5, including the method used for generation of the reference design. However, in this case, as

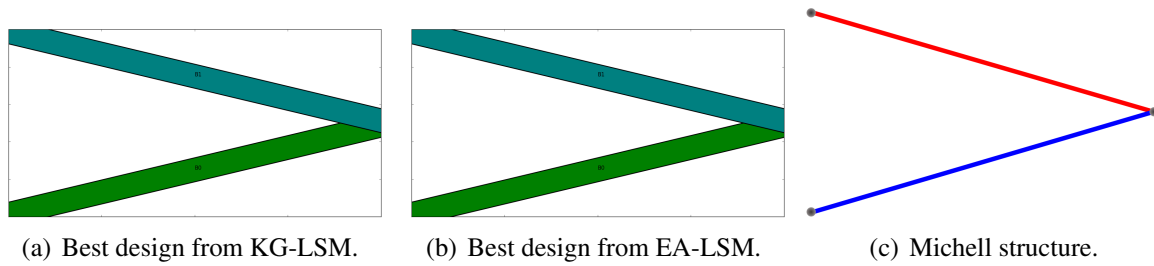


Figure 7.22 Comparison of the MMC layouts for the cantilever beam problem with 2 design variables optimized using KG-LSM and EA-LSM (Raponi et al. (2019a)). Both of the methods yield designs consistent with the corresponding Michell structure (Michell (1904)) consisting of 2 components.

pointed out above, this results in a relatively low-performing initial population. Therefore, to eliminate the influence of the unfavorable initial designs for EA-LSM, we compared the performance of both algorithms by using the best design from the DoE as a reference design for the generation of the initial population. In contrast to the previous experiments, for all 30 runs of KG-LSM, always the same DoE was used. Figure 7.23 shows the comparison of the averaged convergence characteristics for KG-LSM and EA-LSM in this case.

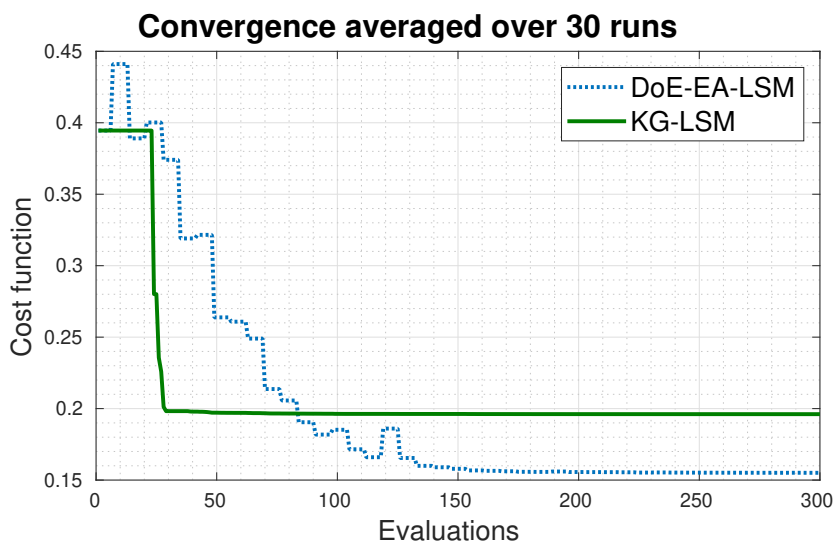


Figure 7.23 Comparison of the convergence behavior of KG-LSM and EA-LSM using the best design from the DoE as the reference (referred to as DoE-EA-LSM). Optimization problem with 2 design variables, averaged over 30 optimization runs (Raponi (2017)). KG-LSM does no longer improve after the fast convergence in the initial phase of the optimization, since it concentrates mostly on the exploration of the regions of the search space associated with high uncertainty of the surrogate model.

The results show that the KG-LSM still outperforms EA-LSM in the initial phase of optimization even when the influence of the initialization of both algorithms is eliminated. This is consistent with most of the studies on surrogate-assisted methods, where they turn out to require considerably less function evaluations than EAs for low-dimensional optimization problems (Ratle (2001) Lim et al. (2007)). Furthermore, as shown by Raponi et al. (2019a), an addition of the structural feasibility check (Equation (6.42)) in the two-dimensional static problem results in a further im-

provement of the performance of KG-LSM, which is able to outperform significantly the standard EA-LSM.

Optimization problem with 9 design variables

As the next step, an optimization problem with 9 design variables parametrizing 6 MMCs was studied. With this parametrization, the volume of the structure can change considerably by adjusting the overlap between the MMCs, and the probability of generating structurally infeasible designs rises significantly, as well. Therefore, for KG-LSM, a 50% volume fraction constraint was imposed by utilizing the CEI as the acquisition function. Furthermore, CEI was modified according to Equation (6.46), to eliminate structurally infeasible designs, as well. For the initial DoE, 300 samples were created using standard OLHS. For each of the generated samples, the structural feasibility of the corresponding design was first checked, which did not involve any FE evaluation. Finally, only the structurally feasible designs were simulated and they compose the initial set of samples used for fitting the first Kriging model. This results in an average number of 19 samples among 30 different DoE sets.

The reference design used for initialization of the EA-LSM is shown in Figure 7.24. Again, the state-of-the-art CMA-ES optimizer was used in the background. For a 9-dimensional optimization problem, the rule from Appendix B.2 yielded a population consisting of $\mu = 5$ parent and $\lambda = 10$ offspring individuals. Similarly to KG-LSM, a volume constraint is added. For EA-LSM, by default, this is done by using penalty-based constraint handling (Section 5.3.4). Since EA-LSM performs a much more local search than KG-LSM, the constraint on structural feasibility does not have to be enforced, because the selection operator can eliminate all of the disconnected designs.

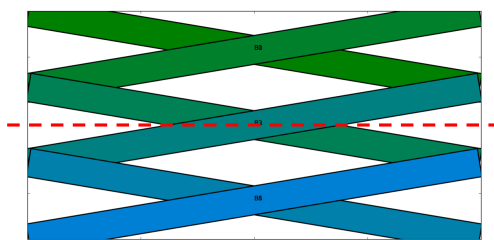


Figure 7.24 Reference design for the cantilever beam problem with 9 design variables (Raponi (2017); Raponi et al. (2019a)).

Figure 7.25 shows the evolution of the averaged compliance values in terms of FE evaluations for both KG-LSM and EA-LSM. Similarly to the two-dimensional optimization problem, KG-LSM converges much faster in the initial phase of the optimization. After ca. 250 evaluations, EA-LSM reaches on average comparable values of compliance to KG-LSM, and is able to better fine-tune the final solution probably due to its superior exploitative properties. The best designs obtained with both approaches after 500 evaluations are depicted in Figure 7.26.

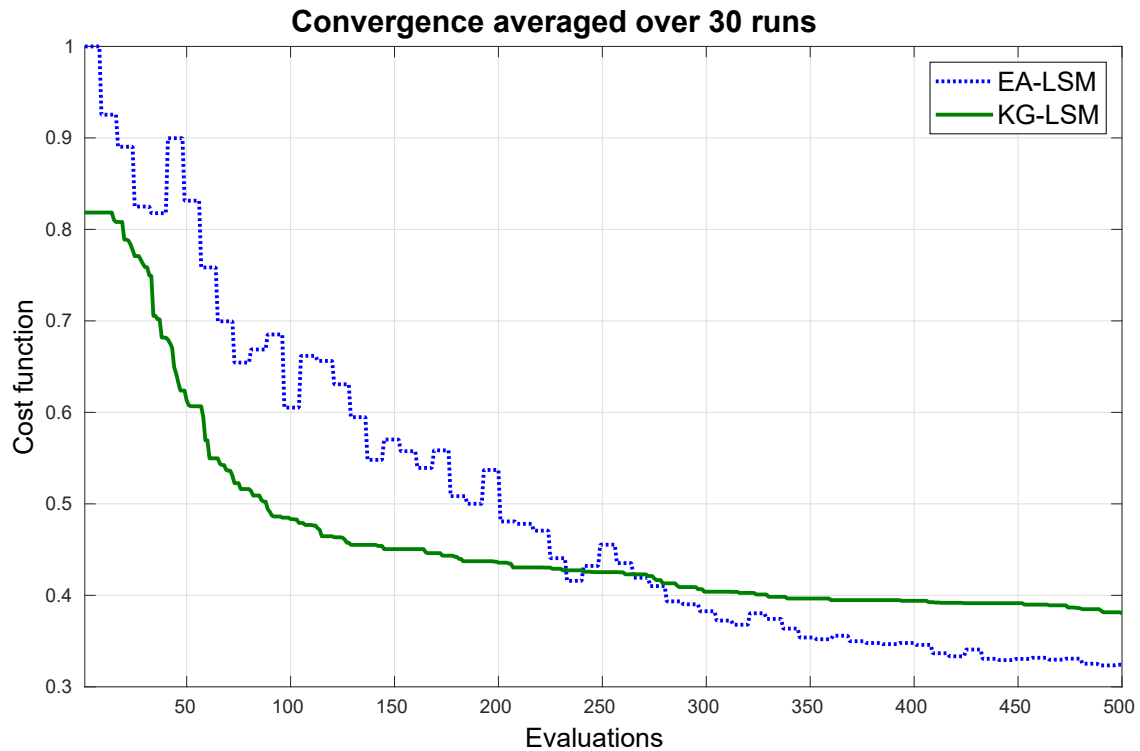


Figure 7.25 Comparison of the convergence behavior of KG-LSM and EA-LSM based on an optimization problem with 9 design variables, averaged over 30 optimization runs (Raponi et al. (2019a)).

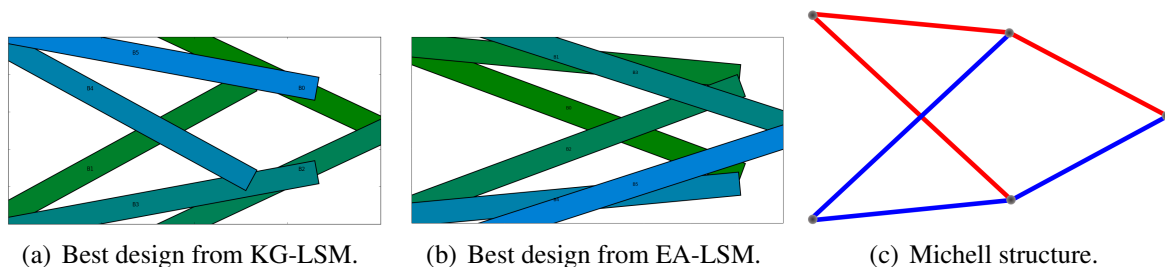


Figure 7.26 Comparison of the MMC layouts for the cantilever beam problem with 9 design variables optimized using KG-LSM and EA-LSM (Raponi et al. (2019a)). Please note that the corresponding Michell structure (Michell (1904)) for the design with 6 components might not correspond to the optimal design for the considered problem, where a part of parameters describing MMCs is kept fixed.

The experiments carried out for the 9-dimensional problem demonstrate the ability of KG-LSM to find solutions close to the reference ones for a problem with a volume constraint. The fast convergence of the algorithm in the initial phase of the optimization makes it interesting especially in the context of practical applications, where the total budget of FE evaluations is frequently heavily restricted.

Optimization problem with 15 design variables

Finally, an optimization problem with 15 design variables and 6 MMCs, subject to a symmetry condition, was considered. Similarly to the case from the previous section, both the structural feasibility as well as the volume constraint (50% volume fraction) were imposed. For all of the KG-LSM optimization runs, 600 initial DoE samples were generated and only the structurally feasible designs were evaluated, which corresponded to 29 samples on average.

For comparison, EA-LSM based on CMA-ES with $\mu = 6$ parent and $\lambda = 12$ offspring individuals (see Appendix B.2) was used. To initialize the population, the same reference design as in the problem with 9 design variables, shown in Figure 7.24, was used.

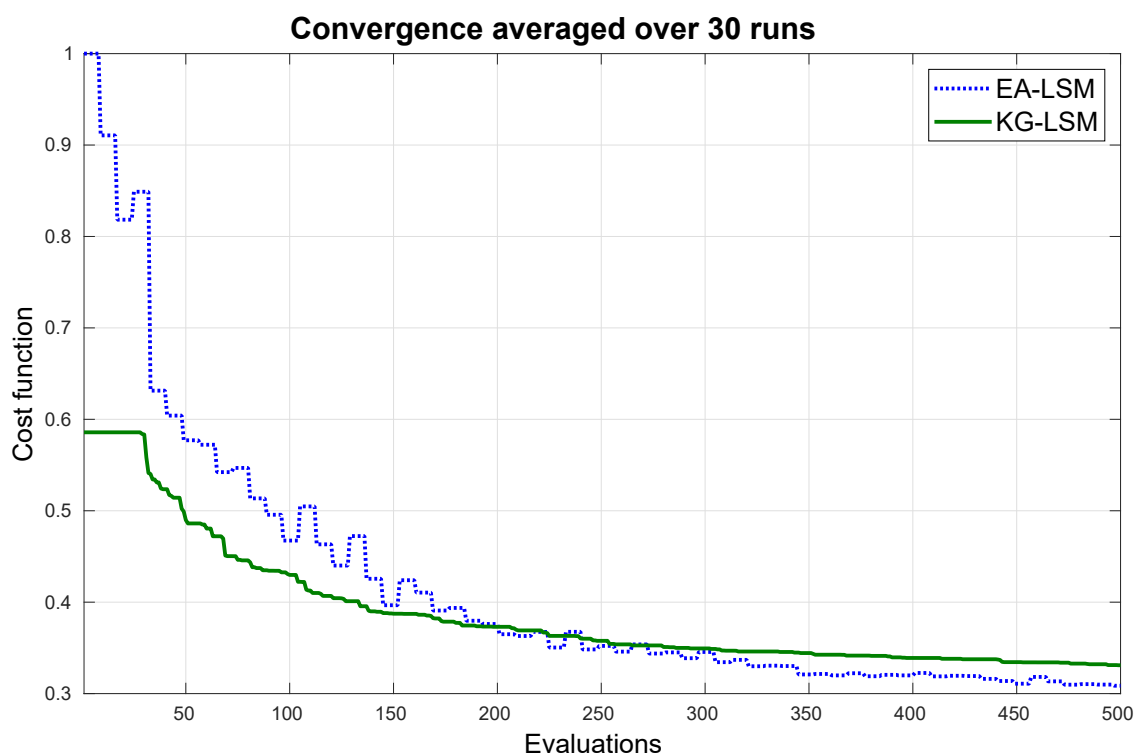


Figure 7.27 Comparison of the convergence behavior of KG-LSM and EA-LSM based on a optimization problem with 15 design variables, averaged over 30 optimization runs (Raponi et al. (2019a)).

Figure 7.27 presents the averaged convergence of KG-LSM and EA-LSM in terms of evaluations. For a 15-dimensional optimization problem, KG-LSM still outperforms EA-LSM in the initial phase of the optimization, until ca. 200 evaluations. However, the difference between the methods decreases, showing that for this problem dimensionality, at least for the compliance minimization case, KG-LSM is no longer able to offer a significantly better performance than EA-LSM. The comparison between the best designs optimized using KG-LSM and EA-LSM also shows that EGO is not able to find the topology corresponding (most probably) to the global optimum. As shown in Figure 7.28, the best designs obtained with KG-LSM and EA-LSM are qualitatively considerably different. The compliance value for the best design obtained using EA-LSM is ca.

10% lower than for the design optimized using KG-LSM.

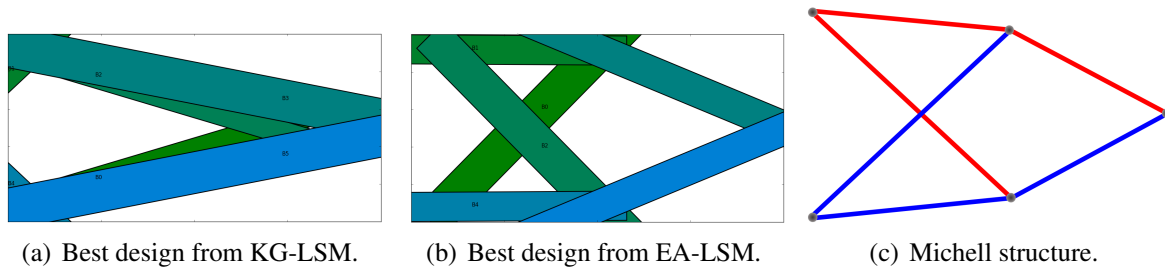


Figure 7.28 Comparison of the MMC layouts for the cantilever beam problem optimized using KG-LSM and EA-LSM (Raponi et al. (2019a)). Please note that only the design obtained using EA-LSM is consistent with the corresponding Michell structure (Michell (1904)) consisting of 6 components.

To summarize, in all of the studied cases, KG-LSM was on average significantly better than EA-LSM in the initial phase of the optimization. This is a very important property in practice, when the number of costly FE evaluations is limited. In contrast, EA-LSM performs better in the final stage of the optimization, which shows that hybrid approaches can be a promising direction for the future research (Raponi et al. (2019b); Mohammadi et al. (2015)). However, with rising dimensionality of the optimization problems, the performance of KG-LSM drops, which is related to the "curse of dimensionality" (Bishop (2007); Forrester et al. (2008)) and is a common problem in surrogate-assisted optimization. For 15-dimensional problem, the best topology obtained using KG-LSM is significantly different from the one optimized using EA-LSM and corresponds only to a local optimum. In this case, most probably, more evaluations would be needed to reach the design found by EA-LSM. An interesting research direction in the future would be also to investigate the performance of EGO algorithms for high-dimensional problems (Bouhleb et al. (2016, 2018)) within the proposed framework for structural TO. All in all, KG-LSM can be considered as an interesting alternative to EA-LSM, especially for the problems where design evaluations are very expensive, and the granularity provided by the representation relying on a very limited number of design variables is satisfactory for the designer. This is especially valid for the problems in structural crashworthiness, which are addressed with use of KG-LSM in Section 8.1.3.

7.1.4. Adaptive evolutionary level set method (A-EA-LSM)

This section demonstrates the potential of using an adaptive optimization approach based on a graph representation (A-EA-LSM), as discussed in Section 6.3. Considering the cantilever beam optimization problem with a design domain aspect ratio of 1:2, an intuitive interpretation of the working principles of the A-EA-LSM is given. Finally, the efficiency of using the learning-based topology variation prediction is evaluated by comparing the performance of the optimized structures with the performance of the designs obtained using the random approach based on 25 independent optimization runs⁵ for each approach. In all of the cases, a compliance minimization

⁵ Due to project-related time restrictions, 25 instead of 30 runs, as in the other studies in this thesis, were performed. However, this is still sufficient to carry out statistical comparison between the methods.

problem with 50% volume constraint is considered.

Table 7.12 Setup of A-EA-LSM using both learning-based and random topology variations (Bujny et al. (2018)).

Property	Symbol	Value
Number of parent individuals	μ	20
Number of offspring individuals	λ	70
Initial step size	σ_{init}	0.02
Adaptation damping	d_{ad}	2
Penalty constant for the volume constraint	P	1000
Step size threshold for topology variations	σ_{th}	0.01
Number of topology variations	N_{TV}	10
Number of edges in the initial topology	-	2
Number of LHS samples in variation step	N_{LHS}	8

For both, i.e. for the learning-based as well as for the random topology variation approach, a setup of A-EA-LSM as specified in Table 7.12 is used. The large values of both, μ and λ , are dictated mainly by the fact that in A-EA-LSM, multiple species, representing different types of topologies have to coexist together. For low values of λ , the effective size of each species would be too small to be able to optimize within a niche. In particular, similar approaches (Stanley and Miikkulainen (2002)) also use relatively large population sizes. The ratio between parent and offspring individuals was also kept above the default 1/7 value (Bäck and Schwefel (1993)), to avoid too aggressive competition between the species. The value of the damping parameter d_{ad} was chosen based on the initial experiments and the recommendations of Hansen and Ostermeier (2001).

For the prediction of topology variations, we use a fully-connected neural network classifier with 2 hidden layers. The exact specification of the dataset and the parameters used for the training process as well as the architecture of the network is given in Table 7.13. The hyperparameters of the neural network were chosen based on an extensive grid search on the number of hidden layers (N_{HL}) and the number of neurons per hidden layer (N_N). As a result, the network architecture giving the highest accuracy on the validation set, with the same number of samples as in the test set, was chosen. The test accuracy on the pairwise comparison task, as described in Section 6.3, reached 85% for the best network. This network was used later on for prediction of favorable topological variations in A-EA-LSM.

Table 7.13 Setup of the neural network classifier (MLP) for prediction of topology variations (Bujny et al. (2018)).

Property	Symbol	Value
Number of hidden layers	N_{HL}	2
Number of neurons per hidden layer	N_N	100
Activation function	–	Rectified linear (LeCun et al. (2015))
Optimization algorithm	–	L-BFGS (Nocedal and Wright (2006))
Loss function	–	Log-loss
Number of sampled topologies	N_T	317
Number of training samples	N_S	259,025
Number of test samples	N_{TS}	28,780

To understand the working principles of the proposed method, let us have a look at the evolution of different species in an exemplary optimization run with the learning-based topology variations, shown in Figure 7.29. The optimization starts from a very simple topology, composed of 2 components only, connected at one end, whose position along y -axis is chosen randomly. For the ca. 40 first iterations, the optimizer improves the design based on an 8-dimensional representation of the topology. Once the optimization converges, the step size of the best individual decreases and eventually drops below the predefined threshold $\sigma_{th} = 0.01$, which, according to the algorithm, is the criterion for introducing a new topology variation to further improve the performance of the design. At this stage, the best design is modified by inserting a topology variation, and over 50% improvement in the compliance value is achieved. Even though the modified design is considerably better than the 2-beam design, the algorithm protects the species of 4-beam structures in a niche by modifying their cost function values according to the Equation (6.48), thus giving them the chance to optimize before competing with already well-optimized, 2-beam designs. Directly after the first topology modification, the adjusted cost of the newly created design, composed of 4 MMCs, is therefore equal to $f_4^l = 0.11 \cdot 1 = 0.11$, because there is only one such individual. At the same time, the adjusted cost of the best 2-beam design is equal to $f_2^l = 0.24 \cdot 69 = 16.61$. As a result, the number of individuals representing 4-beam topologies quickly rises, since they are strongly favored by the (μ, λ) -selection operator. When the ratio of 4-beam and 2-beam topologies increases, the protection of the 4-beam designs decreases, to become neutral for the 50% ratio. At this point, only the relative compliance values of the designs in two species decide about the survival of the individuals. Since a 4-beam design is considerably better, the size of the 2-beam species shrinks. As a smaller species, it is more and more strongly protected by decreasing the

adjusted cost. However, the decreasing population size within the 2-beam species reduces the effective ratio of offspring and parent individuals within this class of topologies, which results in a gradual decrease of the convergence velocity and eventually to a deterioration of the 2-beam designs when non-elitist (μ, λ) -selection is used. In particular, for $\mu = \lambda$, the search becomes a random walk (Bäck (1996); Pearson (1905)). As a result, the population of 2-beam designs dies out around the 65th generation. In the meantime, around the 60th generation, the third species, composed of 6 MMCs, appears. This topology corresponds to the best designs obtained with both EA-LSM and SIMP for the same optimization problem (5th row, Table 7.2). This species survives during the entire optimization run and eliminates more complex designs to become the only species around generation 200. Please note that the predicted intermediate topology variations are actually reasonable from the mechanical point of view, but none of them generates a design able to take over the population.

As can be seen in Figure 7.29, the algorithm converges fast at the beginning of the optimization, partially due to the low-dimensionality of the initial search space. After introducing the first topology variation, the compliance values do not change much until the end of the optimization. The fast convergence at the beginning is a very important property in the industrial setting, where typically the number of FE evaluations is strongly limited. Moreover, A-EA-LSM gives the user the possibility to stop the optimization at any moment of time to favor simpler design concepts, which are preferable in many cases, especially when traditional manufacturing technologies are considered.

In the bottom part of Figure 7.29, the dotted red curve shows the convergence of the average step size of the individuals in the population. One should note the stabilizing effect of the derandomized step size adaptation, preventing the step size from reducing too much when the topology improves considerably due to the topology variation. The step size slightly rises each time when topology modification takes place to be able to exploit quickly the potential improvement introduced by the extended representation. As a result, the step size stays at the level above $\sigma_{th} = 0.01$ until the last topology variation. After 200 generations, the step size slowly drops, as the optimization with a single species converges.

The example discussed above illustrates the mechanisms governing the A-EA-LSM optimization based on a one out of the 25 runs used for the evaluation of topology variation methods. However, neither the population size nor the initial step size were fine-tuned to maximize the efficiency of the algorithm. To evaluate the effectiveness of the learning-based topology variations for different initial configurations, as mentioned before, the exact position of the connection point between the two beams was decided randomly, as well. As a result, the initial optimization phase takes considerably longer than for a good initial design. However, once the optimization for a 2-beam design converges, the optimization progresses very fast. From the moment of the first topology variation, within 30 generations, corresponding to 2100 finite element evaluations, the optimiza-

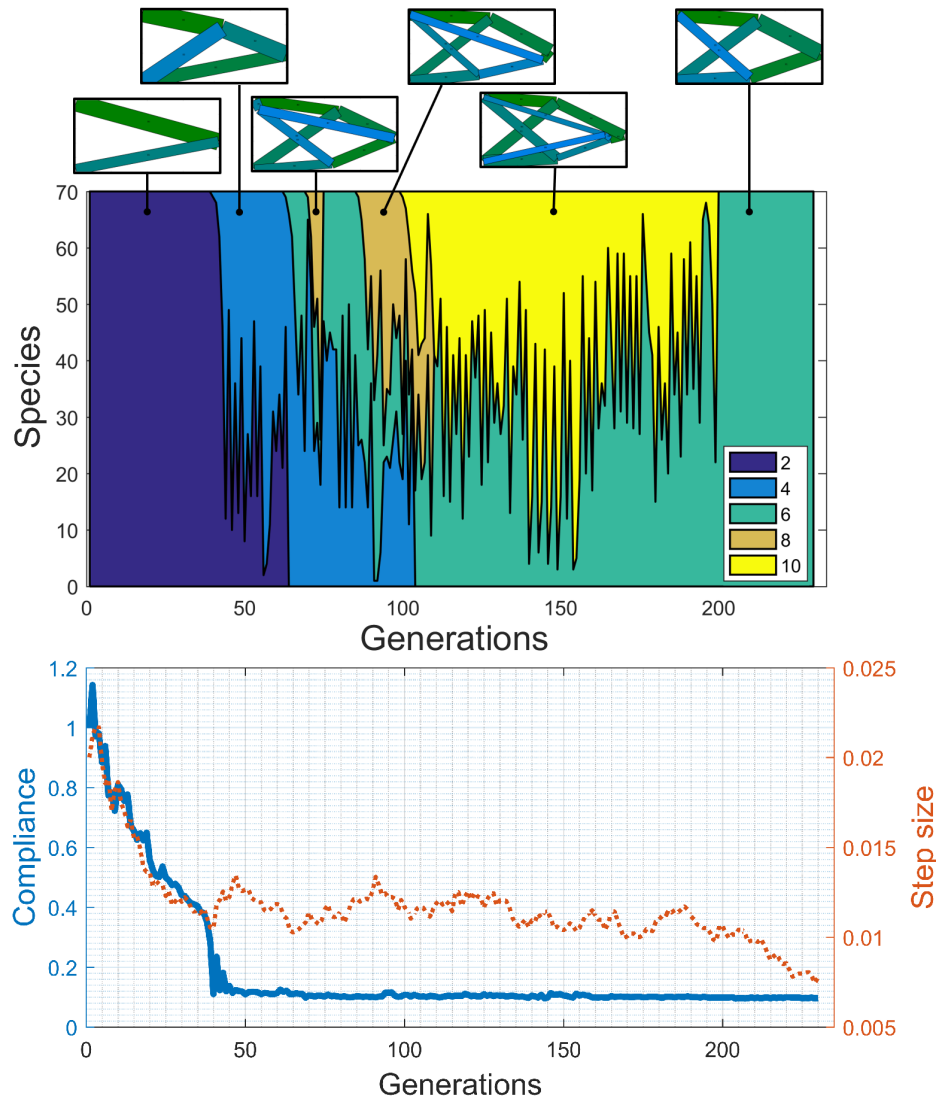


Figure 7.29 One of the runs of the A-EA-LSM with the learning-based topology variations illustrating the working principles of the method (Bujny et al. (2018)). Visualization of speciation during the optimization with color-coded species (top). Topologies are divided into species depending on the number of components specified in the legend. Convergence of the compliance normalized with the initial value (solid blue curve) and evolution of the step size averaged over the population of parent individuals (dotted red curve).

tion practically converges, with compliance values over 2 times lower than for the best 2-beam design.

One should also note that with the current formulation of the criterion for qualifying a design to a given species, relying exclusively on the number of MMCs composing the design, multiple different topologies with the same number of beams can coexist together in a niche. As a result, the final number of species appearing during the optimization is lower than the number of topology variations N_{TV} . One disadvantage of this approach is that different topologies with the same number of components are not protected against each other, which leads to domination of a single topology within a niche. Potentially, this could be improved by considering other similarity metrics for topologies, e.g. Graph Edit Distance (Sanfeliu and Fu (1983)). For readability, in Figure 7.29, only

the best performing design within each species is shown.

The example presented above used the learning-based topology variation prediction. However, the A-EA-LSM algorithm can also work with random topology variations, so it is not clear how much beneficial are the predictions of the neural network in the overall optimization process. To evaluate that, 25 optimization runs with random and learning-based topology variations were carried out. All of the optimizations started from the simple, 2-beam design and 10 topology variations were applied. In Figure 7.30, box plots of the normalized compliance values after 200 generations are shown. The learning-based approach yields clearly better performing structures, which has been verified with the Wilcoxon rank sum test. The hypothesis about equality of the medians from both datasets was rejected at the 1% significance level. Moreover, the qualitative differences between the obtained structures is shown in Table 7.14. A-EA-LSM using the learning-based topology variations was able to find the design corresponding most probably to the global optimum in 72% of cases. In contrast, in case of the random-based topology variations, only 8% of optimization runs resulted in the best topology type.

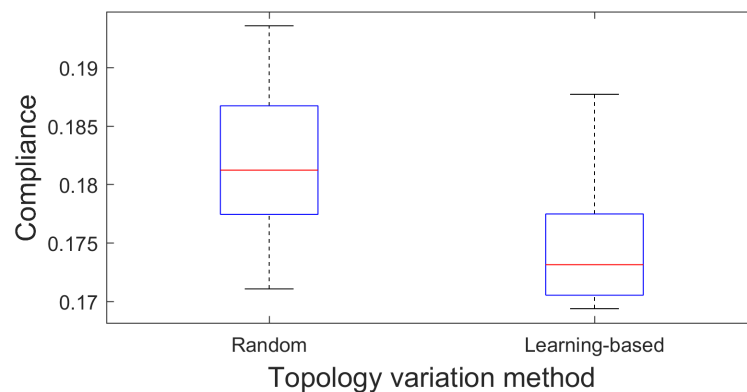






Figure 7.30 Box plots for 25 A-EA-LSM optimization runs after 200 generations comparing the performance of random and learning-based topology variations (Bujny et al. (2018)). The line in center of each box indicates the median, while the top and bottom edges of the boxes correspond to the 25th (q_1) and 75th (q_3) percentiles of the normalized compliance values, respectively. The whiskers, represented using dash black lines, extend to the data points of the highest and lowest compliance values, excluding outliers. The criterion for classification of a point as an outlier requires it to be greater than $q_3 + 1.5(q_3 - q_1)$ or lower than $q_1 - 1.5(q_3 - q_1)$.

All in all, the results for a 2D linear elastic case look promising. Learning-based topology variation works significantly better than a random approach, which shows that such rules can be learned from the sampled data, based on a fully automated process. However, the large population size, necessary for the speciation mechanism to work, is a disadvantage of the approach. Also in case of more complex designs, the number of combinations for feasible topology variations grows considerably, and a higher prediction accuracy of the neural network would be required not to converge to sub-optimal designs.

Table 7.14 Dominant types of topologies obtained in 25 A-EA-LSM optimization runs with random and learning-based topology variations (Bujny et al. (2018)). The compliance is normalized with the value for the best design. Two last columns show the frequency with which a given type of topology has been obtained.

Topology type	Compliance	Random	Learning-based
	1.0	8%	72%
	1.013	8%	4%
	1.062	40%	4%
	1.126	16%	8%

7.2. Three-dimensional topology optimization for minimum compliance

In this section, a study of the 3D variant of EA-LSM, as described in Chapter 5, is carried out. 3D TO problems are considerably more complex than the 2D ones, and as such, deserve a separate section in this work. For many methods, the main difficulty of extending them to 3D comes from their inherently 2D character, e.g. (Ortmann and Schumacher (2013)). In other cases, e.g. explicit parametrization of material boundaries with splines, the extension of the formulation to three dimensions might be very complicated and computationally challenging. In contrast, one of the key advantages of the representation based on the level set description is the simplicity, with which it can be adapted to 3D problems. The second important problem is the high computational cost which rises dramatically for most of the methods when moving from 2D to 3D. This problem emerges from both higher costs of the FEA and the cost of the optimization problem, rising due to the much higher number of design variables in 3D. The MMC-based approaches solve this problem by decoupling the representation of the design and the FE discretization. However, also for those methods, the dimensionality of optimization problem would rise, generally by a factor of 5–10, depending on the number of used MMCs. For non-gradient optimization methods this can already pose considerable difficulties.

All in all, this section aims to answer the question of feasibility of using the 3D version of EA-LSM proposed in Chapter 5. As in Section 7.1, linear elastic cases are considered to be the most suitable for the initial validation of the method.

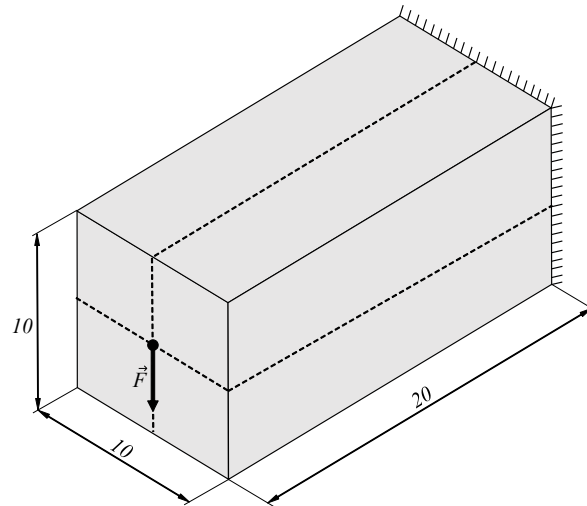


Figure 7.31 Design domain and boundary conditions for the 3D linear elastic cantilever beam test case used for validation of EA-LSM. Dimensioning in mm.

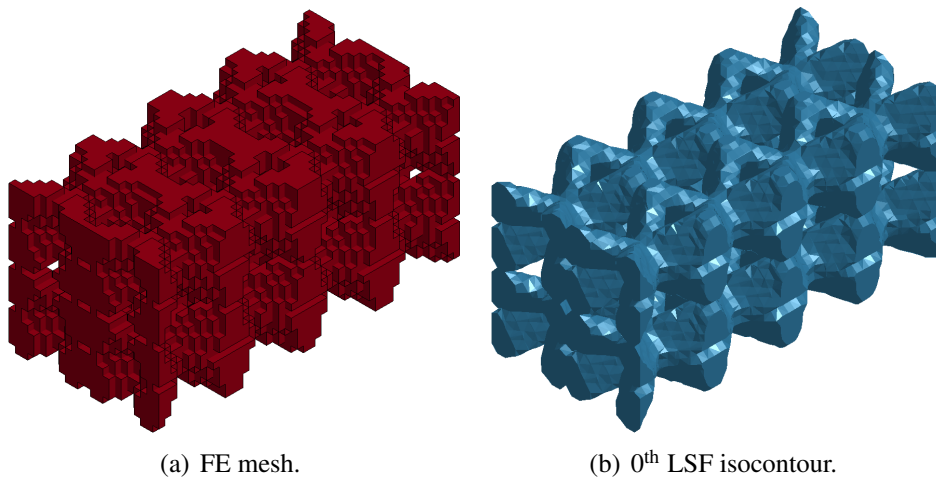


Figure 7.32 Initial design used in the 3D cantilever beam case.

7.2.1. Optimization problem

Similarly to Section 7.1, a compliance minimization problem with a volume constraint of a form given by Equation (7.1) is considered here. We focus on an analysis of a cantilever beam scenario as depicted in Figure 7.31. In order to reduce the computational costs, two symmetry planes along the longitudinal direction are introduced (dashed lines). With 64 3D MMCs, this results in a 144-dimensional optimization problem. The initial LSF and the resulting FE mesh are depicted in Figure 7.32. Finally, the exact configuration of the test case is defined in Table 7.15.

7.2.2. Results

Due to limited computational resources, the evaluation of EA-LSM is based on five independent optimization runs of a standard ES(6,39) with a single step size. Although in the previous experiments CMA-ES gave the best results in terms of the number of evaluations, we use the simpler ES with higher population sizes to improve the performance in terms of generations. This was

Table 7.15 Configuration of the 3D cantilever beam case.

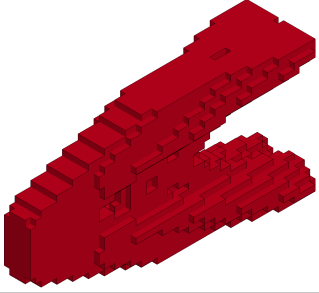
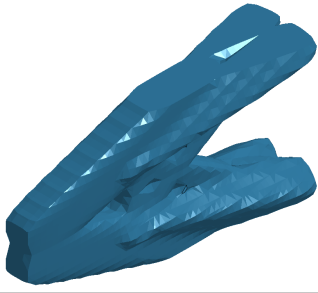
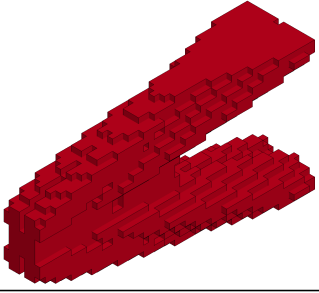
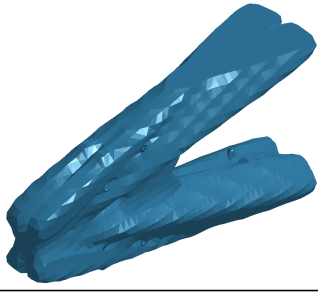
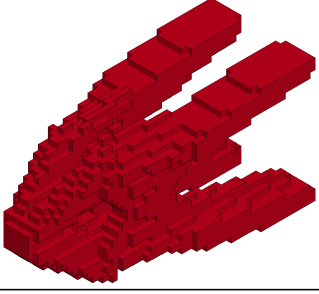
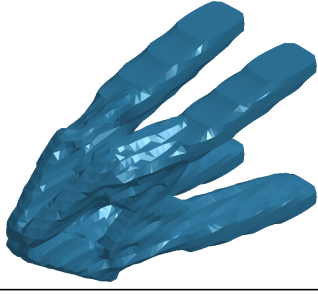
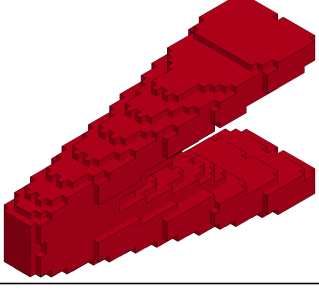
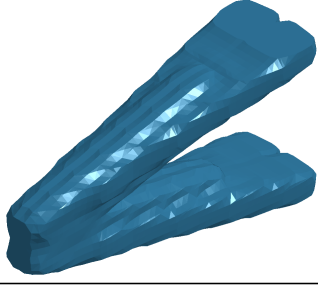
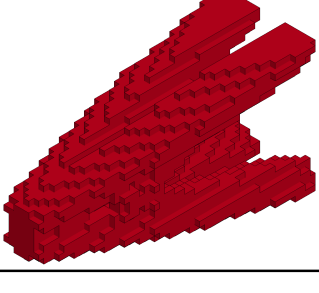
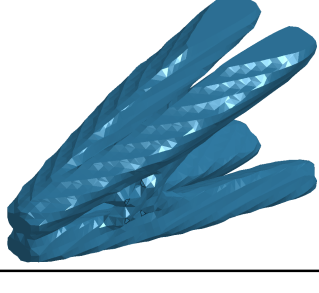
Property	Symbol	Value	Unit
Young's modulus	E	$2.1 \cdot 10^5$	MPa
Poisson's ratio	ν	0.3	-
Load	F	1	N
Required volume fraction	V_f	20%	-
Aspect ratio	-	1:1:2	-
Mesh resolution	-	25 x 25 x 50	-
Solver	-	CalculiX 2.9	-
Element type	-	Eight-node brick element (C3D8R)	-

mainly motivated by the limitation of the total run time of jobs on the used computational cluster. For similar (large) population sizes, CMA-ES gave inferior results compared to ES. The offspring population size $\lambda = 39$ was determined based on the allocated computational resources. More precisely, 5 computational nodes with 8 cores per node were used for each of the optimization runs, parallelized using MPI. This resulted in 40 cores per optimization run, where 1 core is used for controlling the optimization process, yielding 39 cores which can be used to evaluate all the individuals in a population in parallel. Using the usually recommended ratio between the number of parent and offspring individuals $\frac{\mu}{\lambda} \approx \frac{1}{7}$ (Bäck (1996, 2014)), the number of parents was set to $\mu = 6$.

Table 7.16 shows the final designs obtained in five runs of EA-LSM using ES(6,39). The designs are ordered according to their structural performance, which varies by up to 8.8%, depending on the local optimum found by the algorithm. From the mechanical point of view, all the structures make physical sense, exhibiting some similarities to the 2D designs shown at the beginning of this chapter. As in case of the 2D examples, high diversity of designs is observed, being one of the properties of EA-LSM, associated with the use of EAs. This effect might be even stronger for 3D optimization cases due to the high dimensionality of the optimization problem and may strongly depend on the optimization algorithm and the population sizes.

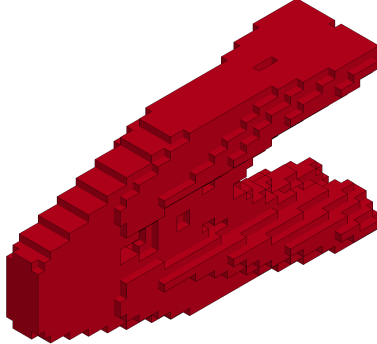
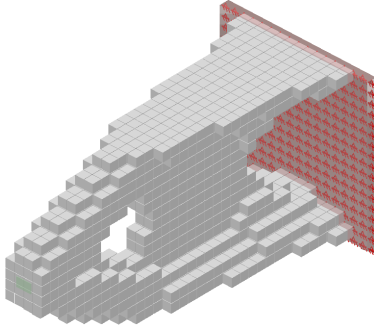
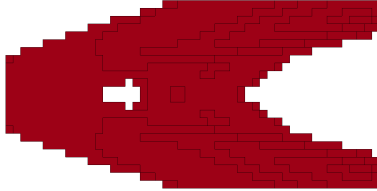
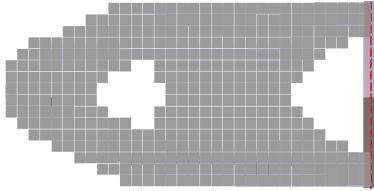
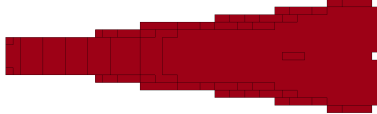
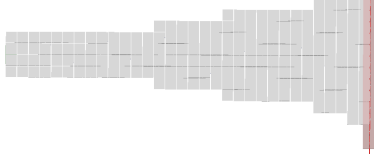
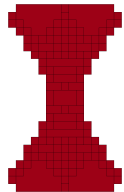
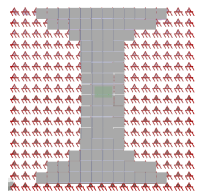
In order to validate the obtained results, the optimization problem was solved using the state-of-the-art gradient-based SIMP approach, as well. The resulting topology is compared with the best design obtained using EA-LSM in Table 7.17. Based on visual inspection from multiple views, one can conclude that both solutions represent qualitatively similar design concepts. The

Table 7.16 Final designs in five independent EA-LSM runs with ES(6,39). The compliance values are normalized with the compliance of the best-performing structure.

Normalized compliance	FE mesh	0 th LSF isocontour
1.0		
1.027		
1.063		
1.068		
1.088		

minor discrepancies between the topologies might predominantly come from differences in the representations used by both algorithms (consider the front view in Table 7.17). First of all, EA-

Table 7.17 Comparison of the best design among five independent EA-LSM runs with a structure obtained with use of the state-of-the-art SIMP approach (DTU (2018)). Filter size 2.0.

View	EA-LSM	SIMP
Isometric		
Side		
Top		
Front		

LSM, unlike SIMP, does not allow for intermediate densities, which changes the optimization problem as well as the shape of the optimal designs. Secondly, the representation used by EA-LSM relies on only 64 MMCs, with 2 symmetry planes, resulting in 144 design variables. In contrast, the optimization problem solved by SIMP in this case involves 8192 design variables, which implies much higher structural attainability of this representation, allowing for fine-tuning the final shape of the design.

In order to analyze the optimization process and estimate the computational costs associated with the 3D EA-LSM, Figure 7.33 presents the evolution of the cost function during the optimization process. Please note that the reference design (Figure 7.32), used for the optimizations in this section, violates the 20% volume fraction constraint, which results in very high initial values of the penalized cost functions. This is done on purpose, to demonstrate the ability of the algorithm to start with an infeasible initial design, which would be usually the case for highly-constrained

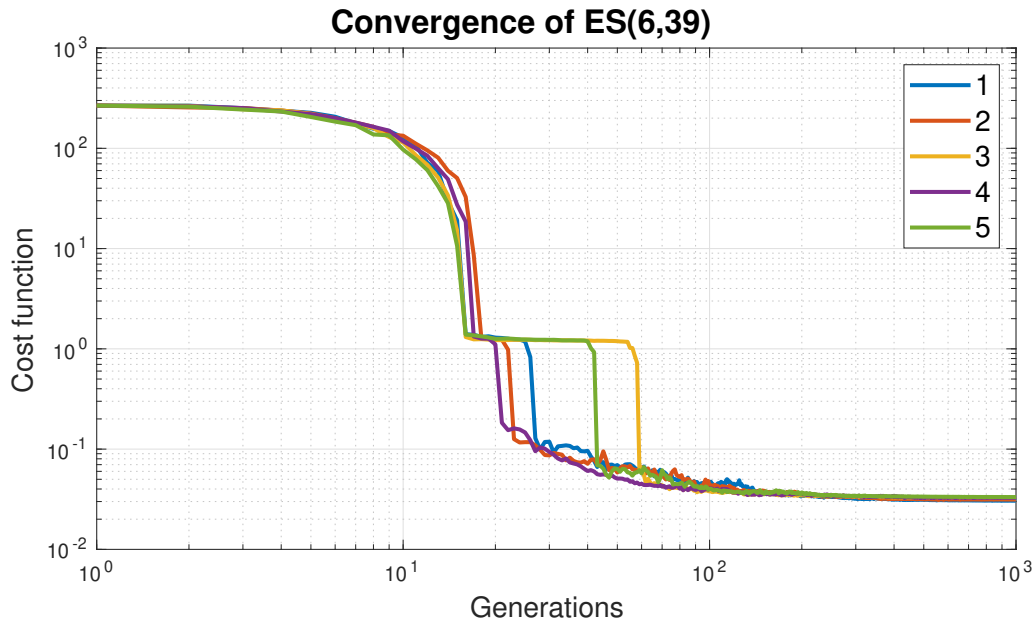
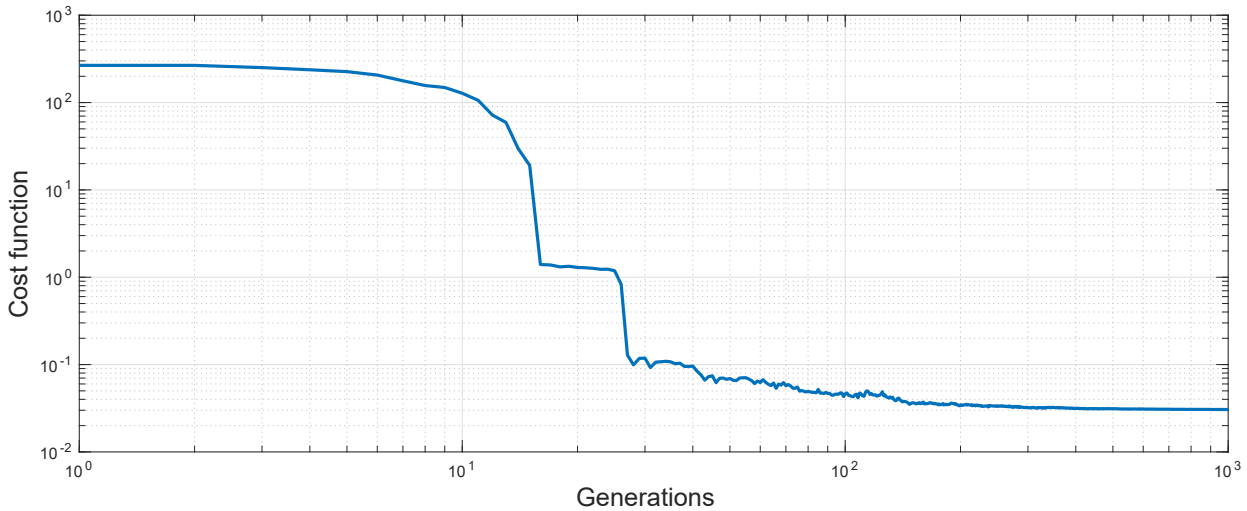


Figure 7.33 Convergence of the cost function for five optimization runs of ES(6,39) for 144-dimensional 3D cantilever beam optimization problem. The numbers in the legend correspond to the ranks of the final solutions according to their compliances, which are equivalent to the row numbers in Table 7.16.

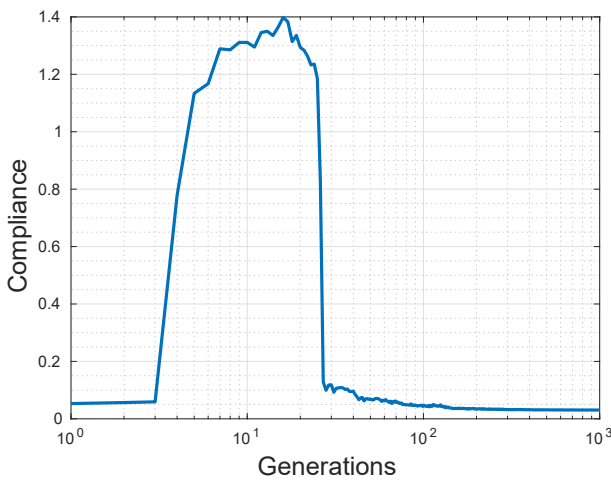
industrial optimization problems. It is clear from the plots that all of the runs are able to find feasible designs, converging after 100–200 generations, corresponding to 3900–7800 FE evaluations. If sufficient computational resources are available, the number of generations is of higher importance, demonstrating that EA-LSM can be used for optimization of 3D TO problems and involves comparable computational times to the state-of-the-art gradient-based techniques (ca. 100 iterations for SIMP). This was the case for the experiments carried out in this section, where all 39 offspring individuals were computed in parallel. Otherwise, if the number of FE evaluations is the limiting factor, ML-based enhancements as described in Chapter 6 can be used to further decrease the computational costs. In particular, the H-EA-LSM seems to be a suitable solution, since its performance, unlike for KG-LSM or A-EA-LSM, does not strongly depend on the dimensionality of the optimization problem. The initial experiments with H-EA-LSM for 3D TO problems were very promising (Krischer (2018)).

Finally, Figure 7.34 shows the convergence of the cost function, compliance, and the normalized volume constraint for the best design. One can easily see how, directly after the second iteration, the algorithm selects the individuals of much higher compliance to decrease slightly the volume of the structure. This strong effect can be seen due to relatively high penalization of the constraint violations (Equation (5.28), $P = 1000$) compared to the value of the objective function. After just 20 generations, the volume constraint, violated initially by almost 30%, is satisfied and the optimizer focuses on the reduction of compliance. After ca. 200 generations, only very small variations of the value of compliance can be observed, indicating convergence of the optimization.

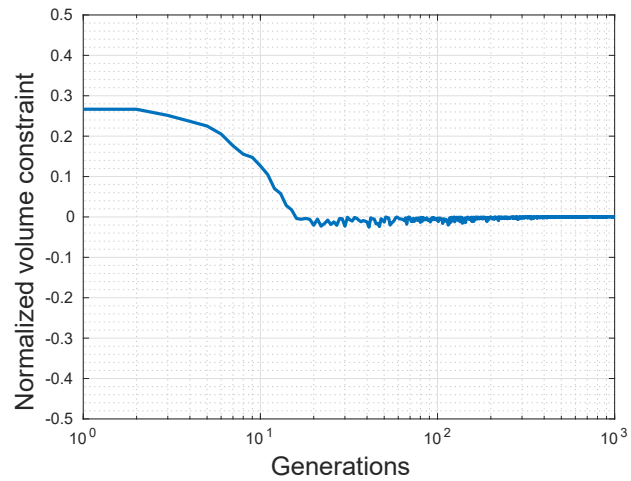
To conclude, this section demonstrates the ability of EA-LSM to address 3D structural TO prob-



(a) Convergence of the cost function.



(b) Convergence of compliance.



(c) Convergence of volume constraint.

Figure 7.34 Convergence of the optimization for the EA-LSM run resulting in the design with the lowest compliance, shown in Table 7.17.

lems. To the best of our knowledge, this is the first time to address problems of such complexity level, close to the industrial cases, with use of evolutionary optimization methods. So far, an extremely high dimensionality of the 3D optimization problems, introduced by the other representations, was the key limiting factor. Thanks to the low-dimensional level-set representation used in this work, 3D evolutionary-based TO became feasible within an affordable budget of FE evaluations. Similarly to the 2D cases considered previously, due to its global search properties, EA-LSM identifies multiple local optima which give rise to alternative design concepts. However, within five optimization runs, a design consistent with the one derived with the state-of-the-art SIMP approach was obtained, demonstrating the correctness of the proposed approach.

7.3. Summary

This chapter presents an experimental evaluation of EA-LSM, as well as of its extensions: H-EA-LSM, KG-LSM, and A-EA-LSM, by analyzing linear elastic static test cases. A minimum compliance optimization problem with a volume constraint is considered in all of the experiments. Since for this problem analytical sensitivity information is available, the results can be directly compared with the reference structures obtained with the state-of-the-art gradient-based approaches. However, one should note that the results obtained with the state-of-the-art methods depend also on the chosen parameter settings. The discussions and comparisons are based on a statistical evaluation of the proposed methods.

All of the proposed approaches yield results consistent with the gradient-based methods. The investigation of the key features of EA-LSM reveals its ability to derive multiple design concepts for an ideation in the early phases of design process, relatively good robustness to different initial designs, and insensitivity of the final designs to different mesh sizes. EA-LSM can derive designs of different structural complexity, depending on the number of basis functions used. For a comparable number of design variables as in other TO approaches relying on EAs, the representation based on MMCs results in solutions qualitatively more similar to the ones obtained with gradient-based methods. The main drawback of EA-LSM compared to gradient-based or heuristic approaches is a relatively high number of required FE evaluations. However, the performance of those approaches and EA-LSM in terms of optimization iterations can be very similar when using parallel computing, thanks to a very good scalability of EAs.

To further reduce the number of necessary FE evaluations, we show that the solutions from the field of ML can be useful for achieving that goal. H-EA-LSM is able to considerably improve the convergence properties of the evolutionary search, both in terms of convergence velocity as well as the quality of the final solutions. We illustrate also how the models of sensitivities, matching exactly the analytical gradients, can be built based on a supervised learning approaches and the data sampled using finite-differencing. For MMC parametrizations leading to low-dimensional optimization problems, we show that the number of costly evaluations can be reduced significantly using KG-LSM, implementing a problem-specific variant of EGO. Finally, we demonstrate how the concept of adaptive representations utilized by A-EA-LSM can be used in TO to reduce the effective dimensionality of the search space, determining the convergence velocity of EAs. Again, via learning of rules for performing topology variations within the A-EA-LSM framework, we show that supervised learning techniques can be used to enhance TO methods.

Finally, the possibility to solve 3D TO problems with EA-LSM is demonstrated. To our knowledge, this is the first successful application of evolutionary optimization to this type of problems. Though the computational costs in this case rise considerably, the parallelization of computations makes the

optimization affordable, and the further mitigation of computational costs, e.g. using H-EA-LSM, is potentially possible.

We conclude that the proposed approaches are able to provide solutions consistent with the state-of-the-art methods. The higher computational costs of the proposed methods is justified by their generic character, implying their potential applicability to a wide range of diverse problems in structural mechanics where the gradient-based approaches cannot be used. To evaluate that, in the following chapter, the proposed methods are used to solve nonlinear dynamic crash TO problems.

Chapter 8

Optimization of nonlinear crash cases

In this chapter, the applicability of EA-LSM and its extensions to crash problems is studied based on academic test cases. We focus on two different types of optimization problems, addressing typical criteria for crashworthiness design, such as intrusion and peak deceleration. Although those criteria are of very high practical relevance, none of the state-of-the-art methods for crash TO can address them directly. In particular, considering deceleration is very interesting, since it could pose severe difficulties for the optimizers, due to very high levels of numerical noise. In all of the numerical experiments, transverse bending problems, widely studied in literature, are considered. Finally, the designs obtained with the proposed methods are discussed and compared with the ones optimized with the state-of-the-art methods – ESL and HCA.

Similarly to Chapter 7, this chapter is divided into two parts. The first part – Section 8.1 presents the results of optimizations with the proposed methods on 2D crash problems. First of all, in Section 8.1.1, EA-LSM is thoroughly evaluated based on several crash problems. Based on the obtained results, general conclusions about the properties of the method and the optimized structures are derived. Secondly, in Section 8.1.2, H-EA-LSM, utilizing physical (ESL) and mathematical (ML) sensitivity models, is used to optimize crash structures for minimal intrusion. Thirdly, the properties of KG-LSM on a crash problem are evaluated in Section 8.1.3. Finally, the concept of adaptive representations and learning-based topology variations is studied based on A-EA-LSM in Section 8.1.4. In the second part of this chapter, Section 8.2, very high-dimensional 3D crash TO problems are addressed with the use of EA-LSM. We show that the proposed method is suitable for solving 3D crash problems involving very high levels of plastic deformations.

8.1. Two-dimensional topology optimization for crashworthiness

Unlike in the case of TO of linear elastic structures, for crash TO, there is no general consensus regarding the benchmark cases used in the community. The problem arises already at the level of defining appropriate objectives and constraints, which is very challenging due to the complexity of the entire vehicle design process. Overall crash performance of a car depends on interactions between several components, developed usually by different departments. Therefore, precise definition of objectives, constraints, and boundary conditions at the early design stages of a structural component is very difficult, resulting in a variety of optimization cases.

One of the more frequently used test cases in crash TO is the transverse bending of a 2D rectangular beam (Aulig (2017), Bujny et al. (2017a)) derived from the work of Patel (2007). It is inspired by the side pole impact from the European New Car Assessment Programme (Euro NCAP), representing situations when a driver loses the control over a vehicle due to e.g. slippery conditions and the car is travelling sideways into rigid objects close to the road such as poles or trees. Such situations lead very frequently to severe injuries or death. A properly designed car structure, in such crash events, should prevent the intrusion of the pole into the passenger compartment, while keeping the impact force on a low level.

The nonlinear dynamic crash test case used in this section is depicted in Figure 8.1 and its configuration is defined in Table 8.1. The FE model used in the optimizations is shown in Figure 8.2. In this scenario, a rigid cylindrical pole impacts the middle of a rectangular aluminum beam, which is fixed at both ends, concerning all three degrees of freedom: two in-plane displacements and one in-plane rotation. In order to guarantee a 2D behavior of the system, the displacements of all the nodes in the direction perpendicular to the plane are constrained to zero. To allow for non-symmetrical displacements in the simulation, e.g. due to buckling, no symmetry conditions are imposed at the level of FE modeling. But, the optimized design is restricted to be symmetric w.r.t. the central axis, unless stated otherwise, to reduce optimization complexity. The distribution of the material inside the beam is subject to TO, with deletion of finite elements in the void region $D \setminus \Omega$ (Equation (5.1)), i.e. having density equal to 0.

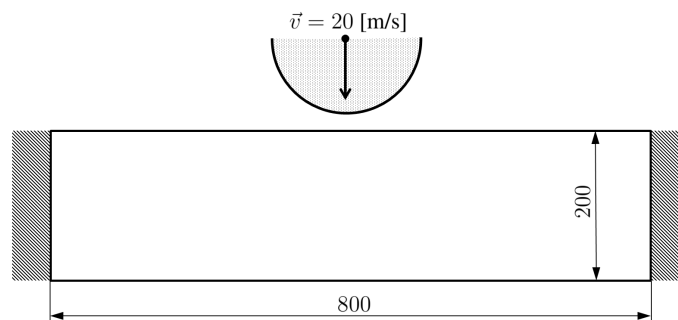


Figure 8.1 Design domain and boundary conditions for the 2D nonlinear dynamic transverse bending crash test case. Dimensioning in mm. The thickness of the design space, i.e. out-of-plane dimension, equals 10 [mm].

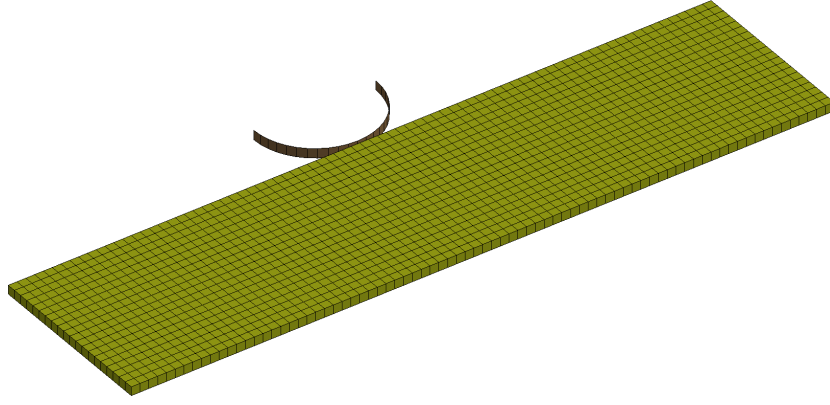


Figure 8.2 FE model used for 2D nonlinear transient crash cases (transverse bending).

Typical formulations of optimization problems for the transverse bending case involve maximization of specific energy absorption, minimization of the intrusion of the impactor, or reduction of the peak force or acceleration (Patel (2007); Bandi et al. (2013); Aulig (2017); Bujny et al. (2017a)). In this section, two different optimization problems will be considered. The first case regarded here as an example, the intrusion minimization problem with a mass constraint, can be formally defined as follows:

$$\begin{aligned}
 & \min_{\mathbf{z}} (d_{max}(\mathbf{z})), \mathbf{z} \in \mathbb{R}^n; \\
 & \text{s.t. } \mathbf{res}(t) = \mathbf{0}; \\
 & \quad m(\mathbf{z}) \leq m_{req},
 \end{aligned} \tag{8.1}$$

where d_{max} is the maximal intrusion of the pole into the structure (before the elastic rebound), \mathbf{z} is the vector of design variables, $m(\mathbf{z})$ is the mass of the whole structure, and m_{req} is the required mass of the optimized design. The condition $\mathbf{res}(t) = \mathbf{0}$ corresponds to the dynamic equilibrium at time t . Such formulation of an optimization problem is directly related to the principal goal of structural design for the side pole impact, and can be used to find optimal topologies for the side sill or the door reinforcement beams. With some modifications of the load case, it can also be transferred to design tasks of A-, B-, or C-pillars such that an overall protection of the passengers can be derived by limiting the deformation of the safety cell.

The second important criterion in the design of crashworthy structures for the side pole impact is the peak acceleration during the crash event. Although a structural criterion is regarded here, it can be roughly related to the highly-important Head Injury Criterion (HIC). Therefore, as the second optimization case, a mass minimization problem with constraints on the maximal structural

Table 8.1 Configuration of the 2D crash test case. For simplicity, the strain rate dependency is not considered here.

Property	Symbol	Value	Unit
Material model of the structure	-	Piecewise linear plasticity	-
LS-Dyna material keyword (structure)	-	*MAT_024 (LSTC (2014))	-
Mass density	ρ_m	$2.7 \cdot 10^3$	kg/m ³
Young's modulus	E	$7.0 \cdot 10^4$	MPa
Poisson's ratio	ν	0.33	-
Yield strength	R_e	241.0	MPa
Tangent modulus	E_{tan}	70.0	MPa
Static coefficient of friction	μ_s	0.2	-
Dynamic coefficient of friction	μ_k	0.2	-
Material model of the pole	-	Rigid	-
Reference pole velocity	v	20	m/s
Reference pole mass	m_p	11.815	kg
Pole diameter	D	139.154	mm
Required structure mass	m_{req}	2.16	kg
Mesh resolution	-	80 x 20 x 1	-
Solver	-	LS-DYNA R7.1.1	-
Element formulation	ELFORM	Constant stress solid element	-
Hourglass control	IHQ	Flanagan-Belytschko	-
Contact model	-	Penalty method	-

acceleration and intrusion is selected. The optimization problem can be defined formally as:

$$\begin{aligned}
 & \min_{\mathbf{z}} (m(\mathbf{z})), \mathbf{z} \in \mathbb{R}^n; \\
 & \text{s.t. } \mathbf{res}(t) = \mathbf{0}; \\
 & \quad a_{max}(\mathbf{z}) \leq a_{req}; \\
 & \quad d_{max}(\mathbf{z}) \leq d_{req},
 \end{aligned} \tag{8.2}$$

where a_{max} denotes the maximal absolute acceleration of the impactor during the crash event and a_{req} its the maximal allowed magnitude. Similarly, d_{req} denotes the maximal allowed intrusion of the pole into the structure. Such formulation makes the optimization problem much more challenging due to the high noisiness of the acceleration constraint. Moreover, some state-of-the-art methods, like HCA, have difficulties with handling more than one constraint (Zeng (2018)), since the constraints are often contradictory and cannot be easily satisfied using simple rules assuming some relationships between their values and the mass of the structure. This is not the case for the EA-LSM and its extensions, so the second problem serves as a relevant case for demonstrating the ability to include multiple constraints within the proposed methods.

The use of the transverse bending crash case in this work is motivated mainly by its simplicity and availability of some reference optimization results. Since the overall goal of this section is to validate the proposed methods on highly nonlinear crash cases, taking into account a variety of different design criteria, it serves as a suitable scenario. Finally, the simplicity of this test case helps to understand the mechanical behavior of the obtained structures.

The results obtained in this section are compared with two state-of-the-art methods for crash TO – HCA (Patel (2007)) and the simplest variant of the ESL method (Cavazzuti et al. (2010)), also referred to as the global ESL (Duddeck et al. (2016)) or replacement loading (Wehrle et al. (2015)).

As discussed in Chapter 2, unlike EA-LSM, HCA does not address a given optimization problem directly, but homogenizes the energy absorption all over the structure instead. This heuristic criterion, however, turns out to yield very good topologies for a number of problems including energy absorption maximization (Patel (2007)) and intrusion minimization (Bujny et al. (2017a); Bandi et al. (2013)). This means that the optimization problems regarded here are not identical and that the results are only in a limited manner comparable. This is justified, because there is, to the knowledge of the author, no other method currently available enabling exact validations. Therefore, HCA is proposed here as a suitable method for validating the EA-LSM approach on the test cases described above. In this work, we use the commercial implementation of HCA from LS-TaSC 3.2¹, which, to great extent follows the methodology proposed by Patel (2007). In all of the cases, the default settings of the program are used.

In the simplified variant of the ESL method used in this work, the dynamic crash load is replaced with a static force² at the location of the contact between the impactor and the structure. Figure 8.3 depicts schematically the ESL counterpart of the crash case from Figure 8.1. In the next step, the material model is replaced with the linear elastic one, with Young's modulus of $E = 7 \cdot 10^4$ [MPa] and Poisson ratio $\nu = 0.33$. After that, a gradient-based SIMP approach as described in the

¹ <http://www.lstc.com/products/lstc-tasc>, retrieved January 17, 2020.

² Due to the linearity of the system considered here for the ESL optimization, the actual magnitude of the force does not matter. For any force level, the compliance minimization with a volume constraint leads to exactly the same final topology.

work of Sigmund (2014), with a filter size of 15 [mm] and the material penalization power $p = 3$, is used to optimize the topology w.r.t. compliance.

Both of the state-of-the-art methods described above operate on densities³ of the finite elements as the design variables. Since the densities can take any values in the range between 0 and 1, the optimized topologies often contain elements with ambiguous, intermediate densities⁴. In order to compare the topologies obtained with those methods and EA-LSM, the results have to be post-processed to guarantee a 0-1 material distribution. To achieve that, we delete all the elements with densities below a given threshold ρ_{th} and assign the full density to the rest of the elements. The threshold ρ_{th} is chosen in such a way that the resulting design satisfies the specified volume or mass constraint.

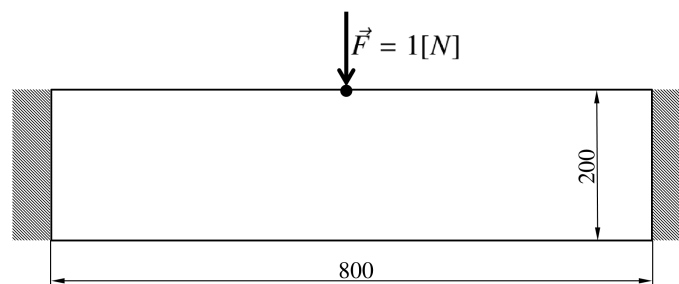


Figure 8.3 Equivalent static load for the crash case depicted in Figure 8.1.

8.1.1. Evolutionary level set method (EA-LSM)

In this section, properties of the standard EA-LSM, as described in Chapter 5, are investigated. The general idea is to evaluate the performance of the method on different types of crash problems, including changing impact energy, impact velocity as well as the intrusion and acceleration constraints. In addition, the impact of the mesh resolution on the final topologies is assessed. The results described in this section are the base for evaluations of the extensions of EA-LSM, described in Chapter 6. In all of the optimizations presented in this section, a representation with 16 MMCs is used. This results in a 40-dimensional optimization problem after imposing symmetry. Consequently, a CMA-ES method with $\mu = 7$ parent and $\lambda = 15$ offspring individuals is used. The population sizes were determined according to the rules specified in Appendix B.2. The initial population is generated based on the reference design shown in Figure 5.9(a). The initial step size for CMA-ES and the standard deviation used to distribute the initial population around the reference design is set to $\sigma_{init} = 0.025$.

³ The design variables used to scale the material properties such as mass density, Young's modulus, yield strength, or hardening modulus, are frequently referred to as element densities (Bendsøe and Sigmund (2004); Patel (2007)). Please note that this is not equivalent to mass density of the material, although usually the density is used to scale the mass density, as well.

⁴ To eliminate intermediate densities, penalization schemes such as SIMP (Bendsøe and Sigmund (2004)) are frequently used in density-based TO. However, in many cases, they are still not sufficient to guarantee clear separation of the material and void.

Validation of the method on intrusion minimization problem

At first, the intrusion minimization problem with 50% mass fraction constraint for the crash transverse bending scenario is considered. In order to examine the sensitivity of EA-LSM to different impact velocities (impact energies), corresponding to different levels of nonlinearity of the cost function, optimizations are performed for initial impact velocities of 10, 20, and 40 [m/s]. The mass of the impactor is kept constant. For each of the 3 impact velocities, 30 optimization runs are carried out to allow for statistical evaluation of the results, accounting for the inherent randomness of the algorithm.

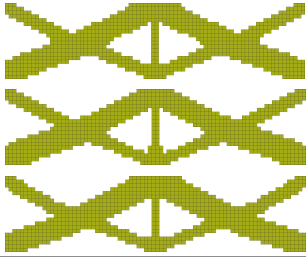


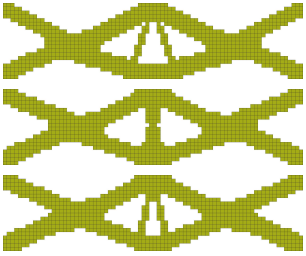



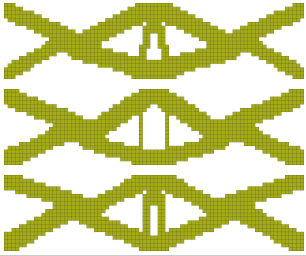

Selected final designs out of the 90 optimization runs are depicted in Table 8.2. For each impact velocity, the best designs are divided into 3 groups corresponding to different topology types, most probably related to local optima of the optimization problem. The performance of the best topologies under impact conditions is presented in Table 8.4.

The types of designs obtained with EA-LSM vary strongly with the changing impact velocity. This questions the use of the ESL approach in general case and justifies the increased computational cost of EA-LSM. For the impact velocity of 10 [m/s], the best design type corresponds to a 'supporting' structure, with members in compression playing the most important role. For the 20 [m/s] impact case, the thicknesses of the members in tension and compression are similar, representing therefore a more 'balanced' design. Finally, for 40 [m/s], the beams in tension play the most important role, leading to a 'hanging' structure. Those results are consistent with the ones presented by Buhl et al. (2000), where the optimal designs for linear and nonlinear modeling, similar to some extent to the crash cases considered here, were compared. To check the validity of the obtained results, the performance of the best designs from EA-LSM was evaluated for all of the considered impact velocities (Table 8.3). The comparison leads to a conclusion that the differences between the structures optimized for different impact velocities are not related to the fact that EA-LSM finds different local optima, but the designs optimized for a given velocity outperform significantly the ones optimized for different impact velocities. Interestingly, the best design for the highest impact velocity is the most robust one w.r.t. the variations of the impactor's speed. What is more, Table 8.2 shows that the types of the optimized designs remain similar for different impact velocities, but their relative performance changes with the velocity.

The results presented above demonstrate the ability of EA-LSM to optimize topologies for crashworthiness criteria, for different levels of nonlinearity associated with varying impact velocities and reflected in different amounts of effective plastic strain. As for the linear elastic test cases, the obtained designs exhibit certain structural diversity, corresponding to different local optima found by the evolutionary optimizer.

In order to validate the obtained results, we compare the best EA-LSM designs with the ones obtained with the state-of-the-art methods for crash TO – global ESL and HCA. The performance of

Table 8.2 Comparison of the best designs obtained with EA-LSM for the intrusion minimization problem with different impact velocities and a constant mass of the impactor (Bujny et al. (2017a)). Intrusions in mm. Designs within each group represent the same type of topology, identified based on visual inspection.

EA-LSM designs in groups representing different (local) optima			
Velocity	Best optimum	Second best optimum	Third best optimum
10 [m/s]	Lowest intrusion: 5.18 	Lowest intrusion: 5.22 	Lowest intrusion: 5.24 
20 [m/s]	Lowest intrusion: 15.01 	Lowest intrusion: 15.22 	Lowest intrusion: 15.37 
40 [m/s]	Lowest intrusion: 51.30 	Lowest intrusion: 51.79 	Lowest intrusion: 53.76 

the structures obtained with the three considered methods is evaluated based on nonlinear transient FE simulations. The structures optimized with different methods are depicted in Table 8.5. In all of the cases, the performances of the obtained structures are very similar. EA-LSM outperforms in terms of maximal intrusions the global ESL for all impact velocities. As expected, once the plasticity effects become stronger for the higher impact velocities, the intrusion difference between EA-LSM and ESL rises up to 3.17 [mm] (over 6% increase w.r.t. EA-LSM) for the impact velocity of 40 [m/s]. This is understandable, since the global ESL is meant to be used mainly for the crash cases with very limited amount of plastic deformation (Duddeck et al. (2016)). For the velocities of 10 and 20 [m/s], EA-LSM slightly outperforms HCA, while for 40 [m/s] the intrusion of the impactor is ca. 0.2% lower for the HCA result. This is surprising since the HCA method

Table 8.3 Comparison of the best designs obtained with EA-LSM for constant impactor mass (Bujny et al. (2017a)). Percentage values express growth of intrusion w.r.t. the design optimized for a given impact velocity.

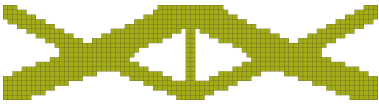
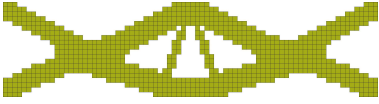

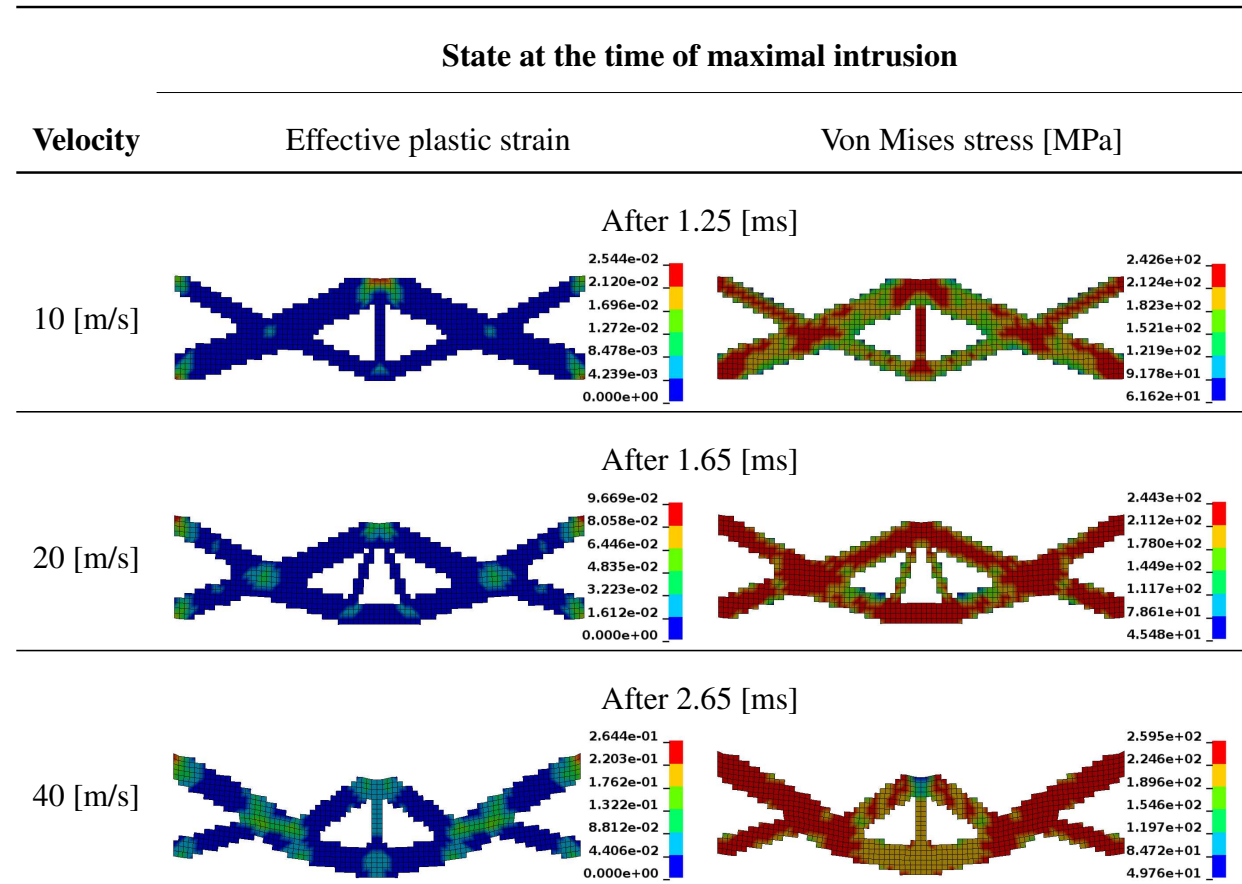




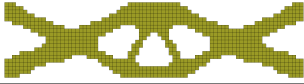




EA-LSM design	Optimized for	Intrusion at different velocities		
		10 [m/s]	20 [m/s]	40 [m/s]
	10 [m/s]	5.18 [mm]	15.09 [mm] +0.51%	53.48 [mm] +4.25%
	20 [m/s]	5.18 [mm] +0.06%	15.01 [mm]	52.64 [mm] +2.62%
	40 [m/s]	5.27 [mm] +1.70%	15.21 [mm] +1.32%	51.30 [mm]

Table 8.4 Deformation of the best topologies obtained with EA-LSM under impact conditions with different velocities (Bujny et al. (2017a)). Effective plastic strain and von Mises stress fields are shown on the deformed structures. The scales in each of the plots are chosen independently to show clearly the variations in the distributions of the depicted fields.



does not address the minimization of intrusion directly, but achieves it through a heuristic energy homogenization criterion. In contrast, EA-LSM in this case tackles the intrusion minimization

Table 8.5 Comparison of the best topologies obtained with EA-LSM, HCA, and global ESL for the intrusion minimization problem.

Best designs obtained with different methods and their performance			
Velocity	EA-LSM	HCA	ESL
10 [m/s]	Intrusion: 5.18 [mm] 	Intrusion: 5.37 [mm] 	Intrusion: 5.21 [mm] 
20 [m/s]	Intrusion: 15.01 [mm] 	Intrusion: 15.54 [mm] 	Intrusion: 15.58 [mm] 
40 [m/s]	Intrusion: 51.30 [mm] 	Intrusion: 51.19 [mm] 	Intrusion: 54.40 [mm] 

problem in a direct way, which results in higher computational costs and should lead to superior performance of the obtained designs. Hypothetically, the reason for such situation lays in the limited structural attainability of EA-LSM. More precisely, while HCA in the considered case operates on 1600 design variables, representing the densities of the finite elements, EA-LSM uses only 16 MMCs, corresponding to 40 design variables when a symmetry criterion is imposed. It is visible from Table 8.5 that with such a coarse representation it is not possible to represent the high complexity of the designs obtained with the HCA approach. One can note that with rising impact velocity, the structures become more and more difficult to be represented with beam-like components, explaining why HCA outperforms EA-LSM for the 40 [m/s] velocity. On the other hand, HCA seems to be very good for the intrusion minimization problem (Bandi et al. (2013)). It turns out that with rising impact velocity, the homogeneity of the effective plastic strain, correlated with the internal energy, rises significantly for the best designs obtained with the EA-LSM (Table 8.4). This partially explains why the design obtained with HCA method for 40 [m/s] impact velocity performs so well. All in all, the following conclusions can be derived from the presented results:

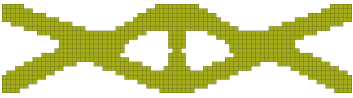


- HCA method performs very well for the intrusion minimization problem. Since it is associated with very low computational costs, it is a preferred method for this type of problems.
- The performance of the structures obtained with EA-LSM would improve once the number of MMCs is increased. In Section 7.1.1, we demonstrated on the linear elastic cantilever beam

case that the performance of the structure can be improved by increasing the number of MMCs. Therefore, EA-LSM should be still able to outperform HCA for all the cases, for the price of higher computational costs.

Influence of impact velocity for constant impact energy

Since EA-LSM optimizes structures based directly on the nonlinear dynamic simulations, it takes into account the inertial effects occurring in the crash events. In order to demonstrate the importance of this property, two additional impact scenarios are introduced. In these scenarios, the initial velocity of the impactor is varied, but the impact energy is kept equal to the one from the reference case, shown in Figure 8.1. We compare the topologies obtained with EA-LSM and the global ESL, which neglects the inertial effects by introducing the replacement loads.

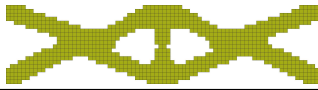

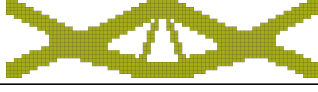



Table 8.6 Comparison of the best designs obtained using EA-LSM for constant impact energy (Bujny et al. (2017a)). Percentage values express growth of intrusion w.r.t. the design optimized for a given impact velocity.

EA-LSM design	Optimized for	Intrusion at different velocities		
		10 [m/s]	20 [m/s]	40 [m/s]
	10 [m/s]	15.15 [mm]	15.04 [mm] +0.17%	14.69 [mm] +2.16%
	20 [m/s]	15.15 [mm] +0.01%	15.01 [mm]	14.57 [mm] +1.38%
	40 [m/s]	15.54 [mm] +2.57%	15.38 [mm] +2.48%	14.38 [mm]

The best designs (always out of 30) obtained with EA-LSM for different impact velocities are presented in Table 8.6. The results show that the inertial effects have significant influence on the type of topology that is found in the optimization. From the mechanical point of view, this time the supporting structure turns out to be the best for the highest impact velocity. This is logical, since this scenario corresponds to the lowest momentum of the impactor and involves comparably small plastic deformations. This makes the problem similar to the 10 [m/s] case from the previous section. What is more, this design is the least robust w.r.t. varying impact velocities (intrusion growth w.r.t. reference case always above 2%). In contrast, the optimization case with 20 [m/s] yields a balanced topology, absorbing impact energy in a much more uniform way (Bujny et al. (2017a)).

The comparison of the obtained results with the global ESL, presented in Table 8.7, confirms the importance of the inertial effects; these results show that linear and static approaches for TO are not capable to derive appropriate designs in cases where dynamic effects or stronger nonlinearities become relevant. For all impact velocities, the structures optimized with EA-LSM perform significantly better than the ones optimized with ESL.

Table 8.7 Comparison of the best topologies obtained with EA-LSM and global ESL for the intrusion minimization problem with different impact velocities and constant impact energy.

Best designs obtained with different methods and their performance		
Velocity	EA-LSM	ESL
10 [m/s]	Intrusion: 15.15 [mm] 	Intrusion: 15.71 [mm] 
20 [m/s]	Intrusion: 15.01 [mm] 	Intrusion: 15.58 [mm] 
40 [m/s]	Intrusion: 14.38 [mm] 	Intrusion: 15.00 [mm] 



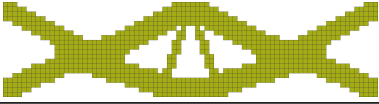

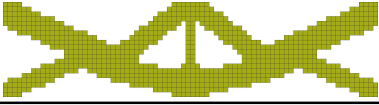

Influence of the mesh resolution

Similarly to the considerations regarding application of EA-LSM to optimization of linear elastic structures, an analysis of the influence of the FE mesh resolution on the final topology is realized. We perform this additional test due to the nonlinear character of the simulations considered in this section, which might lead to a different outcome than in the linear elastic case.

The evaluation of the impact of the mesh resolution on the type of the obtained topology is based on 15 independent optimization trials with mesh resolution doubled in each direction. This results in an eight-fold increase of the number of elements in the FE model, from 1,600 to 12,800. The best final topologies obtained with different FE mesh resolutions are compared in Table 8.8.

Similarly to the linear elastic case, the mesh resolution seems not to have a significant influence on the type of topology developed by EA-LSM. This justifies the use of coarser meshes in the analyses presented in this work in favor of computational efficiency.

Table 8.8 Study of the influence of the mesh resolution on the optimized crash structures (Bujny et al. (2017a)). The best designs obtained in 30 and 15 optimization runs with coarse and fine FE mesh, respectively. Test case with constant impactor mass.

Best EA-LSM designs for different mesh sizes		
Impact velocity	Mesh size: 80 x 20 x 1	Mesh size: 160 x 40 x 2
10 [m/s]		
20 [m/s]		
40 [m/s]		

Mass minimization with intrusion and acceleration constraint

As shown in the previous sections, although EA-LSM can be used to solve the intrusion minimization problem, considerably cheaper state-of-the-art methods yield structures of comparable structural performance. However, real-world crashworthiness optimization problems are usually highly constrained and frequently involve injury-related optimization criteria. This already limits significantly the use of the state-of-the-art methods in the engineering practice.

Since EA-LSM does not make the heuristic assumptions about the properties of the optimization problems the state-of-the-art methods do, it can optimize virtually any quantifiable objective functions. Moreover, any number of constraints can be easily handled with the penalty methods, being a clear advantage over the HCA approach. In this section, a more challenging scenario, aiming to evaluate EA-LSM on optimization problems with more than one constraint, is considered.

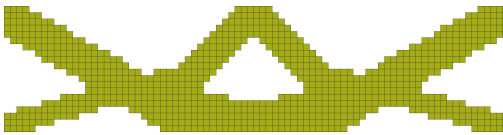
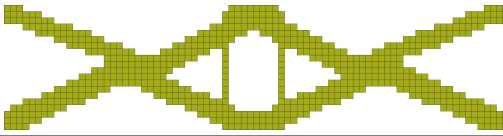
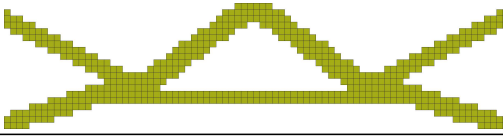
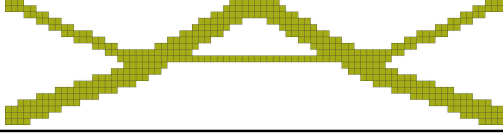
In case of optimization of such structures as pillars, rockers, or side sills, it is crucial not only to reduce the intrusion of the impactor into the structure, but also to keep the maximal acceleration on a possibly low level. This is dictated mainly by biomechanical injury criteria, e.g. HIC. On the other hand, the main objective of the lightweight structural design is the reduction of weight. Therefore, in this section, we consider a problem of mass minimization with constraints on the intrusion and maximal acceleration of the impactor. As in the previous sections, the transverse bending scenario is used. The initial velocity of the impactor in all of the considered cases is set to 20 [m/s]. For each of the four cases with the intrusion constraints of 15 [mm], 20 [mm], 25 [mm] and 30 [mm], 30 optimization runs are performed. In each case, a constraint on the maximal acceleration calculated according to the following formula, is imposed:

$$a_{req} = 1.1 \cdot \frac{v_{init}^2}{2d_{req}}, \quad (8.3)$$

where v_{init} is the initial velocity and d_{req} is the maximal allowed intrusion of the impactor. This condition is motivated by the fact that in the ideal case, in order to minimize the maximal acceleration for a given crash duration, the acceleration should be constant during the whole crash event, until the impactor is stopped. Since it might be very challenging for the optimizer to fulfil this condition, or even not feasible in general case, always an acceleration constraint on a level of 10% higher than the constant one is chosen.

Table 8.9 shows the best topologies obtained with EA-LSM for each of the four cases. The performance of the optimized designs in terms of intrusions and accelerations is presented in Figure 8.4. The deformed structures and effective plastic strain distributions are shown in Figure 8.5. As can be seen in Figure 8.4, the optimized structures exhibit the required behavior, meeting both of the constraints. EA-LSM was able to come up with topologies showing favorable crash behavior for the side pole impact, where the intrusion is limited, while the deceleration is kept almost constant during the crash event.

Table 8.9 Comparison of the best designs obtained using EA-LSM for the mass minimization problem with the constraints on intrusion and maximal absolute acceleration (Bujny et al. (2017a)).

EA-LSM design	Characteristics of the design		
	Intrusion	Maximal deceleration	Mass
	15 [mm]	$1.4239 \cdot 10^7$ [mm/s ²]	2.21 [kg]
	20 [mm]	$1.0566 \cdot 10^7$ [mm/s ²]	1.65 [kg]
	25 [mm]	$8.6117 \cdot 10^6$ [mm/s ²]	1.31 [kg]
	30 [mm]	$7.3057 \cdot 10^6$ [mm/s ²]	1.11 [kg]

Obviously, with the relaxation of the intrusion constraint, the mass of the optimized structures decreases, being approximately inversely proportional to the allowed intrusion. The design concepts change gradually from 'hanging' to 'supporting' structures as more intrusion is allowed, as well. As shown in Figure 8.5, for high intrusions, EA-LSM develops structures with a two-phase crash behavior, allowing for a further reduction of the maximal deceleration of the impactor. For instance, in case of the structure optimized for 30 [mm] intrusion constraint, the first phase of crash (until ca. 1.5 [ms]) involves similar deformations as in case of the designs optimized for minimal

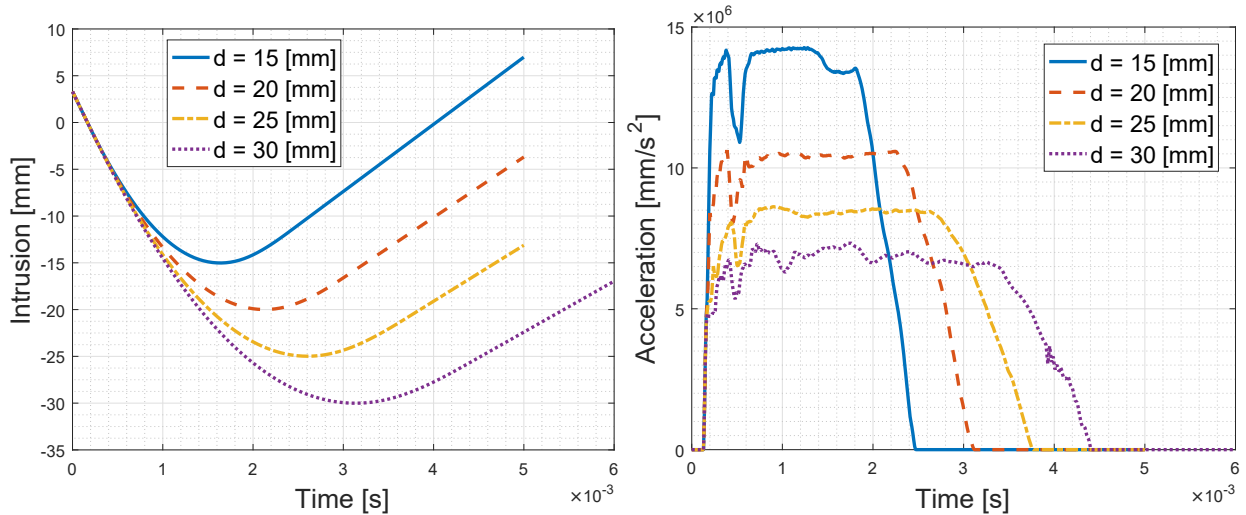


Figure 8.4 Intrusions and absolute accelerations for the best designs obtained with EA-LSM for the mass minimization problem with constraints on intrusion and maximal absolute acceleration (Bujny et al. (2017a)). No numerical filters used.

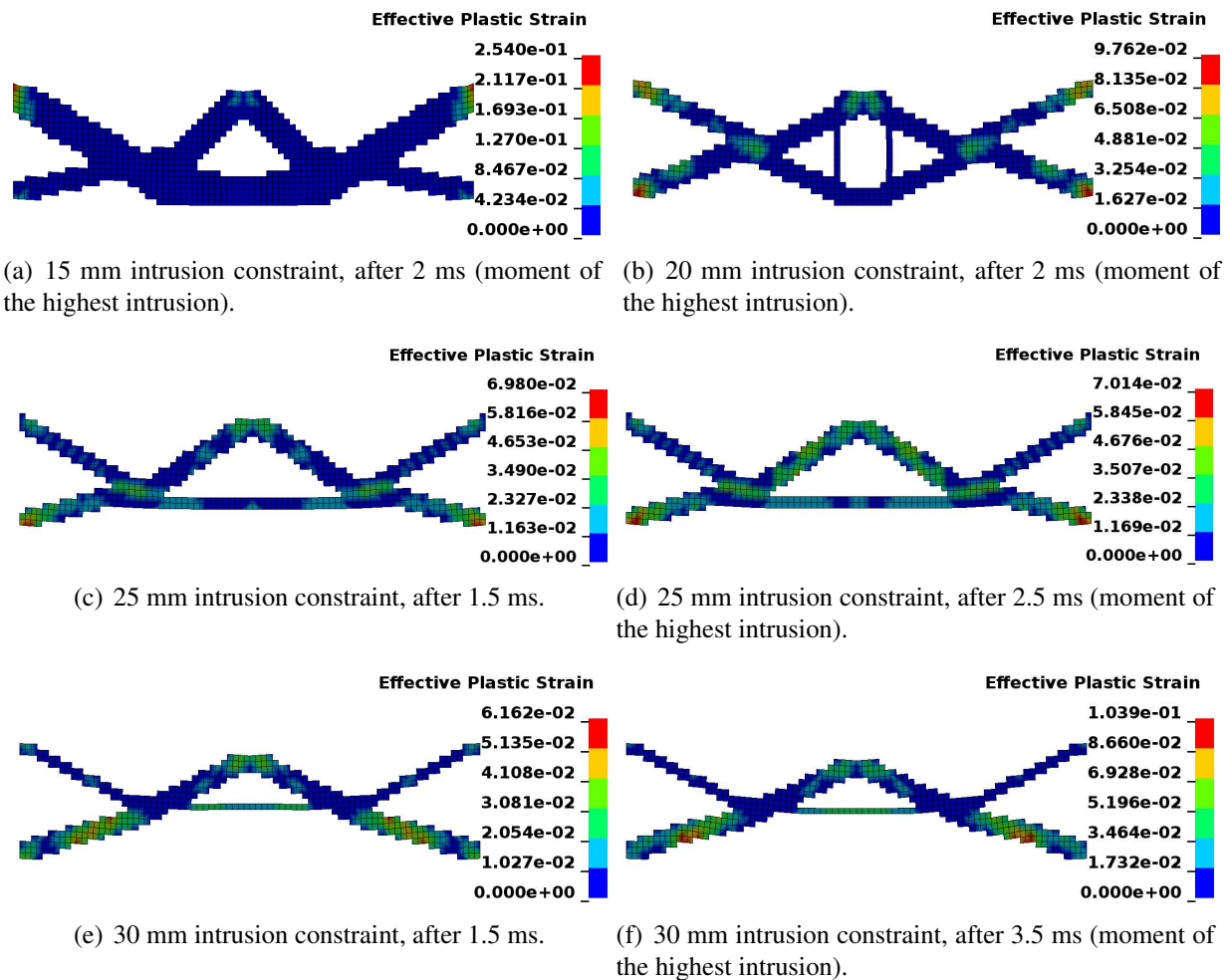
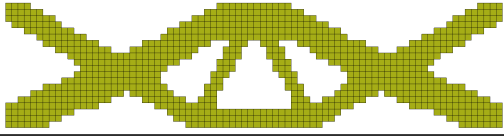
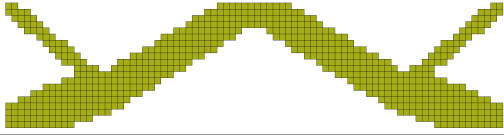
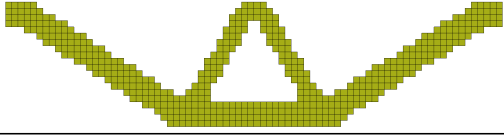
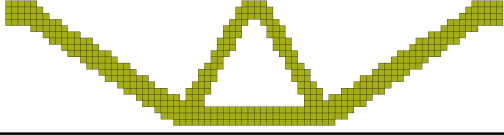


Figure 8.5 Deformed structures and effective plastic strain distributions for the best designs obtained with EA-LSM for the mass minimization problem with constraints on intrusion and maximal absolute acceleration (Bujny et al. (2017a)). The scales in each of the plots are chosen independently to show clearly the variations in the distributions of the plastic strain fields.

Table 8.10 Evaluation of the design obtained with the ESL method. The designs are optimized for minimal compliance with a volume constraint to meet the masses of the designs optimized with EA-LSM (Table 8.9). The percentage values express the growth of intrusion or peak acceleration w.r.t. the respective designs optimized with EA-LSM.

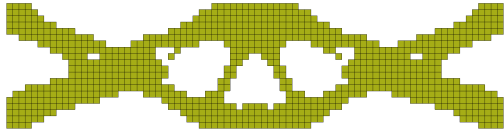
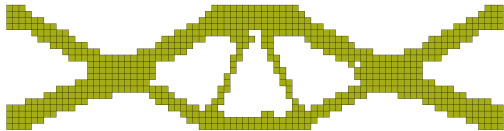
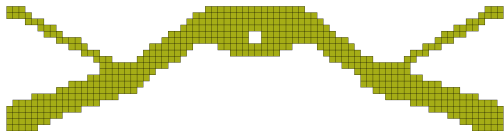
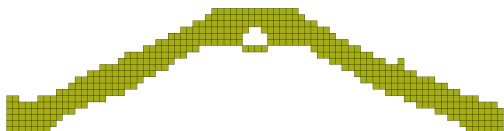
ESL design	Characteristics of the design		
	Intrusion	Maximal deceleration	Mass
	15.42 [mm] +2.8%	$1.468 \cdot 10^7$ [mm/s ²] +3.1%	2.21 [kg]
	21.06 [mm] +5.3%	$1.0826 \cdot 10^7$ [mm/s ²] +2.5%	1.65 [kg]
	25.14 [mm] +0.6%	$1.001 \cdot 10^7$ [mm/s ²] +16.2%	1.31 [kg]
	30.37 [mm] +1.2%	$8.286 \cdot 10^6$ [mm/s ²] +13.4%	1.11 [kg]

intrusion (Table 8.4). In the second phase, however, the members in the bottom corners of design space start to buckle, as well. The tendency towards the buckling behavior and lower masses of the optimized structures would probably hold when further relaxing the intrusion constraint. For those cases, using coarse FE meshes of solid elements would be no longer appropriate and alternative evolutionary-based approaches, e.g. (Bujny et al. (2016a)), should be considered.

For comparison, the results obtained using the state-of-the-art ESL and HCA methods are presented in Tables 8.10 and 8.11, respectively. In case of the ESL approach, since the considered optimization problem formulation cannot be used directly, the topologies were optimized for minimal compliance with a volume constraint leading to the same masses as of the respective structures from Table 8.9. Similarly, HCA method was used to meet the masses of the corresponding structures obtained with the EA-LSM approach.

The results presented above show that both ESL and HCA are unable to generate structures meeting both of the constraints. Moreover, in most cases, the constraints are strongly violated and this problem becomes more severe with relaxing the intrusion constraint. This is understandable, since neither of these methods optimize the underlying problem directly, as EA-LSM does. The heuristic assumptions used by the methods seem not to be valid for the considered case.

Table 8.11 Evaluation of the design obtained with the HCA method. The designs are optimized using the heuristic energy homogeneity criterion with a mass constraint to meet the masses of the designs optimized with EA-LSM (Table 8.9). The percentage values express the growth of intrusion or peak acceleration w.r.t. the respective designs optimized with EA-LSM.

HCA design	Characteristics of the design		
	Intrusion	Maximal deceleration	Mass
	19.05 [mm] +27.0%	$1.4015 \cdot 10^7$ [mm/s ²] -1.6%	2.21 [kg]
	22.46 [mm] +12.3%	$1.0145 \cdot 10^7$ [mm/s ²] -4.0%	1.65 [kg]
	26.13 [mm] +4.5%	$1.1462 \cdot 10^7$ [mm/s ²] +33.1%	1.32 [kg]
	33.07 [mm] +10.2%	$8.0582 \cdot 10^6$ [mm/s ²] +10.3%	1.11 [kg]

8.1.2. Hybrid evolutionary level set method (H-EA-LSM)

In this section, the H-EA-LSM, realizing the concept of gradient-enhanced evolutionary TO, is evaluated based on a nonlinear dynamic crash scenario. In particular, we aim for answering the question if an approximate gradient information obtained from a physical surrogate (equivalent static case) or a pre-trained ML model can improve convergence properties of the evolutionary search in a similar way as when an exact gradient information is available (Section 7.1.2).

In all of the cases discussed here, the intrusion minimization problem with a 50% volume/mass constraint for the impact velocity of 20 [m/s] (Figure 8.1) is used. Similarly to Section 8.1.1, the initial population is generated based on the reference design shown in Figure 5.9(a), using a standard deviation of $\sigma_{init} = 0.025$. For simplicity, we use a fixed step size $s = 0.05$ (design variables normalized as described in Section 5.3.5) in the gradient-improvement step. In order to minimize the influence of any additional mechanisms of EA-LSM (e.g. constraint handling) on the comparison between the considered methods, we consider its simplest variant, without the repair operators except the deletion of thin components, and transform the constrained problem into unconstrained by penalizing the volume with use of a suitable Lagrange multiplier λ_V :

$$f(\mathbf{z}) = d_{max}(\mathbf{z}) + \lambda_V V(\mathbf{z}), \quad (8.4)$$

where $f(\mathbf{z})$ is the cost function to be minimized.

In the first part of this section, we evaluate the properties of H-EA-LSM using the compliance gradient obtained with use of the global ESL method as an approximation of the gradient of intrusion $\frac{\partial d_{max}}{\partial \mathbf{z}}$ in a dynamic crash case. In the second part, based on the samples collected via finite-differencing, we train ML models of sensitivities and use them in an optimization of the crash scenario with H-EA-LSM. This demonstrates the potential of data-driven sensitivity modeling, which is a generic approach that can be used when no suitable physical surrogate models can be found.

Global equivalent static loads method

In order to obtain the approximate sensitivities via the global ESL approach, we consider the linear elastic counterpart of the crash case (Figure 8.3). For this case, the gradient of the following cost function is used as the approximation of $\frac{\partial d_{max}}{\partial \mathbf{z}}$:

$$f_{ESL}(\mathbf{z}) = C(\mathbf{z}) + \lambda_{ESL}V(\mathbf{z}), \quad (8.5)$$

where λ_{ESL} is again a suitable multiplier resulting in the required volume fraction (50%) for the linear elastic case only. The analytical expressions for both $\frac{\partial C}{\partial \mathbf{z}}$ and $\frac{\partial V}{\partial \mathbf{z}}$ were given in Section 6.1.1.

Please note that the computation of approximate sensitivities in this case requires an additional linear elastic FEA to compute the compliance $C(\mathbf{z})$. However, in general case, the computational cost of the static FE simulation can be assumed to be significantly smaller than that of the explicit crash simulation. Since the individuals for the gradient improvement are selected randomly, the hybrid approach requires exactly the same number of crash analyses per generation as the standard EA-LSM. If an increase of convergence velocity can be obtained with this approach, the higher computational costs due to the additional static simulation will be justified.

Finally, the approximate gradients are used in two variants of H-EA-LSM:

- H-EA-LSM-0.2-ESL – Hybrid ES(10,70) with a single step size and 20% of offspring individuals selected randomly for the improvement with the approximate sensitivities.
- H-EA-LSM-0.5-ESL – Hybrid ES(10,70) with a single step size and 50% of offspring individuals selected randomly for the improvement with the approximate sensitivities.

The algorithms mentioned above are compared against EA-LSM using standard ES(10,70) with a single step size. The choice of the population sizes follows the general recommendations by Bäck (1996). In order to focus on the evaluation of the influence of the approximate gradients on the evolutionary search, the investigations in this section are narrowed down exclusively to the simple variant of ES and its hybridizations.

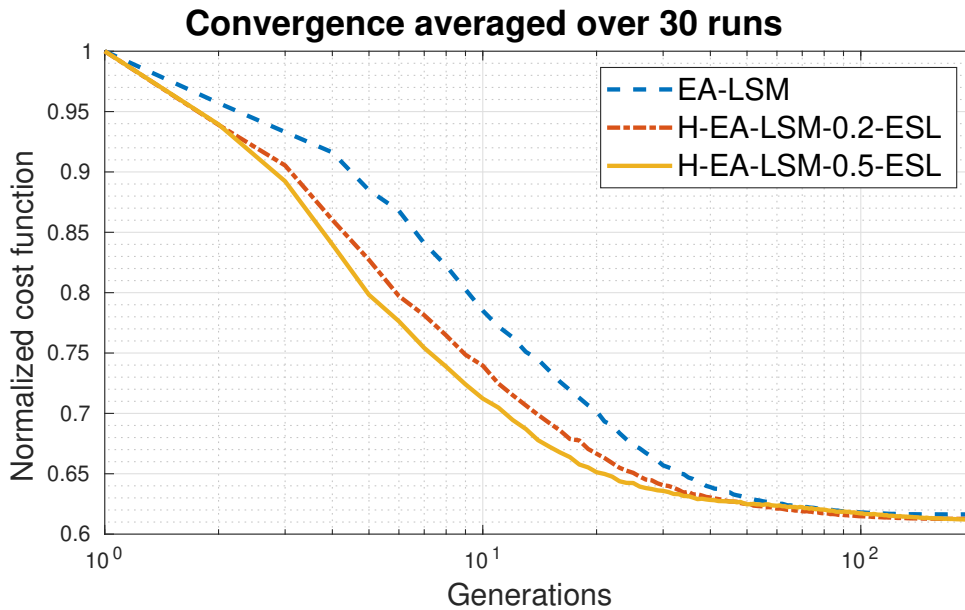
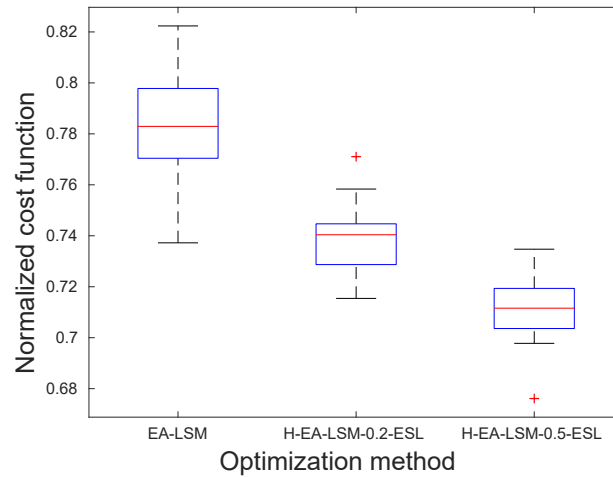


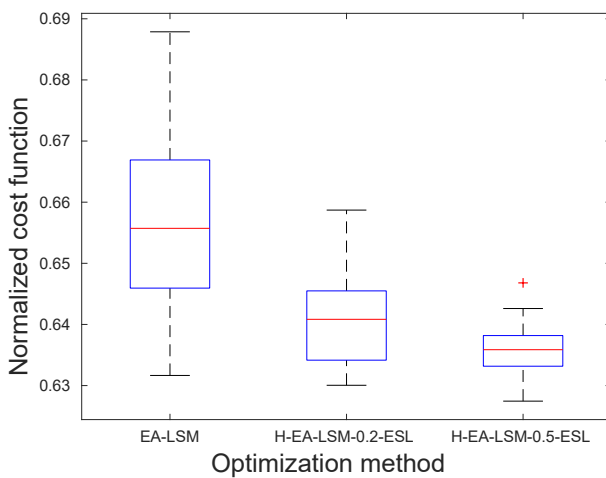
Figure 8.6 Convergence of the cost function in terms of optimization generations averaged over 30 runs of EA-LSM, H-EA-LSM-0.2-ESL, and H-EA-LSM-0.5-ESL. The cost function is normalized with the initial value.

Figure 8.6 presents the convergence plots for each of the considered methods, averaged over 30 optimization runs. One can easily note the increase of convergence velocity due to utilization of the approximate gradients, especially in the critical, initial phase of the optimization. For instance, to reach a similar level of the cost function as H-EA-LSM-0.5-ESL after 10 generations, EA-LSM needs almost twice as many generations, implying ca. 700 additional crash simulations. After 30 generations, H-EA-LSM-0.5-ESL is almost 20 generations ahead of EA-LSM, implying ca. 1400 less crash FEAs. Finally, the comparison between H-EA-LSM-0.2-ESL and H-EA-LSM-0.5-ESL suggests that by increasing the fraction of gradient-improved individuals in the considered optimization scenario, the convergence velocity can be generally improved. These results are consistent with the ones obtained for the energy absorption maximization problem considered in our previous works (Bujny (2015); Bujny et al. (2017b)), where even larger benefits due to hybridization were obtained. Most probably, this effect was observed because of a much stronger similarity between energy absorption and compliance gradients.

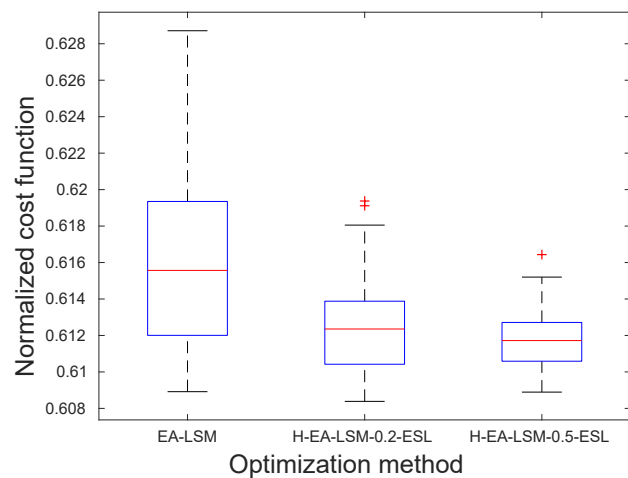
To investigate the distribution of the cost functions among the optimization runs, Figure 8.7 shows box plots for each of the algorithms after 10, 30, and 200 generations. The plots confirm the statements about the benefits of the hybrid approaches from above. A formal statistical evaluation based on Wilcoxon test allows for rejection of the null hypothesis about equality of medians among each of the methods at 5% significance level after 10, 30, and 200 generations. The only exception is the comparison between H-EA-LSM-0.2-ESL and H-EA-LSM-0.5-ESL after 200 generations, where the difference between medians is not statistically significant, implying that both of the hybrid approaches yield designs of similar performance at the end of the optimization. Indeed, for both of the hybrid approaches, in all of the 30 optimization runs, only designs of the type shown



(a) After 10 generations.



(b) After 30 generations.



(c) After 200 generations.

Figure 8.7 Distribution of the cost function values, in 30 optimization runs, after 10, 30, and 200 generations for EA-LSM, H-EA-LSM-0.2-ESL, and H-EA-LSM-0.5-ESL. The cost function is normalized with the initial value. The line in center of each box indicates the median, while the top and bottom edges of the boxes correspond to the 25th (q_1) and 75th (q_3) percentiles of the normalized compliance values, respectively. The whiskers, represented using dash black lines, extend to the data points of the highest and lowest compliance values, excluding outliers, which are marked using the red '+' symbol. The criterion for classification of a point as an outlier requires it to be greater than $q_3 + 1.5(q_3 - q_1)$ or lower than $q_1 - 1.5(q_3 - q_1)$.

in the first column, second row of Table 8.2 were obtained. In contrast, EA-LSM resulted in this type of design only in ca. 50% of cases. Interestingly, running a gradient-based optimization for the equivalent static problem only, with the same step size as used in the improvement step in H-EA-LSM, resulted in an inferior design type, presented in the second row of Table 7.6. This shows that H-EA-LSM is able to benefit from the gradient information, but at the same time, it has much better global search properties compared to a pure gradient-based approach.

Finally, to give more insight into the utilization of the approximate gradients, Figures 8.8 and 8.9 show the average number of gradient-improved individuals selected as new parents and the average frequency of the gradient-improved individual being the best, respectively. The algorithm uses the gradient mainly in the initial phase of the optimization, up to ca. 20 generations, and

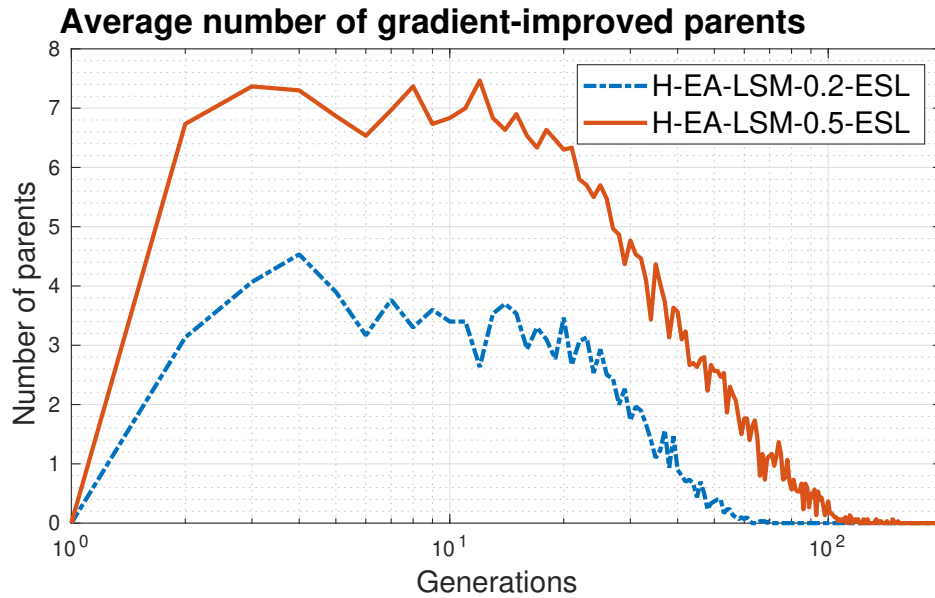


Figure 8.8 Average number of gradient-improved individuals in the parent population. Results based on 30 optimization runs of H-EA-LSM-0.2-ESL and H-EA-LSM-0.5-ESL.

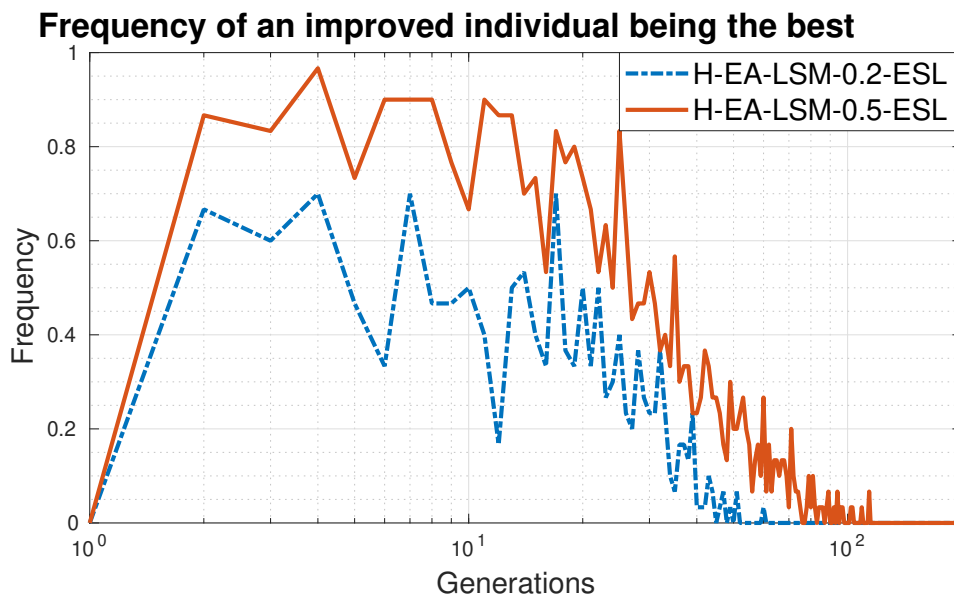


Figure 8.9 Average frequency of the gradient-improved individual being the best. Results based on 30 optimization runs of H-EA-LSM-0.2-ESL and H-EA-LSM-0.5-ESL.

then the use of gradient gradually decreases. After ca. 30 generations, the ratio of the gradient-improved individuals and non-improved ones in the parent population represents approximately the frequency of gradient improvements in the offspring population. Around generation 100 the number of gradient-improved individuals decreases almost to zero, indicating that the gradient-improvements actually decrease the performance of individuals. To some extent, this might be caused by the fixed step size in the gradient improvements, which does not allow for an accurate fine-tuning of the final solution. This was also the case for the evaluation of H-EA-LSM on linear

elastic cases (see Figure 7.18, Section 7.1.2). On the other hand, in the final optimization phase, the ESL gradient might be just too inaccurate for the considered crash case.

To summarize, the use of approximate gradients from a physical surrogate such an ESL case can improve the performance of an evolutionary-based search, which we demonstrated for the intrusion minimization problem. In this case, the benefit from using approximate gradients was high especially at the beginning of the optimization. Later on, when gradient approximations were not accurate enough, H-EA-LSM could successfully reject gradient-modified individuals. The obtained result suggests that an adaptive strategy, where the frequency of gradient improvements would be proportional to their success rate, would be an interesting future extension of the method.

Predicting sensitivities

For many objective functions and constraints considered in crashworthiness optimization, it is difficult to find an appropriate physical surrogate for the approximation of sensitivities. This motivates a development of alternative approaches, using ML techniques. This section targets investigating the potential of using such an approach in H-EA-LSM. The data generation process as well as the sensitivity models used here were developed within the master's thesis of Krischer (2018), supervised by the author of this dissertation.

The dataset of numerical sensitivities was obtained via three independent SD optimization runs for the transverse bending problem with impact velocities of 20, 10, and 5 [m/s]. No symmetry condition was imposed. In all of the cases, a fixed step size of $s = 0.05$ was used, and 50% volume fraction was targeted via a dual optimization technique (Arora (2012)). All of the optimization runs were restricted to 130 iterations. The sensitivity samples were obtained via the central finite difference scheme, requiring additional 160 FE evaluations per optimization iteration. This resulted in a total number of 6240 samples per model. Please note that the costly sampling process is run just once, for a given objective function, and the trained ML models can be used for solving multiple optimization problems at no additional cost.

In the next step, the sensitivity models, for each type of design variables, were fitted using a randomly selected set of 5760 samples. For simplicity, only second order linear regression model based on the Component State Features (CSFs) given by Equations (6.20) and (6.21) was considered. An initial investigation of the input features based on Spearman's rank correlation coefficient (Myers and Well (2003)) showed that the combined features corresponding to different coordinate directions are not important. As a result, the CSFs containing combined terms, i.e. $u_{e,i}u_{e,j}$ with $i \neq j$, were neglected, leading to a total number of 20 CSFs. The R^2 values for the fitted models, calculated on the remaining test samples, are shown in Table 8.12.

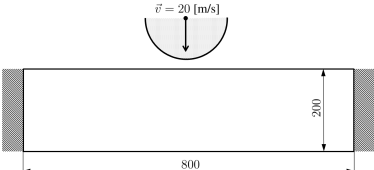


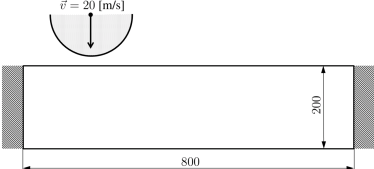


The accuracy of the obtained models is not very high. Probably, more complex models as well as more training data would be needed to improve the results. However, one should note that the

Table 8.12 R-squared values calculated on the test data for the sensitivity models trained by (Krischer (2018)).

Model					
	$\widetilde{\frac{\partial f}{\partial x_0}}$	$\widetilde{\frac{\partial f}{\partial y_0}}$	$\widetilde{\frac{\partial f}{\partial \theta}}$	$\widetilde{\frac{\partial f}{\partial l}}$	$\widetilde{\frac{\partial f}{\partial t}}$
R^2	0.10	0.37	0.21	0.24	0.66

models are used for prediction of individual components of the gradient ∇f . As a result, the accuracy of the predicted magnitudes of individual gradient components is of secondary meaning. What counts for the gradient-based optimization is the accuracy of the prediction of the resultant gradient, or, more precisely, only its direction. To evaluate that, Krischer (2018) carried out gradient-based optimizations based on the gradient models. The resulting designs, together with the reference solutions, are shown in Table 8.13.

Table 8.13 Final designs for gradient-based optimization with the Steepest Descent method using the pre-trained sensitivity models (Krischer (2018)). Similarly to the problem considered in this section, a cost function of the form given by Equation (8.4) was minimized. The load case shown in the second row of the table was never used during the sampling process.

Optimized design		
Load case	Reference solution	SD using sensitivity models
		
		

The results show that, despite relatively low accuracy of the individual models, a pure-gradient based optimization can lead to meaningful designs, comparable to the reference structures. This shows the potential of the TO by predicting sensitivities (Aulig and Olhofer (2014b); Aulig (2017)) also for the MMC-based representation. However, the quality of the final designs, as well as the oscillatory character of the performed optimization runs, suggest using the sensitivity models in H-EA-LSM as a more promising approach.

Therefore, as a next step, 30 optimization runs for each of the following two hybrid approaches were carried out:

- H-EA-LSM-0.2-PS – Hybrid ES(10,70) with a single step size and 20% of offspring individuals selected randomly for the improvement with the approximate sensitivities.
- H-EA-LSM-0.5-PS – Hybrid ES(10,70) with a single step size and 50% of offspring individuals selected randomly for the improvement with the approximate sensitivities.

The averaged convergence plots for EA-LSM, H-EA-LSM-0.2-PS, and H-EA-LSM-0.5-PS, is shown in Figure 8.10. The corresponding box plots after 10, 30, and 100 generations are depicted in Figure 8.11.

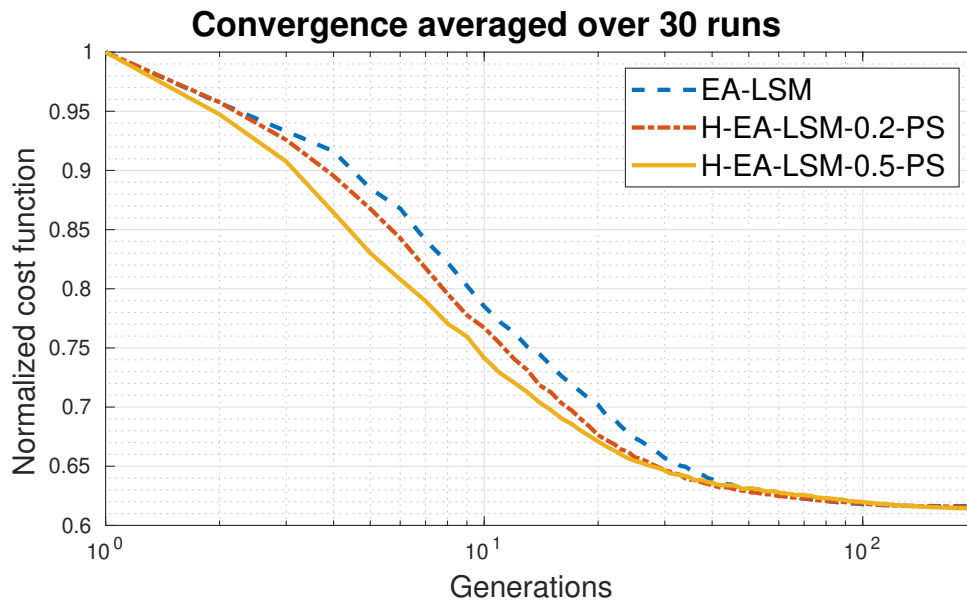
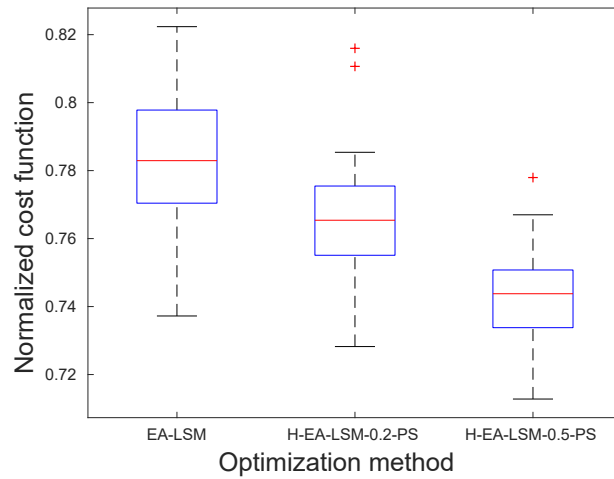


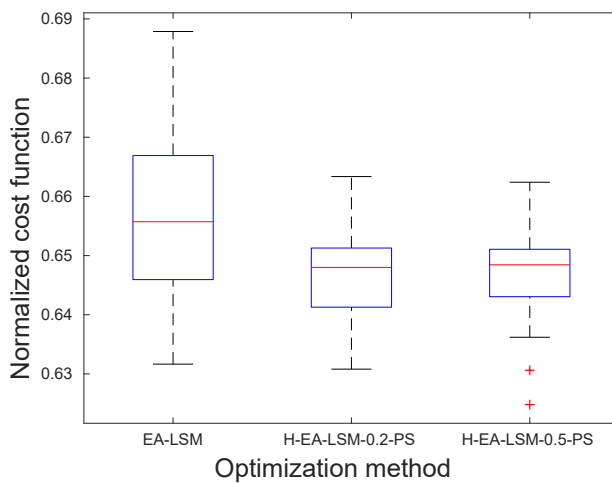
Figure 8.10 Averaged convergence plots of the cost function in terms of optimization generations for EA-LSM, H-EA-LSM-0.2-PS, and H-EA-LSM-0.5-PS. The cost function is normalized with the initial value.

The results show that, especially in the initial phase of the optimization, the hybrid approaches converge faster than EA-LSM. A Wilcoxon rank sum test rejected the null hypothesis about the equality of the medians among all of the methods after 10 generations at 5% significance level. After 30 generations, there is no statistically significant difference between H-EA-LSM-0.2-PS and H-EA-LSM-0.5-PS, but both hybrid approaches are significantly better than EA-LSM. Finally, after 200 generations, there is no statistically significant difference among all of the considered methods.

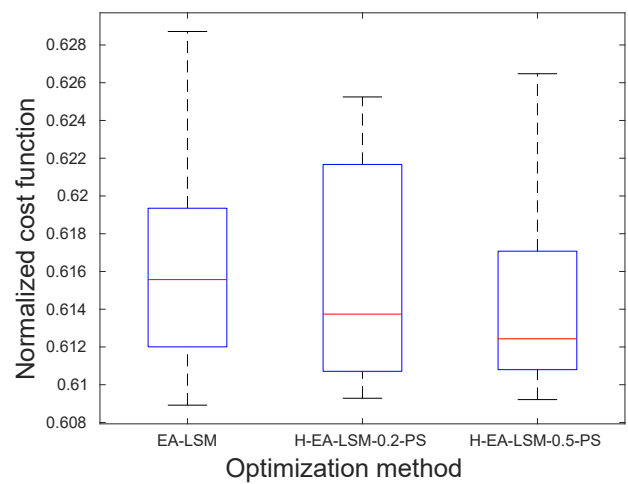
One can easily note that the performance of H-EA-LSM-0.5-PS deteriorates over time. This is due to an inaccuracy of the gradient approximation in the final phase. Since a fixed fraction of 50% offspring individuals are affected by that, effectively, the optimizer can use only half of the population, which deteriorates the search. For H-EA-LSM-0.2-PS, this effect is much less affecting the optimization process. In fact, these unfavorable effects could be very easily eliminated at no additional computational cost. For consistency with the optimization using the ESL approach, as well as to keep the offspring population the same as in EA-LSM, the algorithm first mutates the parent individuals to generate offspring, evaluates all of the offspring individuals to be able to



(a) After 10 generations.



(b) After 30 generations.



(c) After 200 generations.

Figure 8.11 Distribution of the cost function values after 10, 30, and 100 generations for EA-LSM, H-EA-LSM-0.2-PS, and H-EA-LSM-0.5-PS. The cost function is normalized with the initial value. The line in center of each box indicates the median, while the top and bottom edges of the boxes correspond to the 25th (q_1) and 75th (q_3) percentiles of the normalized compliance values, respectively. The whiskers, represented using dash black lines, extend to the data points of the highest and lowest compliance values, excluding outliers, which are marked using the red '+' symbol. The criterion for classification of a point as an outlier requires it to be greater than $q_3 + 1.5(q_3 - q_1)$ or lower than $q_1 - 1.5(q_3 - q_1)$.

compute CFSs, and then improves a part of already evaluated population. The improved individuals are evaluated again to take part in the selection step. One could simply replicate the mutated (and already evaluated) individuals selected for improvement and perform gradient improvement on the copies. As a result, with an extended offspring population, the performance of the hybrid approaches would be always at least as good as EA-LSM. Since this section aims exclusively for investigation of the addition of sensitivity models on the evolutionary search, this approach is not studied here.

Interestingly, when comparing the final designs, the best topology type (first column, second row, Table 8.2) was reached by H-EA-LSM-0.5-PS in ca. 90% of cases, while for H-EA-LSM-0.2-PS in 80% of cases. As mentioned in the previous section, EA-LSM came up with this design only in ca.

50% of cases. This suggests that the inaccurate gradients introduce only some oscillations of the objective function for H-EA-LSM-0.5-PS, but the global search properties of this strategy are still the best. As suggested before, an adaptive strategy for choosing the fraction of gradient-improved individuals could be very promising here, as well.

All in all, in this section, we showed that ML can be used to model the gradients of intrusion w.r.t. the parameters of MMCs, thus providing a generic approach for sensitivity modeling. The H-EA-LSM approach offers the possibility to benefit from the approximate gradients, even when their accuracy is limited. In the future, one could consider also using several different gradient approximations without a negative influence on the optimization process. However, more experiments would be needed to show that the sensitivity models can be obtained for arbitrary optimization criteria.

8.1.3. Kriging-guided level set method (KG-LSM)

This section continues the experimental evaluation of KG-LSM, started in Section 7.1.3. In particular, the examples studied below aim for answering the question of feasibility of using such an approach in TO of nonlinear crash cases.

Similarly to Section 7.1.3, the results presented here are based on the master's thesis of Raponi (2017), supervised by the author of this dissertation, and the following conference and journal publications (Raponi et al. (2017, 2019a)).

To evaluate KG-LSM on nonlinear crash cases, the transverse bending example as described at the beginning of this chapter (Figure 8.1) is used. The performance of the method is tested based on a level-set parametrization using 6 MMCs, with a symmetry condition as shown in Figure 8.12. In the considered case, all of the parameters defining MMCs are changed by the optimizer, resulting in a total number of 15 design variables. Additionally, CEI is used to enforce the 50% volume fraction constraint. To guarantee sampling of only meaningful designs, the structural feasibility constraint requires connectivity of the topology to the left and the right support (Figure 8.1). As a result, out of 200 candidate sample points generated using OLHS, on average, in 30 optimization runs, only 35 samples are evaluated using FE simulations and used for fitting the initial surrogate model.

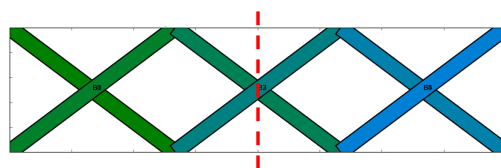


Figure 8.12 Reference design for the transverse bending crash problem with 15 design variables (Raponi (2017); Raponi et al. (2019a)).

For comparison, the EA-LSM algorithm using CMA-ES with $\mu = 6$ parent and $\lambda = 12$ offspring individuals (see Appendix B.2) is used. By default, the initial population is created based on the reference design shown in Figure 8.12. Additionally, KG-LSM is compared against EA-LSM using the best design found during the DoE phase of KG-LSM as a reference for creation of the initial population. This strategy is referred to as DoE-EA-LSM. Due to the non-deterministic character of all of the methods discussed here, for KG-LSM, EA-LSM, and DoE-EA-LSM, 30 optimization runs are performed and the averaged results are used for comparison. The performance of the methods is compared in terms of evaluations, assuming a fixed budget of 500 FE simulations.

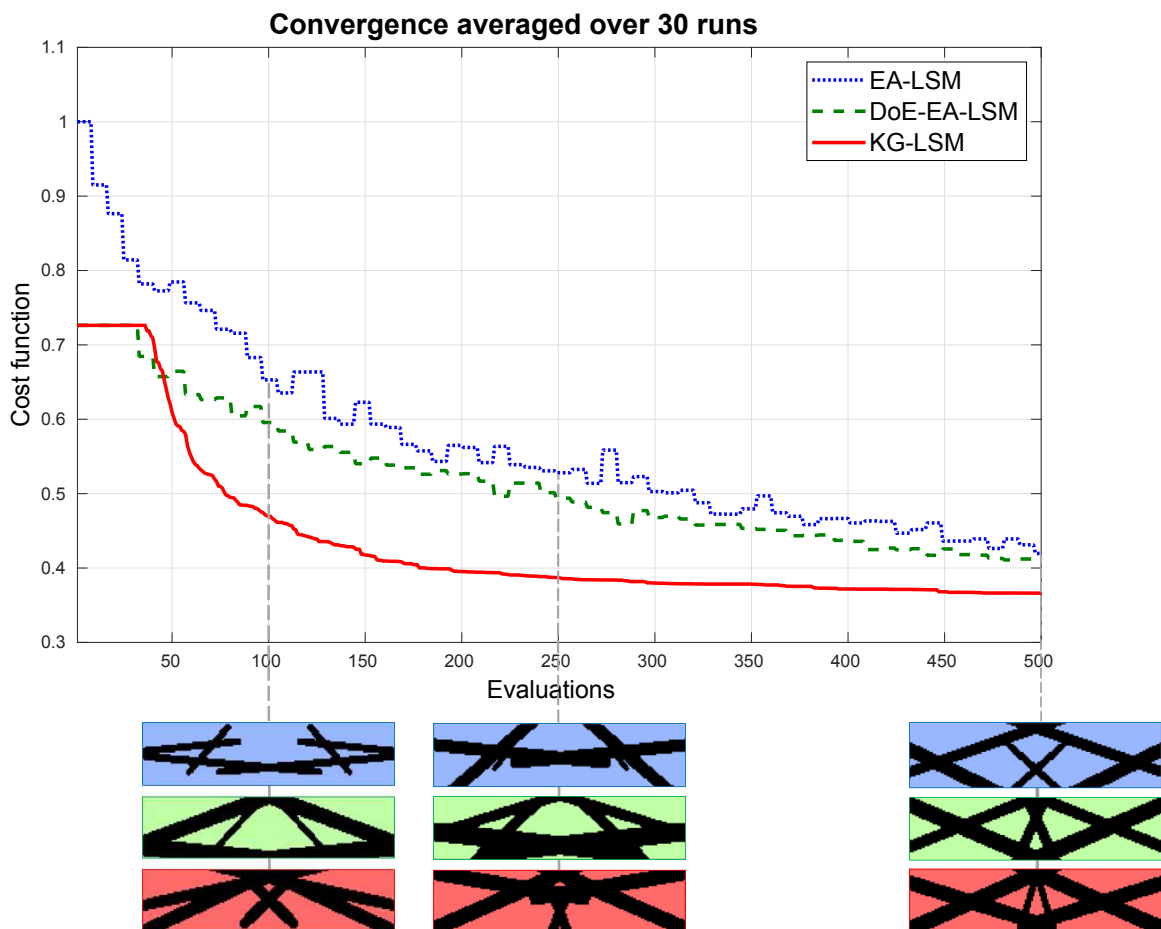


Figure 8.13 Comparison of the convergence behavior of KG-LSM, EA-LSM, as well as DoE-EA-LSM (EA-LSM using the best design from the DoE as the initial reference design) on a crash optimization problem with 15 design variables, averaged over 30 optimization runs (Raponi et al. (2019a)). The values of the cost function are normalized based on the performance of the reference design (Figure 8.12). Additionally, the evolution of the best designs out of 30 runs for each of the algorithms is shown (background colors corresponding to the curves).

Figure 8.13 presents the averaged convergence plots. The results show that KG-LSM converges on average considerably faster than both, EA-LSM as well as DoE-EA-LSM. The latter suggests that the better performance of KG-LSM is independent of the superiority of the designs found in the DoE phase over the reference design used for the initialization of EA-LSM. The distribution of the values of the cost functions for all of the 90 optimization runs are summarized in a form of

box plots shown in Figure 8.14. The statistical comparison of the methods after 100, 250, and 500 evaluations, based on the Wilcoxon rank sum test (Gibbons and Chakraborti (2011)), allows for rejection of the null hypothesis about equality of the medians for KG-LSM and EA-LSM, as well as KG-LSM and DoE-EA-LSM, which confirms superiority of KG-LSM for the considered test case.

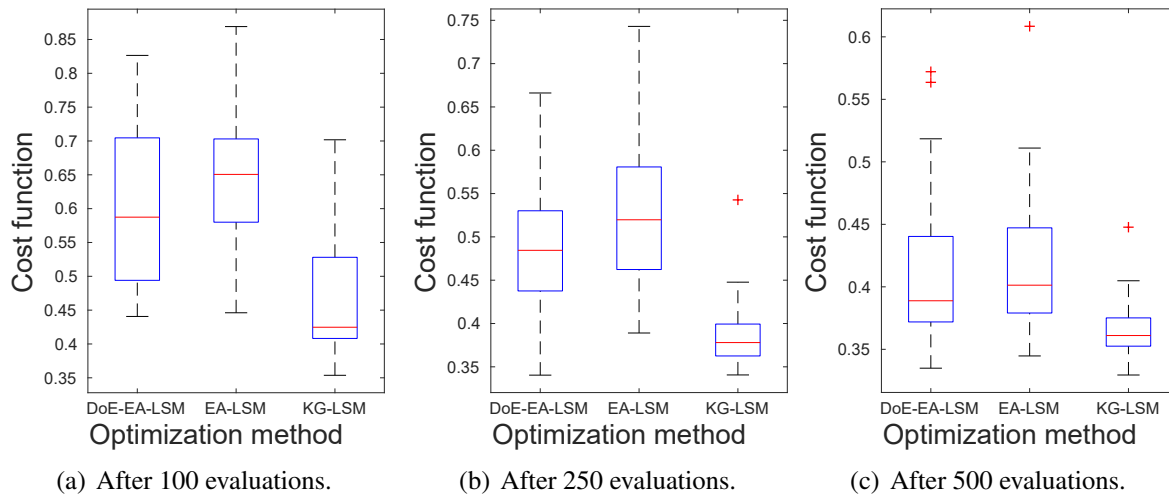


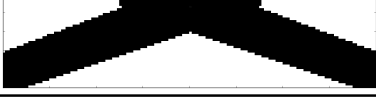


Figure 8.14 Box plots for 30 optimization runs of KG-LSM, EA-LSM, and DoE-EA-LSM (Raponi et al. (2019a)). The values of the cost function are normalized based on the performance of the reference design (Figure 8.12). The line in center of each box indicates the median, while the top and bottom edges of the boxes correspond to the 25th (q_1) and 75th (q_3) percentiles of the normalized compliance values, respectively. The whiskers, represented using dash black lines, extend to the data points of the highest and lowest compliance values, excluding outliers, which are marked using the red '+' symbol. The criterion for classification of a point as an outlier requires it to be greater than $q_3 + 1.5(q_3 - q_1)$ or lower than $q_1 - 1.5(q_3 - q_1)$.

Finally, Table 8.14 shows the main types of topologies found in 30 optimization runs of each of the methods. Additionally, based on visual inspection, the frequency of obtaining each of the topology type by different methods is calculated. The results show that KG-LSM is able to find much more frequently the superior design types. In contrast, EA-LSM and DoE-EA-LSM much more often end up in local optima of considerably worse performance. One should note also the qualitative differences between the most frequent designs obtained in this section and the previous results for EA-LSM, presented in Section 8.1.1 (Table 8.2). Most probably, this results from the reduced dimensionality of the representation used in this section. The structural attainability of the representation using 6 MMC is considerably lower than for the one using 16 MMCs. As a result, the relative performance of similar design concepts with two different representations can change, which seems to be the case when comparing Table 8.14 and Table 8.2 (designs for the initial velocity of 20 [m/s]).

The results presented in this section show that KG-LSM can be successfully used in the optimization on nonlinear crash problems with a limited number of design variables. The surrogate-assisted approach outperforms considerably EA-LSM in terms of both, convergence velocity and the quality of the final designs. In fact, this is quite surprising in the context of the evaluation of KG-LSM on the linear elastic case with 15 design variables, discussed in Section 7.1.3, where, especially in

Table 8.14 Three dominant topology types obtained with KG-LSM, EA-LSM, and DoE-EA-LSM, and the frequency with which they occur in 30 optimization runs. The topologies are ordered starting from the best performing designs on the top to the worst ones at the bottom (Raponi et al. (2019a)).

Topology type	Method		
	KG-LSM	EA-LSM	DoE-EA-LSM
	3%	0%	3%
	23%	7%	13%
	17%	57%	23%

the final phase of the optimization, it was performing considerably worse than EA-LSM. Potentially, this might be associated with the stronger multi-modality of the crash problem (Raponi et al. (2019a)). EA-LSM distributes the initial population of individuals around the reference design and performs a much more local search than KG-LSM. As a result, for highly multi-modal problems, it might fail to locate the superior optima much more frequently than KG-LSM.

All in all, KG-LSM seems to be suitable for solving crash TO problems. Thanks to its superior performance in terms of (potentially very costly) evaluations, it is very promising in the context of industrial applications, where the budget of FE simulations is often heavily restricted. The main limitation of the method, as demonstrated in Section 7.1.3, remains the dimensionality of the optimization problems, which should be kept on the minimal level providing satisfactory structural attainability for identification of novel design concepts.

8.1.4. Adaptive evolutionary level set method (A-EA-LSM)

This section discusses briefly an application of the A-EA-LSM to a crash problem. The main goal here is to investigate if such an approach is suitable for this type of problems and understand the overall optimization process. Again, we consider the transverse bending problem with 50% volume fraction constraint and the initial velocity of impactor of 20 [m/s].

Following the data generation procedure described in Section 7.1.4, resulting in a total number of 129,941 samples, and using the same parameters as for the linear elastic problem (Table 7.13, Section 7.1.4), a neural network classifier was trained for the pairwise comparison task. The accuracy of the network's predictions on the test data, corresponding to 10% of the total number of samples, reached 79%. The classifier was used for the prediction of the ranking of topology variations in A-EA-LSM.

Figure 8.15 shows the convergence of the normalized intrusion for an exemplary optimization run. The configuration of the optimization problem for this case follows the one for the static problem, given in Table 7.12. The corresponding illustration of different species during the optimization run is presented in Figure 8.16.

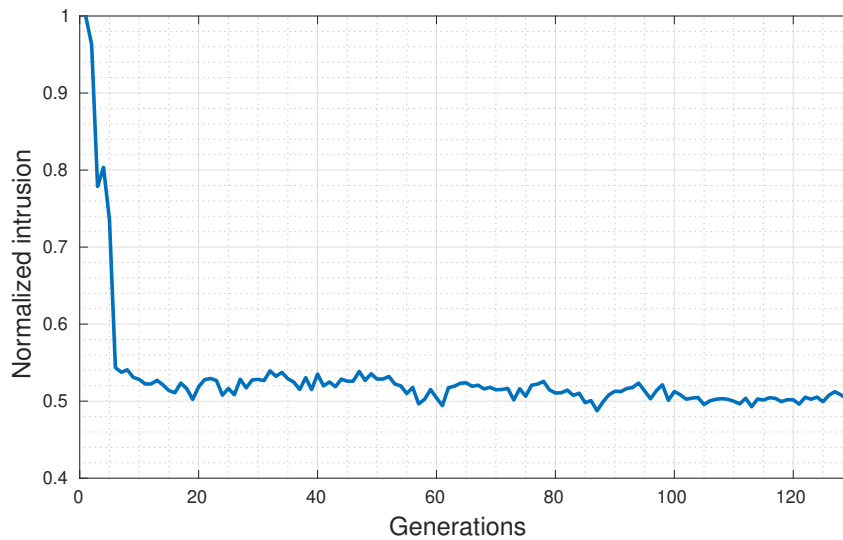


Figure 8.15 Convergence plot of the intrusion normalized with the initial value for an exemplary optimization run of A-EA-LSM for a crash problem.

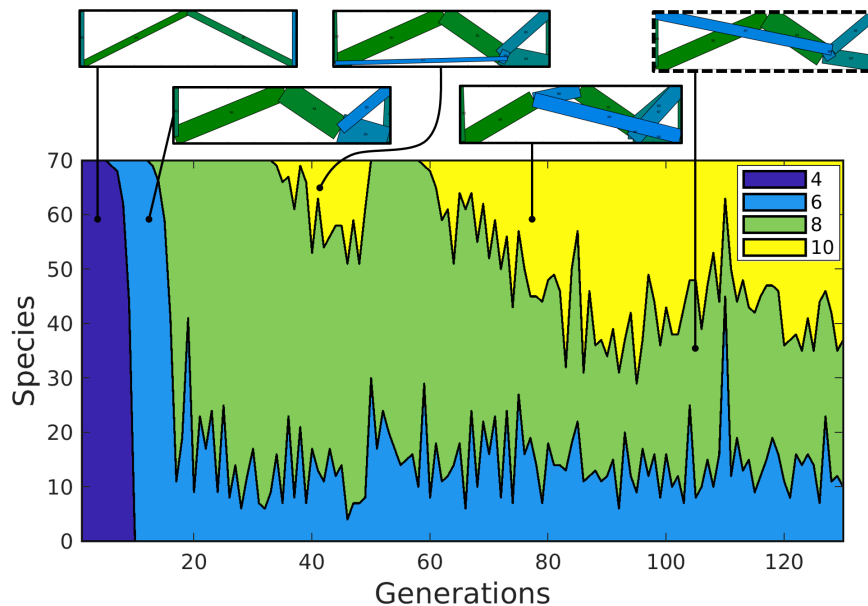


Figure 8.16 An exemplary run of the A-EA-LSM with the learning-based topology variations illustrating the working principles of the method for a crash problem. Visualization of speciation during the optimization with color-coded species (top). Topologies are divided into species depending on the number of components specified in the legend. The final best design is shown in a dashed frame.

One can easily note a very fast descent of the intrusion in the initial phase of the optimization. In that phase, the optimizer can take advantage of the low-dimensional parametrization of the design as well as benefit from the first topology variation. Within the first 20 generations, the optimizer

finds already a good design in the lower-dimensional search space. Subsequent topology variations are hardly able to improve the objective, although they gradually lead to topologies qualitatively more similar to the solutions known from the high-dimensional parametrizations (Table 8.2). The final design is close, but not identical to the optima found by a pure EA-LSM (Table 8.2). Perhaps, for a rising number of feasible edge-node combinations, the accuracy of the classifier performing pairwise comparisons is not high enough to predict the necessary intermediate topology variations. However, please note that the predicted topology variations (Figure 8.16) are meaningful from the mechanical point of view. The accuracy of the ranking prediction could be potentially improved by collecting more data, training more complex ML models, or including more input features. On the other hand, using exclusively the information about the structural state of a given node and edge might be not sufficient to judge where to introduce the topology variation. In this case, the information about the overall state of the structure might be necessary. To address that, initial experiments with use of Convolutional Neural Networks (CNNs) (Bishop (2007)) led to some promising results, but more investigations would be needed to prove the usefulness of such an approach. Another problem associated with the solution based on CNNs is a very large amount of data needed for training the models. Finally, there might be a more fundamental problem associated with the use of adaptive schemes in TO. Namely, it is not clear if an optimal design can be always reached by performing a single topology variation at a time. In fact, it might frequently happen that a transition from a simpler (good) design to a more complex (very good) design requires an intermediate topology, whose performance would be considerably worse than the one of the simplest design. The adaptive approach in the form proposed in this work would simply reject the intermediate (bad) design, as it happens for the third topology variation shown in Figure 8.16, and would not allow for the transition to the desired (complex) topology. Interestingly, this probably happens also for the state-of-the-art GHA method (Ortmann and Schumacher (2013)), but, since, to the best of our knowledge, GHA was never evaluated on linear elastic or crash problems with known reference solutions, this cannot be explicitly shown. Anyway, in the practical context, one wants just to improve the current design and not necessarily find the global optimum, which can be accomplished both by GHA and A-EA-LSM. The main advantage of the approach proposed in this work is its generic character, with the rules not based on the expert knowledge, but extracted explicitly from the sampled data with use of ML models.

To summarize, in this section, we demonstrated that A-EA-LSM can improve the performance of crash structures through successive adaptations of the representation. Thanks to the low initial dimensionality of the optimization problem, it can converge very fast at the beginning of the optimization process. The predictions of topology variations with an ML model lead to gradual improvements of the design. However, it is not clear if this type of approach can lead to structures identical to the ones obtained with EA-LSM and more investigations are necessary to evaluate that.

8.2. Three-dimensional topology optimization for crashworthiness

In the following section, 3D variant of EA-LSM, as described in Chapter 5, is evaluated on a crash optimization problem with 144 design variables. This section aims to demonstrate the ability of EA-LSM to solve strongly nonlinear, high-dimensional crash TO problems in 3D. This type of problems, to the best of our knowledge, has never been addressed with use of evolutionary optimization methods in the literature. As a result, experimental verification of the proposed optimization concept would be valuable to demonstrate its potential.

8.2.1. Optimization problem

The experiments are carried out based on the test case depicted in Figure 8.17. As in case of the 2D optimizations, the test case is inspired by the side pole impact from the Euro NCAP. In the optimized scenario, a rigid pole impacts in the middle of a beam structure, clamped at both ends. The whole beam constitutes the design domain, which is occupied by 3D MMCs, constrained by two symmetry planes as shown in Figure 8.17 (dashed lines). Initially, the MMCs are distributed as shown in Figure 8.18. The FE model used in the optimizations is presented in Figure 8.19. It is the same model as in the work of Aulig et al. (2015), with different boundary conditions. The setup of the test case is defined in Table 8.15 and 8.16.

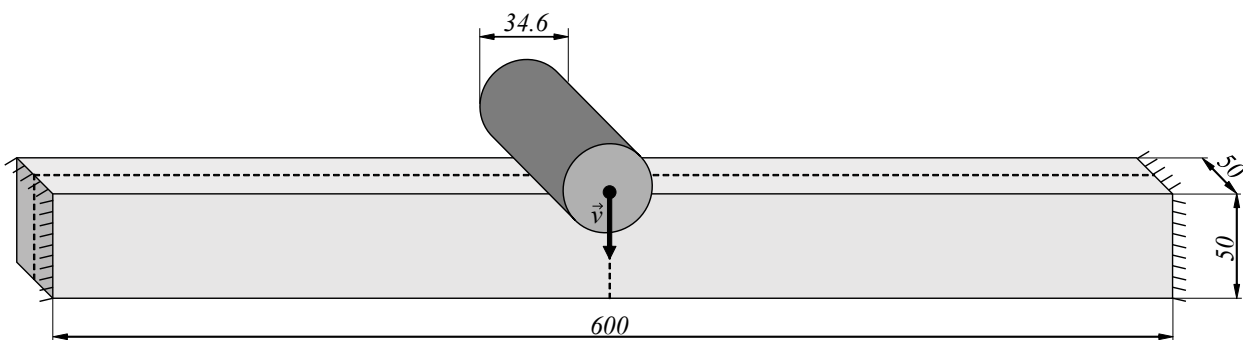


Figure 8.17 Design domain and boundary conditions for the 3D nonlinear dynamic transverse bending crash case. Dimensioning in mm. Dashed lines indicate the location of the symmetry constraints applied to the distributions of MMCs.

In contrast to Section 8.1, only the intrusion minimization problem with a mass constraint is considered. The formal definition of the optimization problems considered here is identical to the one given by Equation (8.1).

In all of the considered cases, a 50% mass fraction constraint is considered. The tests are carried out for three different impact energy levels: 409 [J], 2000 [J], and 10000 [J], resulting in different amounts of plastic deformation. Different impact energies are obtained by scaling the mass of the impactor, while keeping the impact velocity constant, i.e. equal to 20 [m/s].

For the comparison of the results obtained in this section, both HCA and a simple variant of the (global) ESL method are used. As shown already in Section 8.1, both HCA and ESL perform very

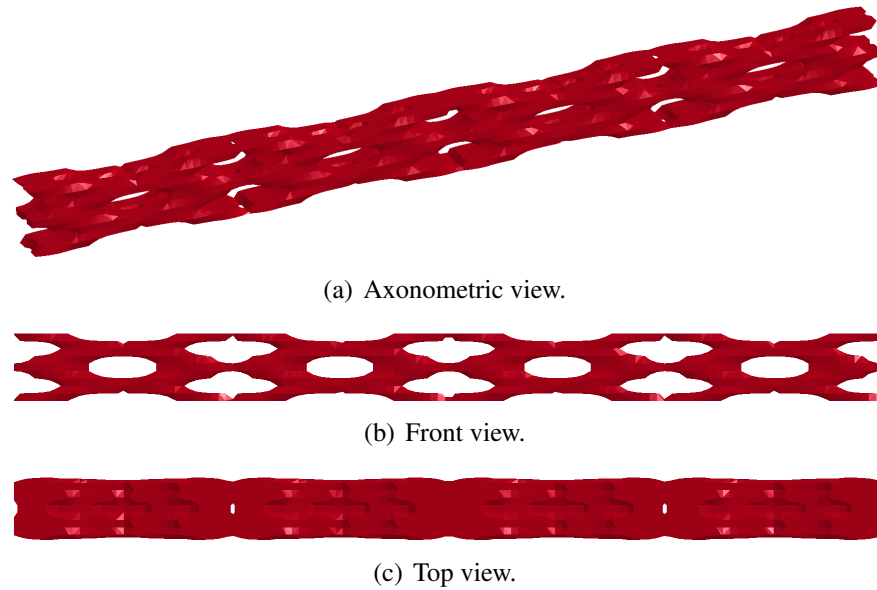


Figure 8.18 0th LSF iso-contour for the initial design used in the 3D nonlinear dynamic transverse bending crash case.

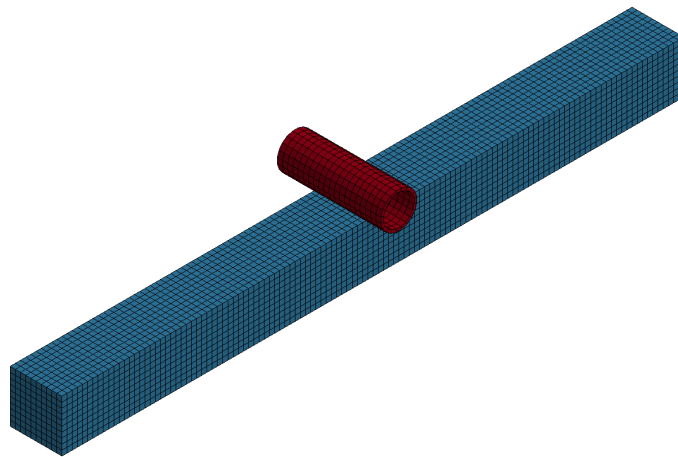


Figure 8.19 FE model used in the simulations of the 3D nonlinear dynamic transverse bending crash.

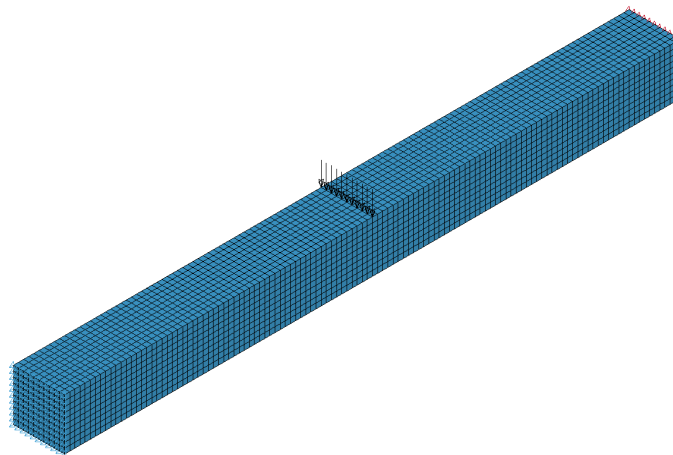
well on this type of problems, therefore they are suitable for validation of the results obtained with EA-LSM. The optimizations with the HCA method are carried out with use of LS-TaSC 3.2, with the default settings. The post-processing of the TO results follows the method described in Section 8.1.

Similarly to the approach described in Section 8.1, in the simple version of the ESL method, we replace the dynamic crash load with a distributed load on the segment corresponding to the location of the contact between the structure and the impactor in the dynamic crash scenario. The FE model with the boundary conditions after the simplification is depicted in Figure 8.20. The material model is changed to the linear elastic, with Young's modulus of $E = 2.7 \cdot 10^3$ [MPa] and Poisson ratio $\nu = 0.33$. Based on the simplified model, the topology is optimized with use of the gradient-based SIMP approach with a filter size of 15 [mm] and the material penalization power $p = 3$.

Table 8.15 Setup of the nonlinear dynamic 3D crash test case. For simplicity, the strain rate dependency is not considered here.

Property	Symbol	Value	Unit
Initial pole velocity	v	20.0	m/s
Pole mass	m_p	10.0	kg
Total mass of the design domain	m_{tot}	4.05	kg
Required structure mass	m_{req}	2.025	kg
Mesh resolution	-	120 x 10 x 10	-
Solver	-	LS-DYNA R7.1.1	-
Element formulation	ELFORM	Constant stress solid element	-

Since the considered optimization problem cannot be addressed directly, the method optimizes the structure to achieve the minimal compliance. As for the HCA method, the optimized structure is post-processed using the approach described in Section 8.1. Finally, the performance of the resulting topology is evaluated on the nonlinear dynamic crash case as defined at the beginning of this section.

**Figure 8.20** Boundary conditions for the ESL case corresponding to the considered crash problem.

8.2.2. Results

The structures optimized with EA-LSM, in five independent optimization runs, for the impact energy of 409 [J] are presented in Table 8.17. For this scenario, EA-LSM based on a standard ES(6,39) with $\mu = 6$ parent and $\lambda = 39$ offspring individuals was used⁵. As in the case of the 3D

⁵ The dimensionality of the optimization problems considered in this section is identical to the one for the static 3D cases (Section 7.2). Therefore, the same population sizes were used and are justified in Section 7.2.

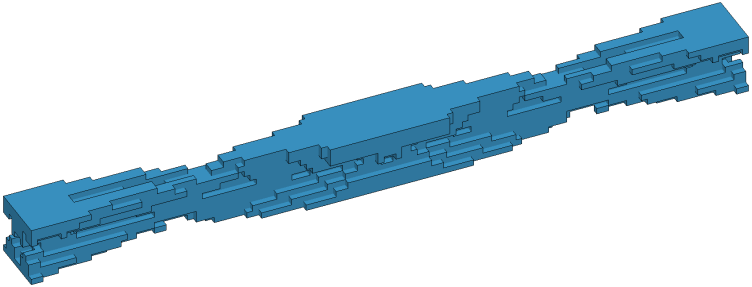
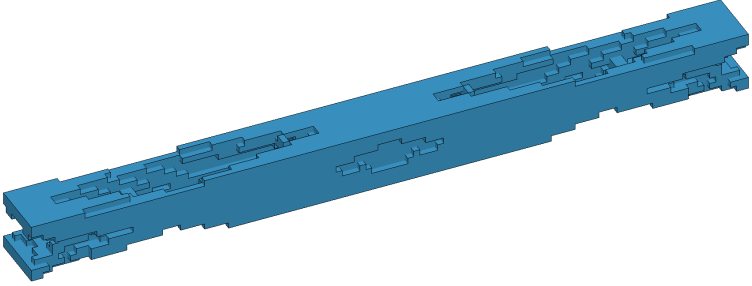
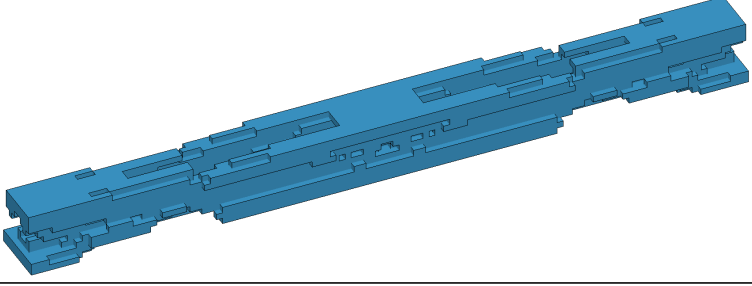
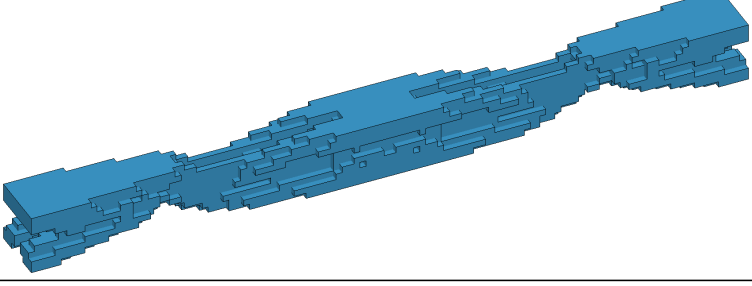
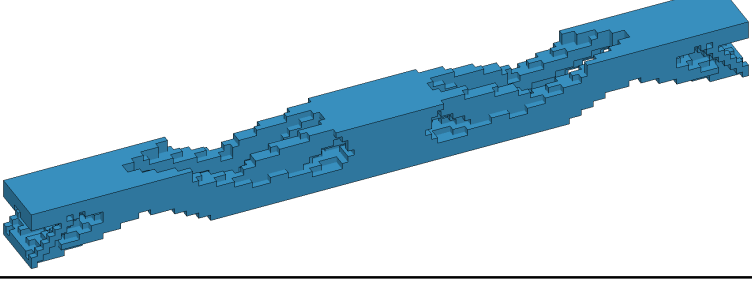
Table 8.16 Material model used in the nonlinear dynamic 3D crash test case. For simplicity, the strain rate dependency is not considered here. A more realistic hardening model than for the 2D cases, consistent with the test cases considered in the literature (Aulig et al. (2015, 2018); Ramnath et al. (2019)) is used.

Property	Symbol	Value	Unit
Material model of the structure	-	Piecewise linear plasticity	-
LS-Dyna material keyword (structure)	-	*MAT_024 (LSTC (2014))	-
Mass density	ρ_m	$2.7 \cdot 10^3$	kg/m ³
Young's modulus	E	$7.0 \cdot 10^4$	MPa
Poisson's ratio	ν	0.33	-
Yield strength	R_e	180.0	MPa

Effective plastic strain	Effective stress [MPa]
0.01	190.0
0.02	197.0
0.05	211.5
0.10	225.8
0.15	233.6
0.20	238.5
0.40	248.5

optimizations for linear elasticity, each of the runs resulted in a distinct structural concept, with intrusions ranging from 6.81 to 7.14 [mm]. The best design concept considerably differs from the ones obtained with HCA and ESL, shown in Table 8.18. The intrusions of the impactor for the designs obtained with HCA and ESL are by 3.2% and 4.3% higher than for the best design optimized with EA-LSM, respectively. This shows that EA-LSM can be successfully used for optimization of 3D crash structures and can lead to better structures than the ones obtained with state-of-the-art methods. However, as shown in Figure 8.21, the deformation of the structures at the moments of the highest intrusions among all the structures is very small. In fact, the structure reacts mostly elastically, so this load case is very similar to the static one. Surprisingly, the obtained results show that even for this case, EA-LSM outperforms considerably the ESL approach, so the dynamic effects play an important role even in the cases where very limited amount of plastic deformation

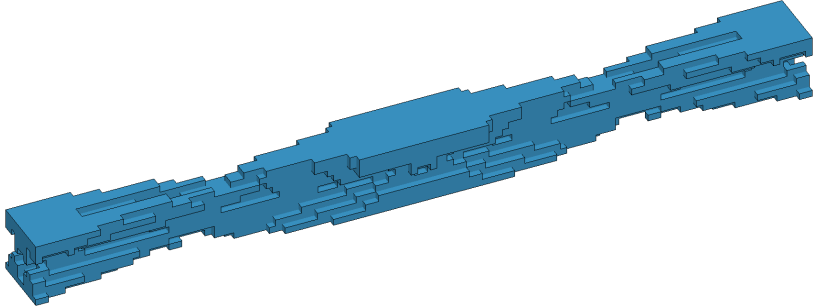
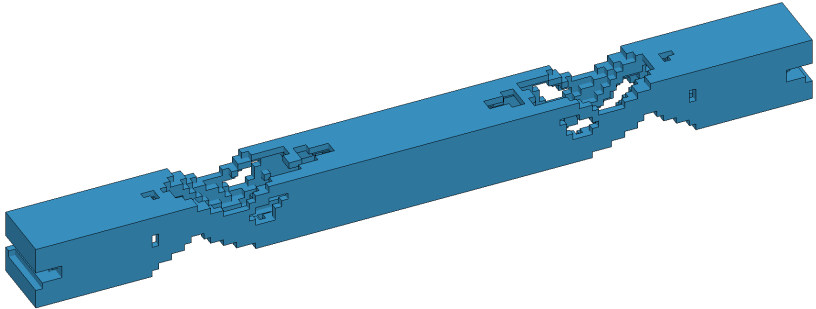
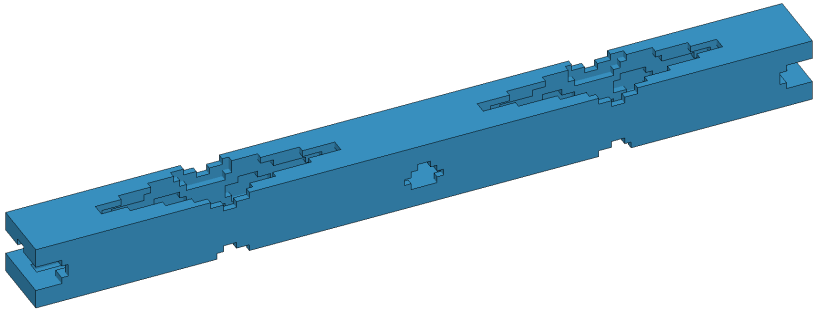
Table 8.17 Final designs in five independent EA-LSM runs with ES(6,39) for the impact energy of 409 [J].

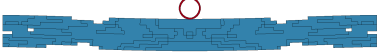
Intrusion [mm]	FE mesh
6.81	
6.94	
7.07	
7.11	
7.14	

appears. In fact, the second and the third best design for EA-LSM exhibit high similarity to the structure obtained with the ESL method, suggesting that ESL might produce a design close to a


local optimum for the considered dynamic problem.

Table 8.18 Comparison of the best EA-LSM design out of five independent optimization runs with ES(6,39) and the structures obtained with HCA and ESL for the same mass fraction. Results for the impact energy of 409 [J]. The values in parentheses express the percentage growth of intrusion w.r.t. the EA-LSM design.


Method	Intrusion [mm]	FE mesh
EA-LSM	6.81	
HCA	7.03 (+3.2%)	
ESL	7.10 (+4.3%)	



(a) EA-LSM



(b) HCA



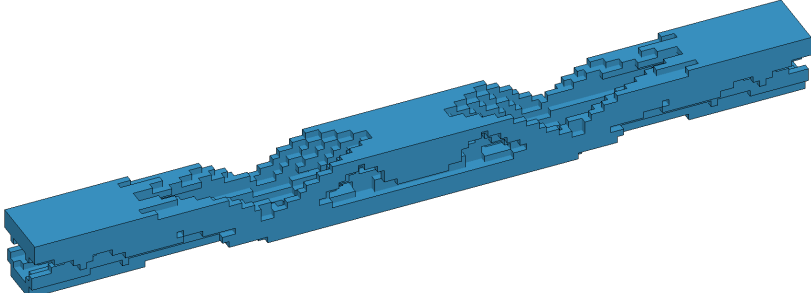
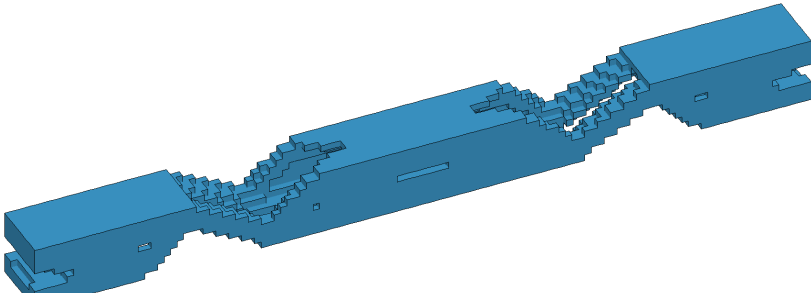
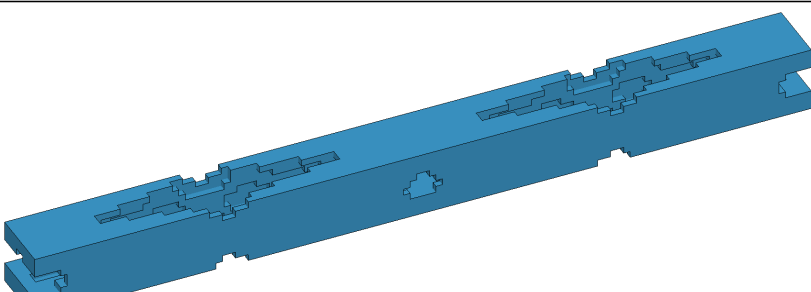
(c) ESL


Figure 8.21 Deformation of the topologies optimized using EA-LSM, HCA, and ESL, for the 3D transverse bending crash problem at the moment of the highest intrusion of the impactor. Results for the impact energy of 409 [J].

In order to evaluate the potential of EA-LSM for optimization of 3D structures with larger amounts of plastic deformation, the impact energy was increased to 2000 [J]. For this test case, again, we use the ES(6,39) in five independent optimization runs. A comparison of the best design obtained with EA-LSM with the ones generated with HCA and ESL is presented in Table 8.19. The deformed structures at the point of the highest intrusion (after ca. 3 [ms]) are presented in Figure 8.22. It turns


out that EA-LSM and HCA produce similar structures with comparable intrusions (difference less than 1%). The very good performance of the HCA algorithm might be surprising, but it has already showed very good performance on the intrusion minimization problems presented in Section 8.1. On the other hand, EA-LSM is strongly limited on the representation level and could possibly yield much better results for a higher number of MMCs. In contrast, the topology obtained with use of the ESL performs almost 6% worse than the best design obtained with EA-LSM.

Table 8.19 Comparison of the best EA-LSM design out of five independent optimization runs with ES(6,39) and the structures obtained with HCA and ESL for the same mass fraction. Results for the impact energy of 2000 [J]. The values in parentheses express the percentage growth of intrusion w.r.t. the EA-LSM design.


Method	Intrusion [mm]	FE mesh
EA-LSM	25.67	
HCA	25.46 (-0.8%)	
ESL	27.23 (+6.1%)	



(a) EA-LSM



(b) HCA



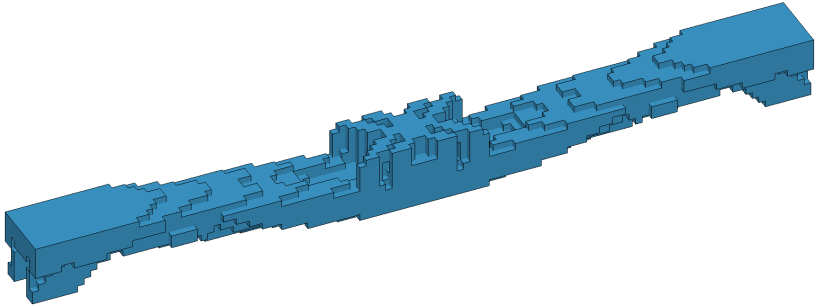
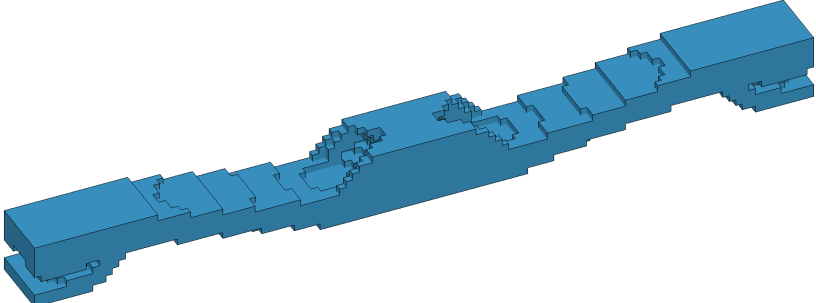
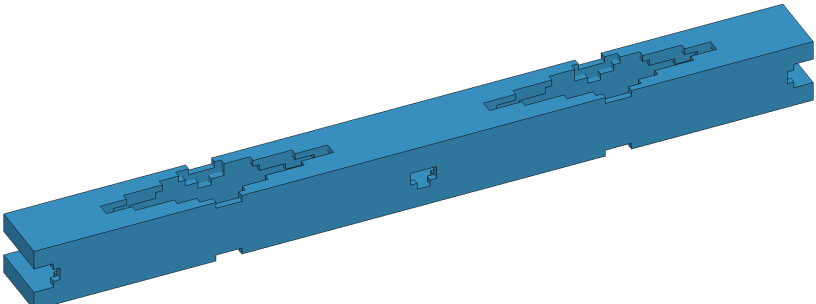
(c) ESL

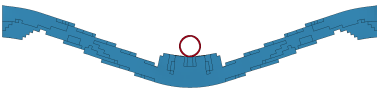
Figure 8.22 Deformation of the topologies optimized using EA-LSM, HCA, and ESL, for the 3D transverse bending crash problem at the moment of the highest intrusion of the impactor. Results for the impact energy of 2000 [J].

One should note that the optimized designs changed considerably compared to the structures ob-


tained for the lowest impact energy (Table 8.18). In, fact the last design from Table 8.17 exhibits some similarity to the best design in Table 8.19, showing a similar tendency as observed for 2D structures, where, for different impact energies, similar local optima were always found, but their relative performance was changing.

Table 8.20 Comparison of the best EA-LSM design out of five independent optimization runs with ES(6,39) and the structures obtained with HCA and ESL for the same mass fraction. Results for the impact energy of 10000 [J]. The values in parentheses express the percentage growth of intrusion w.r.t. the EA-LSM design.


Method	Intrusion [mm]	FE mesh
EA-LSM	81.7	
HCA	81.2 (-0.6%)	
ESL	94.5 (+15.6%)	



(a) EA-LSM



(b) HCA



(c) ESL

Figure 8.23 Deformation of the topologies optimized using EA-LSM, HCA, and ESL, for the 3D transverse bending crash problem at the moment of the highest intrusion of the impactor. Results for the impact energy of 10000 [J].

Finally, in order to evaluate the performance of EA-LSM on highly nonlinear 3D crash cases, the impact scenario with energy of 10000 [J] was considered. The best design obtained in five

independent optimization runs is shown and compared with HCA and ESL in Table 8.20. One can easily note a strong change in the types of designs obtained with EA-LSM and HCA. Again, for the considered objective, HCA provides very good results, showing that it is suitable for solving problems where very stiff structures are required. The difference in terms of intrusion for the structures obtained with EA-LSM and HCA stays below 1%. Most probably, the results with EA-LSM could be considerably improved if more MMCs were used. When it comes to the design obtained with the ESL approach, the intrusion is over 15% higher than for EA-LSM and HCA, showing that it is not a suitable approach for the considered case. Finally, Figure 8.23 shows the optimized structures at the moments of the highest intrusions of the impactor. In this case, the intrusion is almost two times higher than the overall height of the beam, exhibiting considerable amounts of plastic deformation.

To summarize this section, the results show that EA-LSM can be successfully used in optimization of 3D crash structures involving highly nonlinear dynamic behavior. Though the problem dimensionality for 3D TO is very high (144 design variables), the method can be still applied utilizing extensive parallelization of computations. Irrespective of the involved nonlinearities, resulting from different impact energies, the number of evaluations necessary to converge (ca. 6000) is comparable to the one for the 3D static optimization problems of the same dimensionality (Section 7.2). Finally, the comparison of the structures optimized with ESL approach and EA-LSM show that the dynamic effects are very important and the explicit FE crash simulations should be used during the optimization to obtain satisfactory results.

8.3. Summary

This chapter shows the applicability of the EA-LSM and its extensions: H-EA-LSM, KG-LSM, and A-EA-LSM, for nonlinear crash problems, based on high-fidelity explicit FE simulations. 2D and 3D versions of a transverse-bending crash problem, frequently studied in literature, are used to evaluate the properties of the proposed methods.

First of all, we show that the 2D version of EA-LSM is able to generate structures superior to the ones obtained with the state-of-the-art methods – HCA and ESL. Hence, the increased computational costs related to EA-LSM can be justified by the better performance of the resulting designs. However, in case of the considered intrusion minimization problems, the differences between EA-LSM and HCA are often very small, showing only that the inertial and nonlinear effects play an important role, and consequently, indicating the need for using nonlinear crash simulations in the optimization process. Nevertheless, in case of the mass minimization problems with intrusion and acceleration constraints, the differences between EA-LSM and both, HCA and ESL, become much more significant, showing that for such problems the heuristic assumptions used by the state-of-the-art methods are no longer applicable.

In the second part of the chapter, we show the usefulness of the ML-based enhancements of EA-LSM on the intrusion minimization problem. Firstly, we demonstrate how the approximate gradient information from physical and mathematical surrogates can increase the convergence velocity and the quality of the obtained structures in the H-EA-LSM approach. Especially in the initial phase of the optimization, H-EA-LSM can strongly benefit from the approximate gradients, reducing significantly the number of iterations required to reach similar performance as EA-LSM. When the gradient information is not accurate enough, the proposed approach can successfully limit its negative influence on the optimization process. Secondly, we evaluate the KG-LSM approach, which uses ML for modeling of the crash responses considered in the optimization. The obtained results show that the number of FE evaluations can be significantly reduced compared to EA-LSM, provided that the dimensionality of the optimization problem is kept on a low level. This result is very promising especially in the context of applications of KG-LSM to very expensive crash optimization problems. Thirdly, we investigate the applicability of the concept of optimization based on adaptive representation and learning-based topology variations using A-EA-LSM. Although the obtained topology differs from the designs optimized using EA-LSM, A-EA-LSM takes advantage of the low dimensionality of the representation and converges fast in the initial phase of the optimization.

Finally, the applicability of EA-LSM to high-dimensional optimization problems, resulting from 3D MMC parametrization, is demonstrated. For the considered intrusion minimization crash cases, with three different impact energy levels, qualitatively similar structures to the ones optimized with HCA were obtained, showing the validity of the proposed approach. However, for the lowest impact energy, EA-LSM yielded a considerably different design concept, superior to the one coming from HCA. In all of the cases, both EA-LSM and HCA were able to generate considerably better designs than ESL, with intrusion difference reaching over 15% for the highest impact energy.

All in all, we conclude that EA-LSM and its extensions are capable of solving academic nonlinear crash TO problems. Due to the generic character of the proposed methods, also other problems in structural TO could be also potentially addressed using those approaches. The increased computational costs of EA-LSM are justified by the superior performance of the obtained topologies compared to the ones generated with the state-of-the-art methods, and can be further reduced using ML-based enhancements.

Chapter 9

Industrial applications with EA-LSM

This chapter presents an application of the EA-LSM to a challenging, real-world optimization problem. Due to the rising interest of the industry in the metal Additive Manufacturing (AM) technology, a design of a hybrid S-rail concept structure, composed of an energy-absorbing, thin-walled tube and a 3D-printed metal joint, was considered as an interesting and suitable optimization scenario, exhibiting similar complexity as encountered in industrial optimization problems. The axial impact case considered here belongs to one of the most difficult problems in crashworthiness optimization, incorporating high noisiness and discontinuity of the responses, considered as objectives and constraints. Together with a very high dimensionality of the optimization problem, all of the aspects mentioned above make the considerations presented in this chapter relevant for addressing the research question of feasibility of using EA-LSM for large-scale industrial applications. All of the obtained structures are compared with the designs optimized using state-of-the-art methods for crash optimization – the ESL method and the HCA approach. The robustness of the obtained solutions w.r.t. the changing boundary conditions is studied, as well.

The chapter is structured as follows: Section 9.1 discusses concept 3D-printed crash structures proposed recently in the automotive industry, which inspired the test cases used for the optimization in this chapter. The test cases as well as the exact setup of the optimization algorithms are presented in Section 9.2. In Section 9.3, the results of the first optimization problem, aiming for minimization of mass subject to a displacement constraint of a joint, are described and analyzed. Subsequently, in Section 9.4, a scenario with modified boundary conditions is studied to evaluate the robustness of the optimized structures. Finally, Section 9.5 describes the second optimization

case, striving for a minimal mass of the joint with a constraint on the peak deceleration of the impactor. Section 9.6 summarizes the chapter.

9.1. Concept crash structures using additive manufacturing

AM, also called 3D printing, is a technology promising a lot in terms of efficient utilization of material in design engineering. In particular, due to the minimal limitations on the shapes that can be manufactured with this technology, it is seen as a great solution for the manufacturing of parts generated with use of TO. Recent developments in the field of metal 3D printing, including efficient simulation methods of the manufacturing process, allow for production of functional parts in a wide spectrum of applications, i.a. in aerospace, automotive, civil, and medical engineering (Gibson et al. (2015)). 3D printing allows for significant reduction of the number of parts compared to subtractive manufacturing, resulting in a much easier assembly process and consequently, much shorter production cycles. What is more, the possibility to manufacture parts of very high complexity opens the potential of integrating multiple functions into the 3D-printed components, resulting in innovative designs with superior properties. Although currently AM is seen rather as a technology applicable mainly in a unit production and prototyping, the manufacturing costs continue to decrease, offering a potential for using metal 3D printing in serial production. Compared to 2017, the costs of the whole manufacturing process with metal 3D printing technology are expected to decrease by 38% by 2020 and by 79% by 2026 (Canisius (2017)).



(a) 3D-printed VW Caddy front-end structure developed within the 3i-PRINT project.



(b) Hybrid space frame concept by LZN, EDAG, BLM and Concept Laser (LZN (2017)).

Figure 9.1 Concept crash-relevant vehicle structures manufactured using metal 3D printing.

Despite the uncertainty regarding the potential of applying AM in serial production, the interest of the industry in this technology, including automotive sector, tends to grow, reflecting in a continuous growth of the AM market. In particular, the issue of applying metal 3D printing to the manufacturing of the components relevant for crash performance of the car seems to be very

compelling. Interesting crash-relevant concept structures manufactured using metal 3D printing, presented recently by 3i-PRINT¹ and the Laser Zentrum Nord (LZN (2017)), are shown in Figure 9.1.

In the first case (Figure 9.1(a)), the entire front structure of the VW Caddy was 3D-printed in separate parts and joined together with 59 welding seams. An industrial metal 3D printing system EOS M 400 was used for manufacturing of the individual parts. The entire structure was fabricated using the Scalmalloy[®] aluminum alloy, known for its high strength, fracture-resistance, and very good weldability. The frame was designed based on the results from TO taking into account several real-world load cases. The structure fulfills current requirements regarding crash and vehicle safety as well as thermal management functions. The total weight of the complete frame concept before connecting the individual parts was reduced to 34 [kg], which shows a great potential for using 3D printing in manufacturing of vehicle components.

In the second vehicle concept structure (Figure 9.1(b)), AM was used to fabricate metal joints connecting conventional thin-walled profiles. This is an interesting solution, since it allows for exploitation of the energy-absorbing properties of the thin-walled extrusions, while taking advantage of the high stiffness of the 3D-printed joints. Joints are known to have very big influence on the overall stiffness of the body-in-white as well as the energy absorption potential. Moreover, such solution could be more easily integrated into the production, allowing for a gradual introduction of metal 3D printing on an industrial scale. The concept structure is claimed to have 20% lower weight than a functionally-equivalent structure used currently in the automotive industry.

9.2. Test case and optimization problems

Taking inspiration from the hybrid frame concept design described above, we prepared a simplified model to evaluate the feasibility of using EA-LSM in optimization of industrial problems. The FE model used in this work is presented in Figure 9.2. It is composed of a thin-walled tube with thickness of 1.5 [mm] (brown) and a solid joint composed of a design domain (red) and an interface (green). The joint is modeled using solid finite elements and the thin-walled structure using shell finite elements. For simplicity, the interface nodes between tube and the joint are merged, meaning that the translational degrees of freedom are transferred between solid and shell elements. The detailed description of the numerical model can be found in Table 9.1. To enhance the initiation of the crash at the front part of the structure, triggers, as shown in Figure 9.2, are added. Both parts of the hybrid frame are modeled as aluminum with piecewise linear hardening characteristics defined in Table 9.2.

Please note that, to fully account for the metal AM process, it would be needed to consider modified, more complex material models, as well as special techniques to integrate the manufactur-

¹ Partnership between Altair, APWORKS, csi entwicklungstechnik, EOS, GERG, and Heraeus (www.3i-print.com).

ing constraints into the optimization. For instance, one of the commonly addressed problems in structural TO is generation of self-supporting designs, which do not require complex supporting structures and their costly removal (Brackett et al. (2011); Leary et al. (2014); Gaynor and Guest (2016)). Such an approach, for 2D structures, was integrated into EA-LSM in our past work (Bujny et al. (2017c)), and extended later to the 3D domain in the master's thesis of Mulard (2018), co-supervised by the author of this dissertation. However, since neither the manufacturing constraints nor the detailed material modeling for AM is the primary focus of this thesis, those problems are not addressed here and pose an interesting direction for the future research.

In the considered crash scenario, a rigid wall of mass of 500 [kg] is impacting a hybrid structure with an initial velocity of 5 [m/s]. All degrees of freedom of the surface nodes located at the lower half of the joint are fixed. In all of the considered optimization cases, an additional constraint ensures the absorption of the full energy of the impactor by requiring the rigid wall to move in the opposite direction at the end of the simulation.

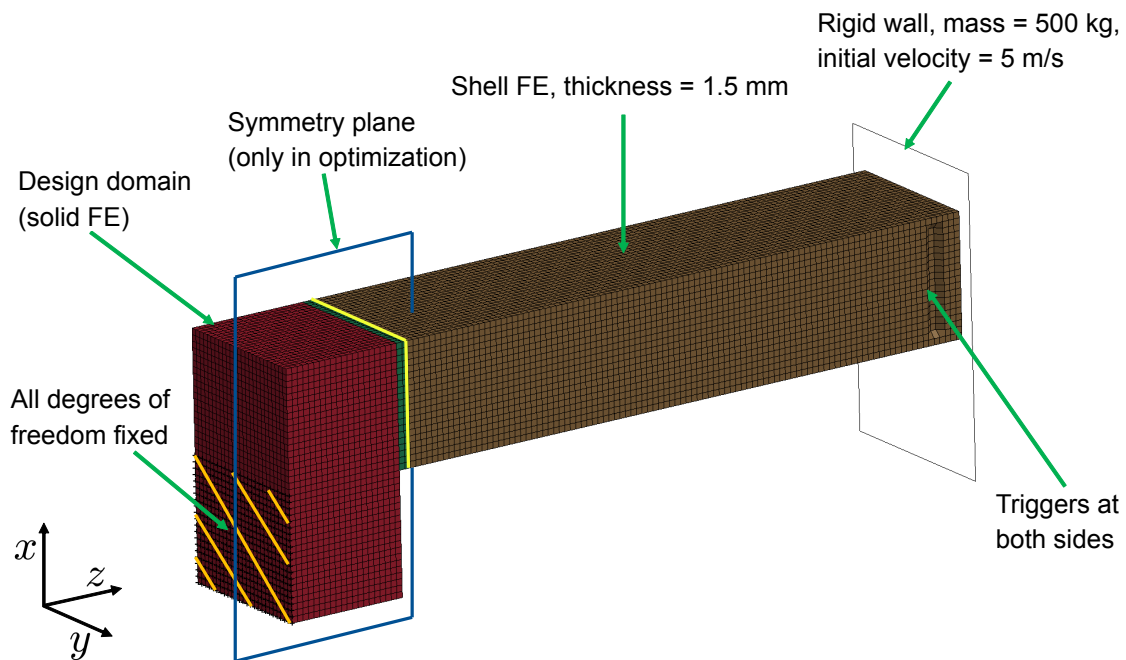


Figure 9.2 The hybrid S-rail FE model. A rigid wall impacts the structure composed of a thin-walled tube and a 3D-printed solid joint. The lower part of the joint is clamped (orange hatching). The nodes marked with yellow lines are used for the measurement of the displacement of the joint.

The axial impact case considered here belongs to one of the most challenging and important optimization scenarios in vehicle crashworthiness and is widely studied in the literature (Duddeck et al. (2016); Liu (2008); Sun et al. (2014); Yin et al. (2014); Tang et al. (2013)). However, unlike in most of the works, instead of optimizing the structure of the thin-walled tube, we focus on optimization of the 3D-printed joint supporting it w.r.t. the performance metrics of the overall system. This makes the problem potentially even more difficult for the state-of-the-art methods, due to the indirect relationship between the design variables, which influence only the structure of the joint,

Table 9.1 Configuration of the hybrid S-rail FE model.

Property	Symbol	Value	Unit
Material model of the structure	-	Piecewise linear plasticity	-
LS-Dyna material keyword (structure)	-	*MAT_024 (LSTC (2014))	-
Mass density	ρ_m	$2.7 \cdot 10^3$	kg/m ³
Young's modulus	E	$7.0 \cdot 10^4$	MPa
Poisson's ratio	ν	0.33	-
Yield strength	R_e	180.0	MPa
Joint dimensions	-	160 x 100 x 80	mm
Joint mesh resolution	-	40 x 25 x 20	-
Simulation time	t_{end}	120	ms
Number of shell finite elements	-	9000	-
Number of solid finite elements	-	21000	-

Table 9.2 Piecewise linear isotropic hardening defined by the effective plastic strain and the corresponding effective stress. Employed material model is not taking into account strain-rate-dependent behavior.

Effective plastic strain	Effective stress [MPa]
0.01	190.0
0.02	197.0
0.05	211.5
0.10	225.8
0.15	233.6
0.20	238.5
0.40	248.5

and the mechanical behavior of the entire system. From the perspective of an evolutionary opti-

mizer, this problem does not make any difference, provided that the modifications of the joint can lead to an improvement of the considered objective.

In order to fully evaluate the potential benefits of using EA-LSM in an industrial setting, two different optimization problems are considered. In the first case, the maximal displacement of the joint, in the longitudinal direction (z-direction), at any nodes marked with yellow color in Figure 9.2 is constrained to a value below 2 [mm]. Additionally, as mentioned above, a constraint requiring the impactor to bounce back at the end of the crash simulation is introduced. In such a way, it is guaranteed that the simulation takes into account all of the dynamic loads during the crash event. The mass of the joint is subject to minimization. The optimization problem can be defined formally as follows:

$$\begin{aligned}
 & \min_{\mathbf{z}} (m(\mathbf{z})), \mathbf{z} \in \mathbb{R}^n; \\
 & \text{s.t. } \mathbf{res}(t) = \mathbf{0}; \\
 & \quad d_{max}(\mathbf{z}) \leq d_{req}; \\
 & \quad v_{end}(\mathbf{z}) \leq 0,
 \end{aligned} \tag{9.1}$$

where m is the mass of the structure, depending on the vector of design variables \mathbf{z} . The condition $\mathbf{res}(t) = \mathbf{0}$ corresponds to the dynamic equilibrium at time t . d_{max} is the maximum displacement in longitudinal direction of any of the measurement nodes marked in Figure 9.2 and d_{req} represents its maximum allowed value (2 [mm]). Finally, v_{end} denotes the longitudinal component of velocity of the rigid wall at the end of the crash.

The formulation (9.1) is probably the most intuitive definition of an optimization problem for the considered case. The main function of the joint is to stabilize and support the thin-walled structure, which absorbs the energy during the crash event. Therefore, the target is to keep the maximal displacement of the joint under a certain limit, while minimizing the mass to achieve a lightweight design. Intuitively, this formulation should lead to a very stiff joint design, showing very small plastic deformation. For that reason, also state-of-the-art methods suitable for that problem, such as HCA or ESL, should come up with relatively well-performing designs.

In the second scenario, the objective is again to minimize the mass of the structure, but instead of constraining the displacement of the nodes on the joint, we restrict the magnitude of acceleration, which is relevant from the perspective of injury-related criteria, such as HIC. By the second Newton's law of motion, this is equivalent to restricting the maximal force acting on the wall. Both of these criteria are widely studied in the literature (Fang et al. (2016)). The formal definition of the

optimization problem takes the following form:

$$\begin{aligned}
 & \min_{\mathbf{z}} (m(\mathbf{z})), \mathbf{z} \in \mathbb{R}^n; \\
 & \text{s.t. } \mathbf{res}(t) = \mathbf{0}; \\
 & \quad a_{max}(\mathbf{z}) \leq a_{req}; \\
 & \quad v_{end}(\mathbf{z}) \leq 0,
 \end{aligned} \tag{9.2}$$

where a_{max} is the maximal magnitude of acceleration in the longitudinal direction.

The second formulation of the optimization problem gives the joint the possibility not only to support and stabilize the thin-walled part to absorb the kinetic energy of the impactor, but also to enhance the overall crash behavior of the hybrid structure by limiting the peak force. In such a way the optimization problem is formulated to achieve the relevant performance goals directly, instead of considering a surrogate problem for the TO of the joint only (e.g. stiffness maximization). One of the main advantages of EA-LSM over the established methods is the fact that the optimization problem presented above can be considered in a direct way, which does not make the problem definition more difficult than the previous one. More precisely, the performance of the designs considered during the optimization are evaluated exclusively based on the specified objective and constraints, which can be arbitrary criteria, as long as they can be quantified. Therefore, instead of considering objective functions such as displacements or specific energy absorption for particular components, one can target optimization of relevant injury criteria for a bigger subsystem, which determine at the end the safety of a vehicle.

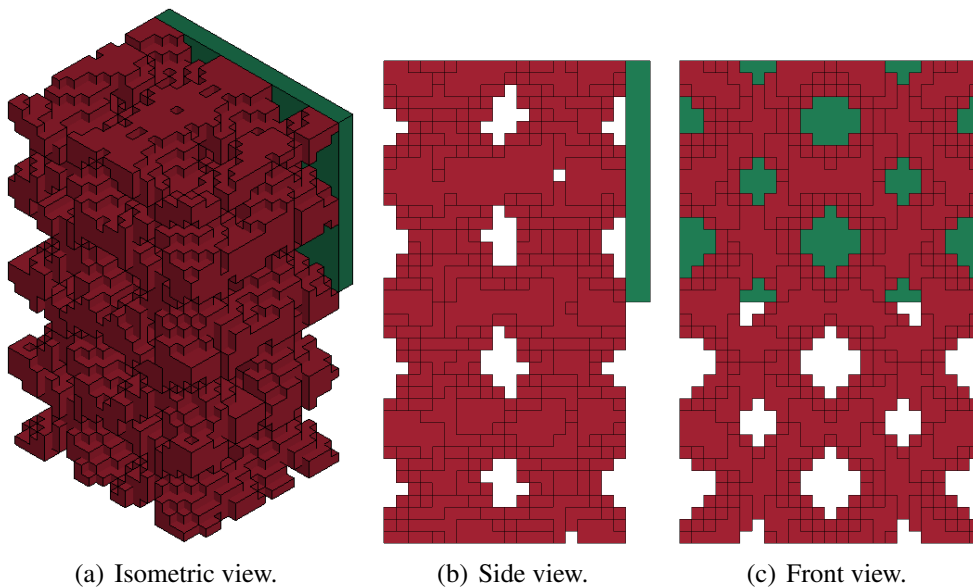
Both of the test cases described above are optimized using the standard version of the EA-LSM, as described in Chapter 5. The exact setup of the optimization algorithm is given in Table 9.3. In both considered scenarios the objectives and constraints are normalized with the values obtained in a simulation of a hybrid S-rail structure with a joint entirely filled with material.

The initial design used in all of the optimization runs with EA-LSM is shown in Figure 9.3. The presented structure results from a diagonal distribution of MMCs, similar to the ones shown in Chapter 7 and 8.

All of the crash simulations are performed using explicit MPP LS-DYNA solver, version R7.1.1, double precision. The evaluations in each generation of EA-LSM are realized in parallel, using a computational cluster and a parallelization with MPI for Python (Dalcín et al. (2005, 2008, 2011)). As a consequence, the duration of the evaluation of the entire offspring generation is practically equal to the time of a single FE simulation. Due to this fact, the choice of the optimization algorithm and the population size is not dictated by the goal of minimizing the number of FE evaluations, but by the total run time. This is the main motivation for choosing a relatively big population size and a standard ES(μ, λ) instead of the CMA-ES, which is more efficient for smaller population sizes. The ratio between the number of parent and offspring individuals is kept slightly lower

Table 9.3 Setup of EA-LSM for the hybrid S-rail optimization.

Property	Symbol	Value
Algorithm type	-	Evolution Strategy (ES)
Parent population size	μ	10
Offspring population size	λ	79
Number of step sizes	n_{σ}	1
Initial step size length	σ_{init}	0.02
Number of MMCs	-	64
Number of design variables	-	288
Initial thickness of MMCs	-	10.25 [mm]
Thickness threshold for deletion of MMCs	t_{min}	4.0 [mm]
Volume fraction for deletion of overlapping MMCs	-	0.9

**Figure 9.3** Initial design of the 3D joint for the optimizations using EA-LSM. The structure is generated according to the procedure described in Chapter 5.

than the recommended $1/7$ level (Schwefel (1987)) to increase the convergence velocity. In order to focus on the fast convergence rather than on global search reliability, the version of ES with a single standard deviation (Section 3.2) is chosen, which minimizes the learning effort for the opti-

mization algorithm. The optimization runs are continued until convergence, which is evaluated by visual inspection of the cost function.

Similarly to the approach from Chapter 8, HCA and global ESL methods are used for comparison purposes. In case of the HCA method, exactly the same FE model and setup as in case of the EA-LSM is used (Figure 9.2). The structure of the joint is optimized with use of LS-TaSC 3.2 with the default settings. The mass constraint is set to meet the mass of the structure obtained with EA-LSM. Since the parametrization in HCA introduces intermediate densities, the resulting topology is post-processed to obtain a 0-1 material distribution and meet the mass constraint (for details see Chapter 8). After the post-processing, the performance of the structure is validated in a crash scenario.

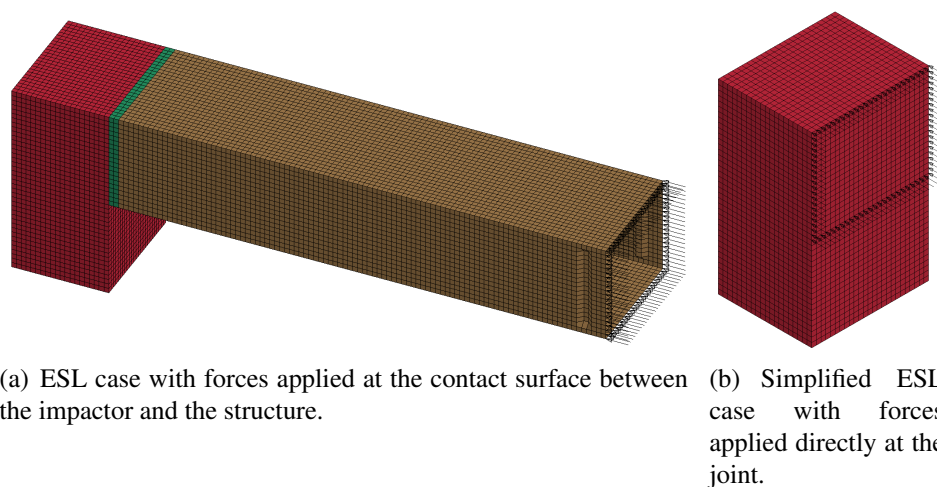


Figure 9.4 The equivalent static cases of the crash scenario. Though more appropriate, the case presented in Figure 9.4(a) causes stability problems for the SIMP algorithm. Therefore, a simplified equivalent static case, presented in Figure 9.4(b), is used.

The ESL approach is based on the load case depicted in Figure 9.4(b). Unlike in the cases presented in the previous chapters, the static forces are not applied at the contact surface between the impactor and the S-rail, but applied directly on the design space for the joint, as shown in Figure 9.4(b). The main reason for this simplification is the fact that the original equivalent static case (Figure 9.4(a)) leads to a TO problem with design-dependent loads. Such problems cannot be optimized with the standard SIMP algorithm (Sigmund (2014)) and need to be addressed with specialized methods (Bendsøe and Sigmund (2004); Chen and Kikuchi (2001); Du and Olhoff (2004)). Similarly to the cases described in Chapter 8, the material of the joint is changed to linear elastic to carry out the optimization of the topology. Since the algorithm used here has also difficulties with providing stable solutions for very low volume fractions, in both of the test cases presented in this section, the topology of the joint is optimized first for a 15% volume fraction. In the second step, the elements with intermediate densities are either deleted or assigned full density to meet the required mass constraint. Finally, the crash performance of the hybrid structure with the topologically-optimized joint is evaluated based on the same setup as in case of the EA-LSM.

9.3. Mass minimization under a node displacement constraint

In the following section, the optimization results for the first problem described in Section 9.2, are presented. First of all, the topology of the joint is optimized with use of the standard EA-LSM algorithm, as described in Chapter 5. The evolution of the cost function during the optimization process is shown in Figure 9.5. The corresponding plots of the mass fraction of the joint and the normalized constraints, taken into account in the optimization problem, are presented in Figure 9.6.

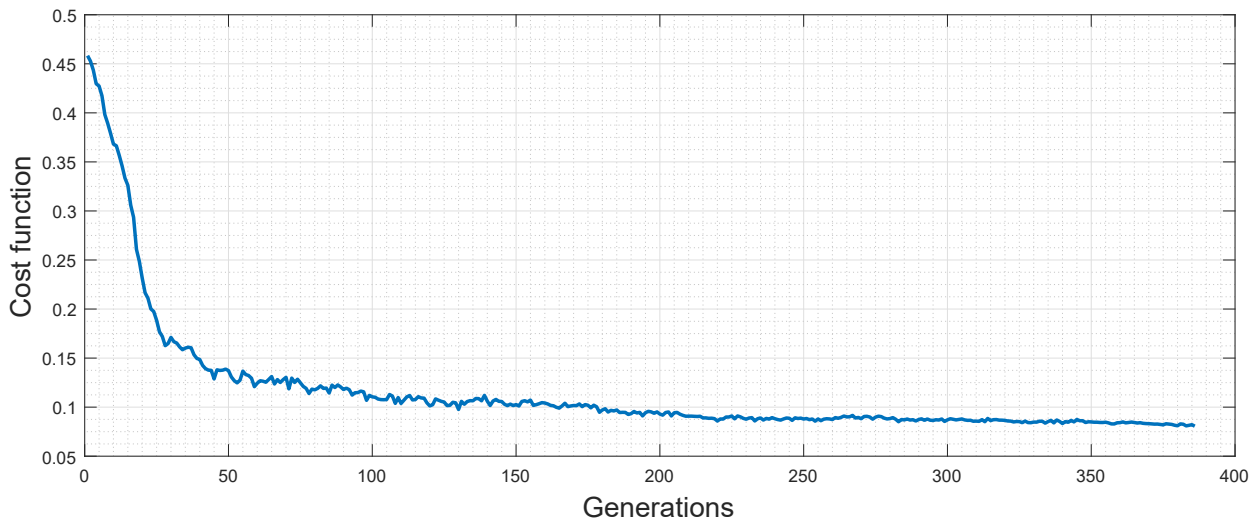


Figure 9.5 Plot of the cost function for the 3D joint mass minimization problem with a constraint on the maximal displacement of the joint, addressed using EA-LSM.

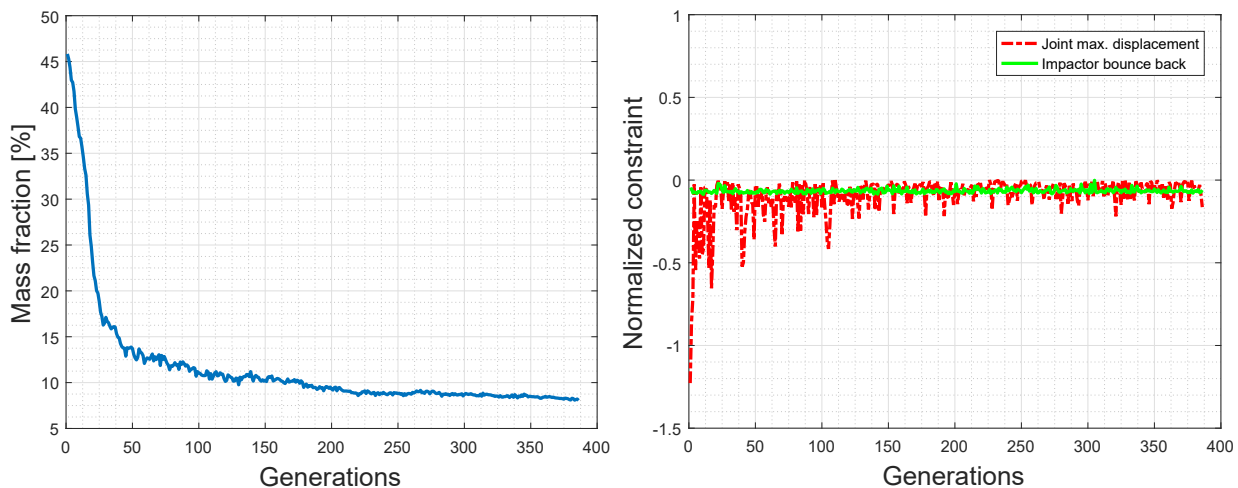


Figure 9.6 Plots of the mass fraction (left) and the normalized constraints (right) for the 3D joint mass minimization problem with a displacement constraint, addressed using EA-LSM. The constraint on the maximal displacement of the joint in the longitudinal direction (red dashed line) is calculated as $d_{max} - d_{req}$, where d_{max} is the maximal displacement in the best design in a given generation and $d_{req} = 2$ [mm] is the maximal allowed displacement. The second constraint (green solid line), requiring the impactor to bounce back at the end of the simulation, or equivalently, to absorb the whole kinetic energy of the impactor, is computed as $\frac{v_{end} - 0}{v_{init}}$, where v_{end} is the longitudinal component of velocity of the impactor (initially positive) at the end of the simulation and $v_{init} = 5$ [m/s] is the initial velocity used to normalize the constraint. A constraint is violated if it is larger than 0.

It should be noted that the initial design was already feasible, therefore the plot of the cost function in this case is identical to the one of the mass fraction. Initially, the design satisfies the displacement constraint with a large margin (the maximal displacement in the longitudinal direction is over 1.2 [mm] lower than the maximal allowed value). During the optimization, the mass of the joint is reduced from almost 46% to only 8.21% of the mass of the full design domain. As a result, the displacement constraint becomes active, while the second constraint, requiring the impactor to bounce back at the end of the simulation, seems to be unaffected, with the final value of -0.065 , corresponding to the velocity of 0.33 [m/s] in the direction opposite to the initial one. This is understandable, since the first constraint already requires the joint to support and stabilize the thin-walled structure, allowing only for small deformations of the joint. Once the displacement constraint is satisfied, most of the times the structure is stable enough to let the thin-walled structure develop a desired folding pattern, leading to the absorption of the entire kinetic energy within the time of the simulation (120 [ms]).

The final design obtained with EA-LSM as well as two other designs, generated using ESL and HCA, are presented in Table 9.4. Table 9.5 presents the crash behavior of the considered structures, while Figure 9.7 shows the displacement history of the node with the highest displacement (in the longitudinal direction) during the crash event. Finally, the performance of the three designs is compared in Table 9.6. Additionally, the effective plastic strain of the joint structures in different phases of the crash event is presented in Figure 9.8 and the evolution of the velocity and displacement of the rigid wall over time is depicted in Figure 9.9.

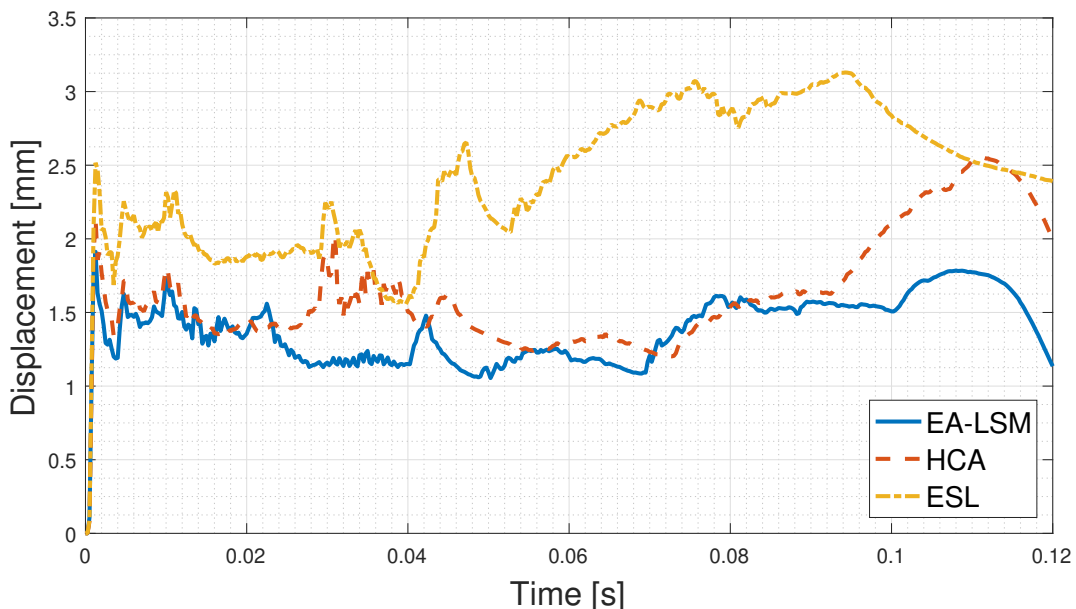


Figure 9.7 Displacement history (in the longitudinal direction) of the nodes on the joint with the highest displacements for the designs optimized with EA-LSM, HCA, and ESL. 3D joint mass minimization problem with the displacement constraint.

One can easily note considerable differences between the design concepts obtained with all the three methods. The structures optimized with use of ESL and HCA support mainly the corners

Table 9.4 Comparison of the joint structures obtained using EA-LSM, ESL, and HCA for the 3D joint mass minimization problem with a displacement constraint.

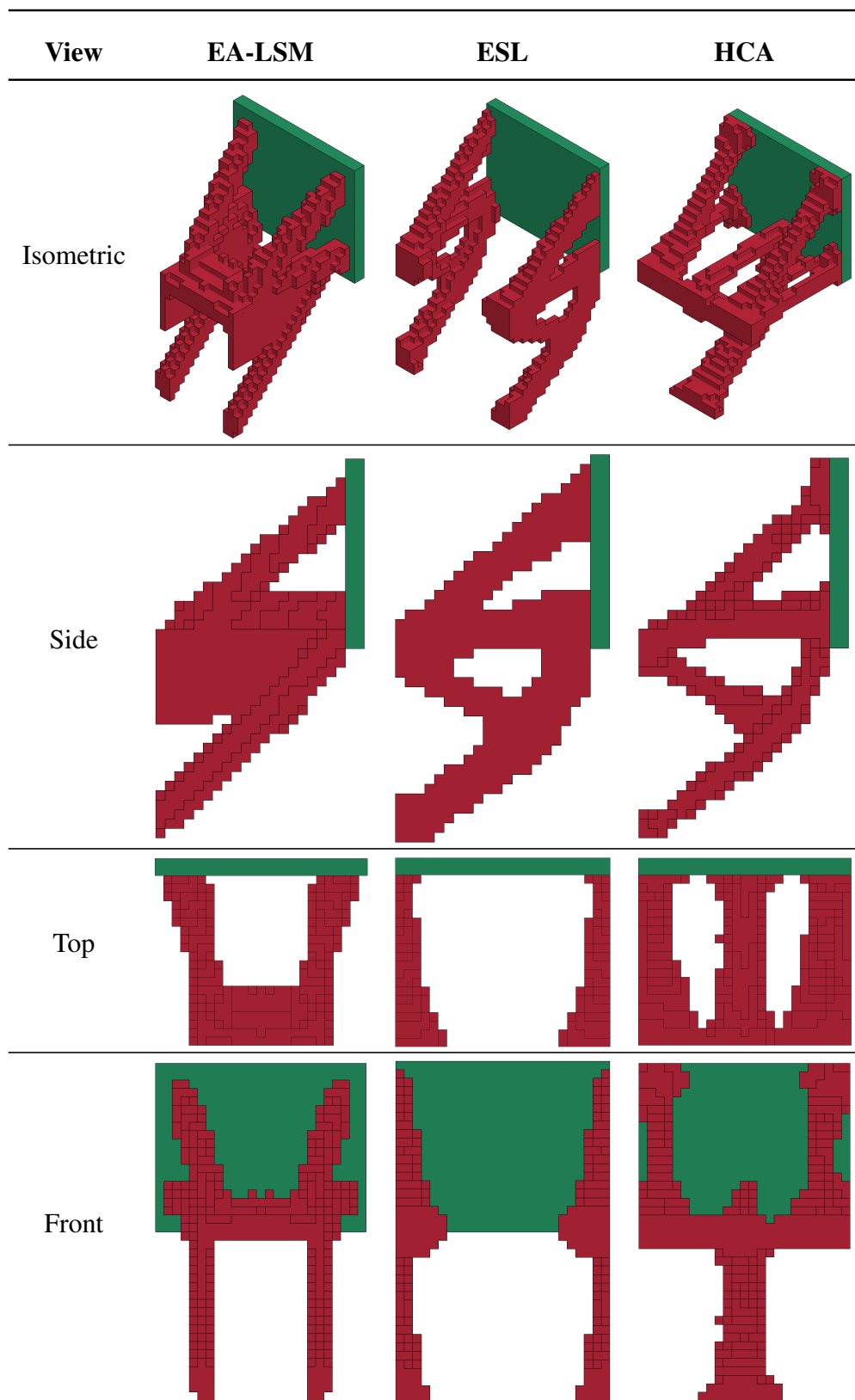
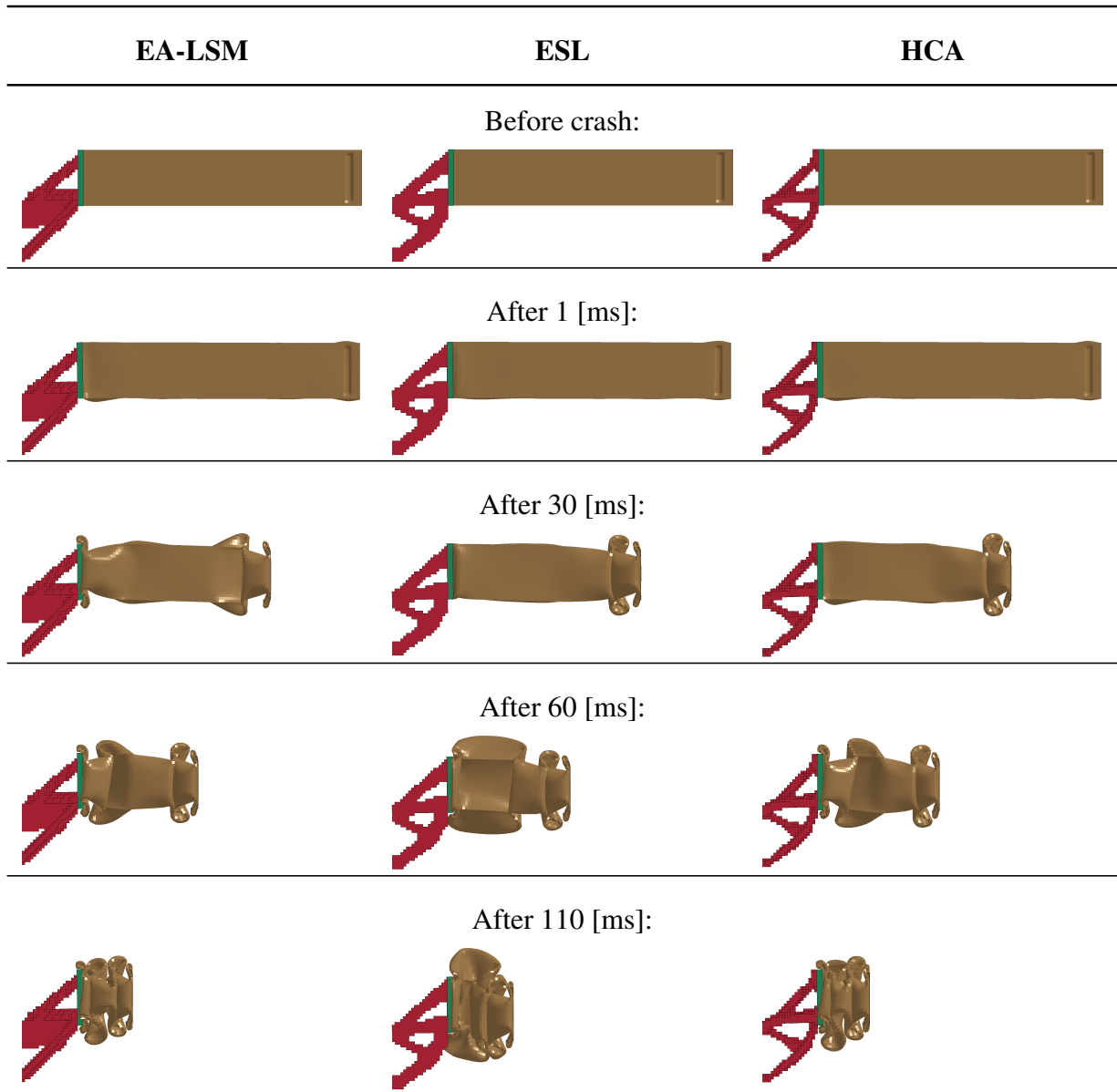


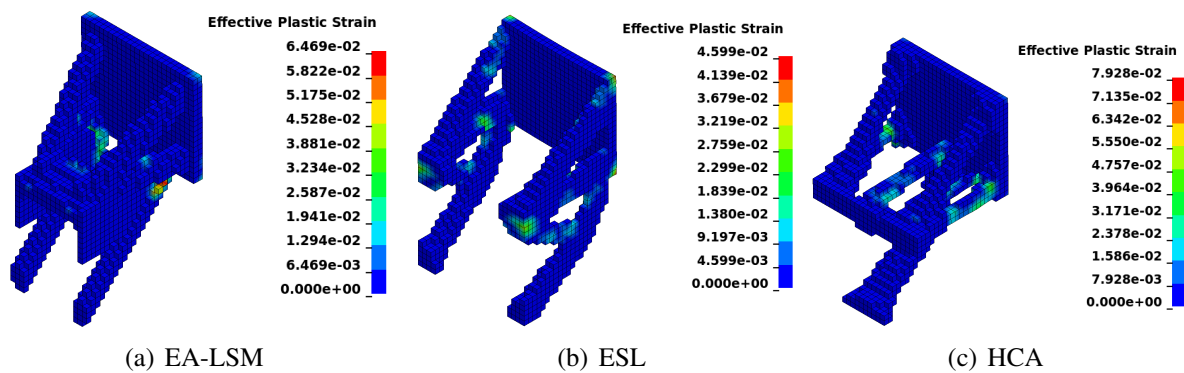
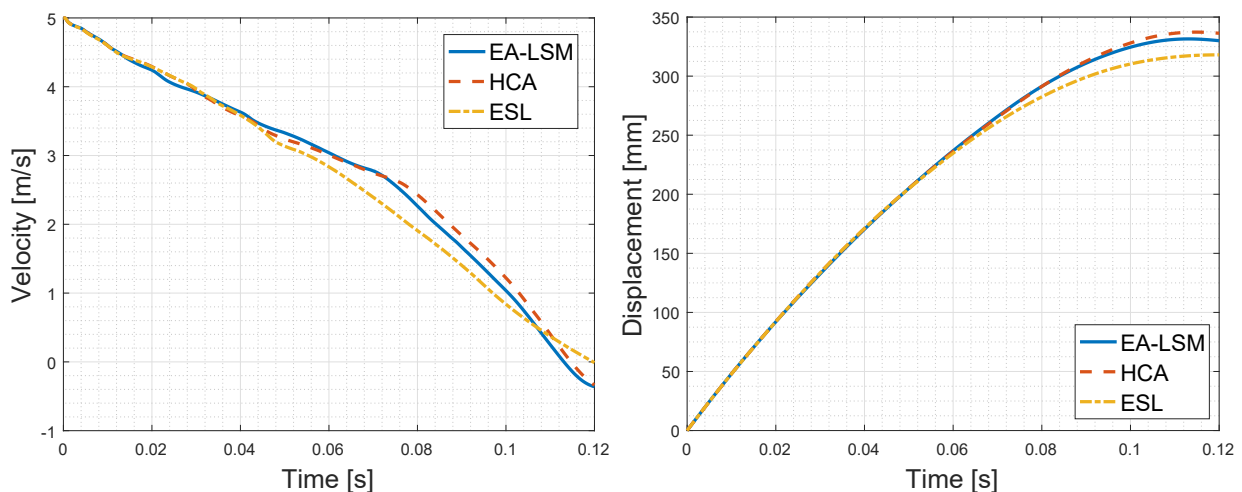
Table 9.5 Comparison of the crash behavior of the hybrid S-rails with joints optimized using EA-LSM, ESL, and HCA for the mass minimization problem with a displacement constraint. Side view.



of the interface plate, while the EA-LSM develops a design supporting more the center of the plate. For ESL, this is an understandable result, due to the simplification of the loading conditions, assumed to be constant over time and applied directly at the boundary of the design domain. In contrast, EA-LSM takes into account the dynamic effects as well as changing loads during the crash event, which leads to a structure supporting also the middle of the plate. The main benefit of this solution is a better support of the thin-walled tube in the second phase of the crash, when it buckles close to the joint and comes into contact also in the center of the plate. The changing loading conditions of the joint are reflected in the plots of the von Mises stress at different moments of the crash simulation (Figure 9.10). Initially, the long, diagonal beams are heavily loaded, while at the end, the central part of the joint is strongly compressed and supports the thin-walled

Table 9.6 Comparison of performance of the structures obtained using EA-LSM, ESL, and HCA for the 3D joint mass minimization problem with a displacement constraint (2 [mm]).

Property	EA-LSM	ESL	HCA
Volume fraction	8.21%	8.24%	8.13%
Max. z-displacement	1.91 [mm]	3.13 [mm]	2.55 [mm]
Displacement increase (w.r.t. EA-LSM)	0%	+64%	+34%

**Figure 9.8** Effective plastic strain at the end of the simulation (after 120 [ms]) for the designs optimized with EA-LSM, HCA, and ESL. 3D joint mass minimization problem with a displacement constraint. The scales in each of the plots are chosen independently to show clearly the variations in the distributions of the depicted fields.**Figure 9.9** Velocity and displacement (in the longitudinal direction) of the rigid wall during the crash event for the designs optimized with EA-LSM, HCA, and ESL. 3D joint mass minimization problem with a displacement constraint.

structure. As a result, the tube is well-stabilized during the whole crash event, keeping the maximal displacement of the joint under 1.91 [mm]. In contrast, the joints obtained using the ESL or HCA method deform strongly after ca. 100 [ms] of the crash (Figure 9.7), when the loading

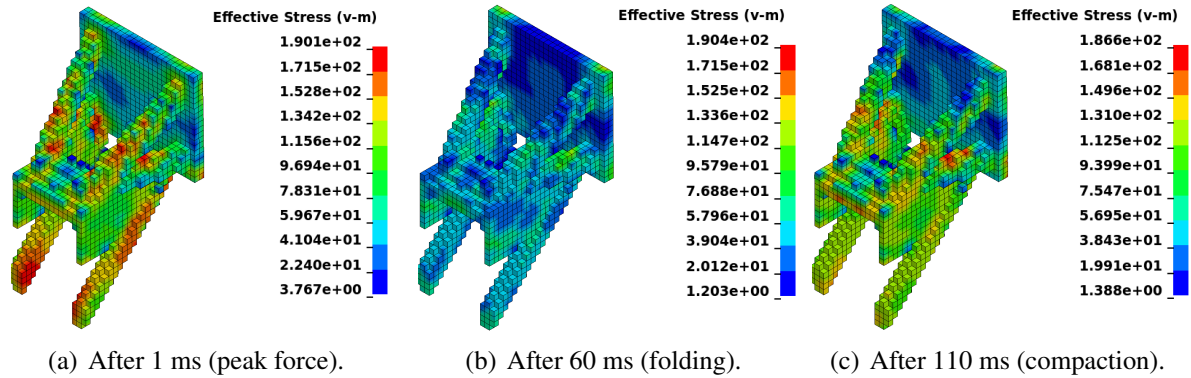


Figure 9.10 Equivalent stress (von Mises) [MPa] and deformation of the joint optimized with EA-LSM in different phases of the crash event. The scales in each of the plots are chosen independently to show clearly the variations in the distributions of the depicted fields. The phases are characterized by the state of the thin-walled tube, specified in the parentheses in the captions. After strong tension and plastic deformation of the lower members during the first contact of the structure with the impactor, the joint operates mostly in the elastic range. 3D joint mass minimization problem with a displacement constraint.

conditions change substantially. All in all, the designs obtained using ESL and HCA result in 64% and 34% higher maximal displacements compared to the EA-LSM design, respectively. This is a considerable difference, showing that EA-LSM can lead to clearly superior designs compared to the state-of-the-art methods even in case of scenarios with relatively moderate plastic deformations of the part subject to optimization. This is quite surprising, since for the considered optimization problem, maximization of the stiffness of the joint seems to be a reasonable simplification, which should have a minor influence on the final result. Nevertheless, this demonstrates the importance of the dynamic effects in the crashworthiness optimization. This may become even more relevant in case of considering strain rate dependency in the material model.

To summarize, EA-LSM comes up with significantly better design than the ESL and HCA methods. Although the overall computational cost of EA-LSM is considerably higher than for the state-of-the-art methods², the individuals in each generation can be evaluated in parallel and the method offers very fast convergence at the beginning (over 70% reduction of mass within the first 50 generations), which is crucial in real-world applications. Furthermore, EA-LSM shows its capability of dealing with more than one constraint, which is a big problem for approaches like HCA. Finally, the optimized design provides a stable support for the thin-walled part during the entire crash event, with the highest displacements of the joint appearing at the beginning of the crash simulation.

In the following section, we analyze the robustness of the generated designs w.r.t. variations of the load cases. Afterwards, in Section 9.5, the results for the second optimization problem, targeting mass minimization under an acceleration constraint, will be discussed.

² For ESL with SIMP, ca. 50 linear elastic static FE simulations were necessary for the optimization to converge. HCA required 30 nonlinear explicit crash simulations.

9.4. Robustness

One of the most important aspects in the design of crashworthy structures is their sensitivity w.r.t. changing boundary conditions. In most real-world engineering problems, in particular involving vehicle safety, there is a substantial degree of uncertainty, ranging from loading conditions to manufacturing imperfections. For this reason, Robust Design Optimization (RDO) plays a more and more important role in engineering applications, with specialized methods allowing for finding designs whose performance does not strongly depend on the variations of the input parameters. An overview of the RDO approaches used in crashworthiness optimization has been given by Aspenberg (2011). Although these methods could be potentially integrated into the EA-LSM, the computational costs would be high. On the other hand, evolutionary optimization itself can be considered a RDO technique. The mutation operator used in ESs can be seen as a robustness tester (Beyer and Sendhoff (2007)), leading to similar variations as the artificial noise in the standard RDO approaches. Moreover, as discussed by Branke (1998), standard evolutionary optimizers would usually focus on the hills of the fitness landscape rather than on sharp peaks since the probability for an individual from the initial population to be located in the basin of attraction of the hill is considerably higher. This concept is illustrated in Figure 9.11, depicting the test problem used by Branke (1998) to validate this hypothesis. In 99% of cases, the EA would converge to the solution A and only 1% of times to the solution B. The above mentioned properties of the evolutionary search could explain why the species developed within the natural evolution process appear to be optimal on average and are not concentrated on a particular point of the "design space".

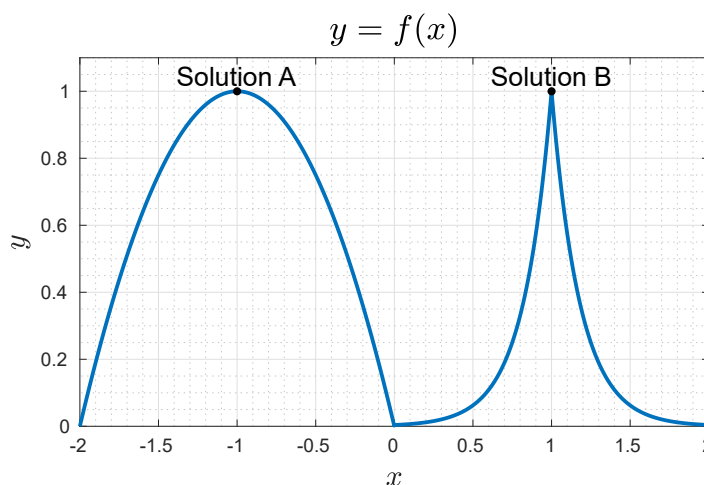


Figure 9.11 Test function with two equally high maxima. A standard EA will turn to the smooth hill (solution A) rather than to the peak (solution B) in 99% of optimization runs according to Branke (1998).

To demonstrate the ability of EA-LSM to develop robust solutions, in this section, the sensitivity of the responses of the previously optimized structures w.r.t. small variations of the loading conditions is tested. The evaluation is based on a modified crash scenario, where the impacting rigid wall is

inclined relative to the initial orientation as shown in Figure 9.12. At first, the crash behavior of the joints from Section 9.3 in a simulation with a rigid wall inclined by 5 [deg], which is the highest inclination considered in this section, is compared. Secondly, to obtain statistically significant results, 100 samples of inclination angles varying between -5 to $+5$ are generated using Monte Carlo sampling according to a uniform random distribution. For each sample, an FE simulation is run for the joints optimized using EA-LSM, ESL, and HCA and the performance of the structures is compared. Finally, conclusions about the robustness of the structures obtained using different methods are derived.

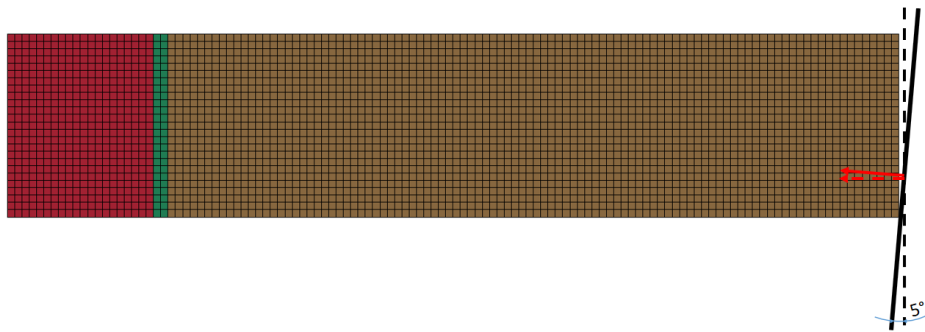


Figure 9.12 Modified crash test case of the hybrid S-rail for evaluation of robustness of the previously optimized designs. The impacting rigid wall is rotated relative to the initial position by 5 [deg] and the impact direction is perpendicular to the rotated wall. Besides, the FE model remains unchanged (see Figure 9.2). Top view.

First of all, we start with a study of crash behavior of the hybrid structures from Section 9.3 in the limit case with inclination angle of 5 [deg]. Table 9.7 presents the evolution of the deformation of the structures at different points in time. Until 60 [ms] of the simulation, all of the structures deform in a similar way. After that, clear differences develop, which is reflected also in the plots of maximal displacements of the nodes at the joint (Figure 9.13) as well as in the plots of velocity and displacement of the rigid walls (Figure 9.14). The joint obtained using EA-LSM remains superior compared to the structures generated with ESL and HCA, which show over 208% and 34% higher displacements, respectively (Table 9.8). Especially in case of the design obtained using ESL, the lack of the structure in the middle of the design space, supporting the thin-walled tube in the second phase of the crash (when it comes in contact with the interface plate) has severe consequences. This results in over 259% increase of the maximal displacement of the joint compared to the previous crash scenario. It seems that in this case, at least two replacement load cases, representing two different types of loading conditions occurring during the crash event should be taken into account to obtain satisfactory results using ESL approach. Surprisingly, HCA leads to a well-behaving structure, with only 36% increase of the maximal displacement, which is comparable to the EA-LSM design with 38% higher maximal displacement than in the case without wall inclination.

Although the EA-LSM solution results in the lowest displacements of the joint, it actually achieves that by allowing for the largest displacement of the rigid wall (Figure 9.14), while limiting the force (acceleration) levels. At the same time, the use of this structure results in the highest energy

Table 9.7 Comparison of the crash behavior of the hybrid S-rails with joints optimized using EA-LSM, ESL, and HCA for the mass minimization problem with a displacement constraint. Modified impact scenario with a rigid wall inclined by 5 [deg]. Top view.

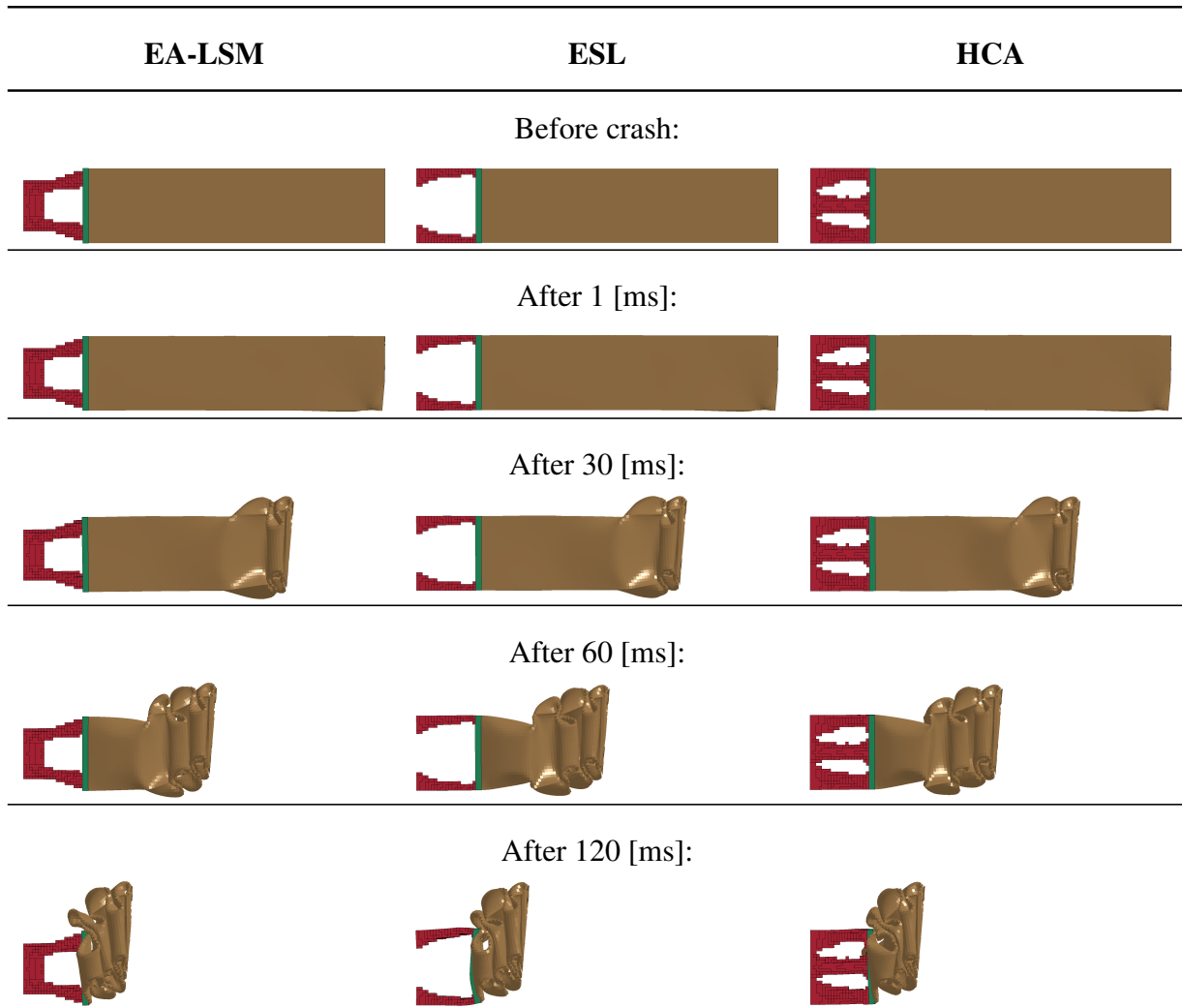


Table 9.8 Comparison of performance of the structures obtained using EA-LSM, ESL, and HCA for the mass minimization problem with a displacement constraint, in a modified crash case with a rigid wall inclined by 5 [deg].

Property	EA-LSM	ESL	HCA
Max. z-displacement	2.63 [mm]	8.10 [mm]	3.46 [mm]
Displacement increase (w.r.t. EA-LSM)	0%	+208%	+32%

absorption among all of the joint designs, related to the lowest velocity magnitude of the rigid wall at the end of the simulation. This is a beneficial behavior from the point of view of crashworthiness, which is achieved implicitly by constraining the displacement.

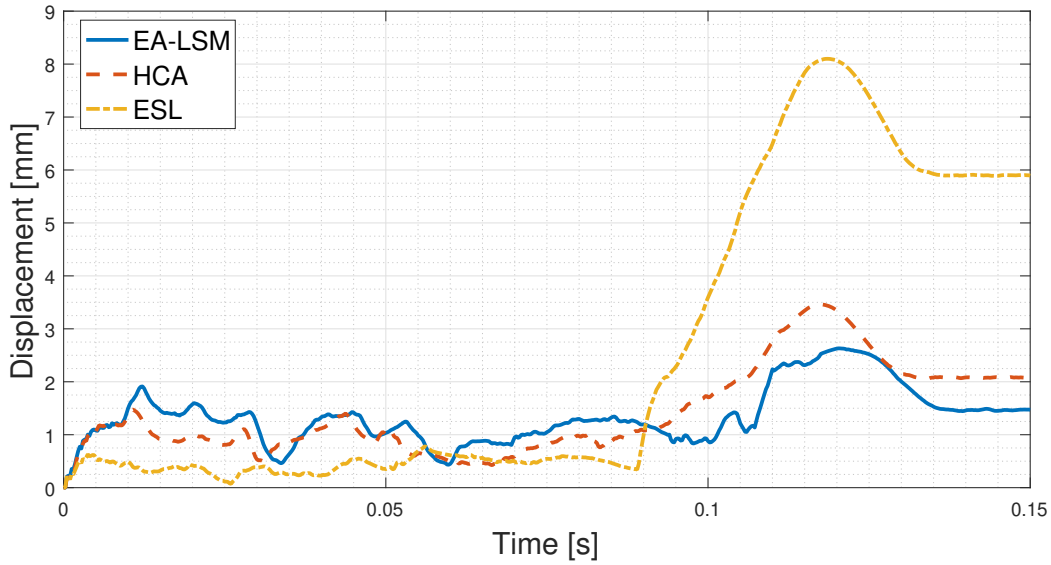


Figure 9.13 Displacement history (in the longitudinal direction) of the nodes on the joint with the highest displacements for the designs optimized with EA-LSM, HCA and ESL for the mass minimization problem with a displacement constraint. Test for a modified crash scenario with a rigid wall inclined by 5 [deg] relative to the initial configuration (Figure 9.12).

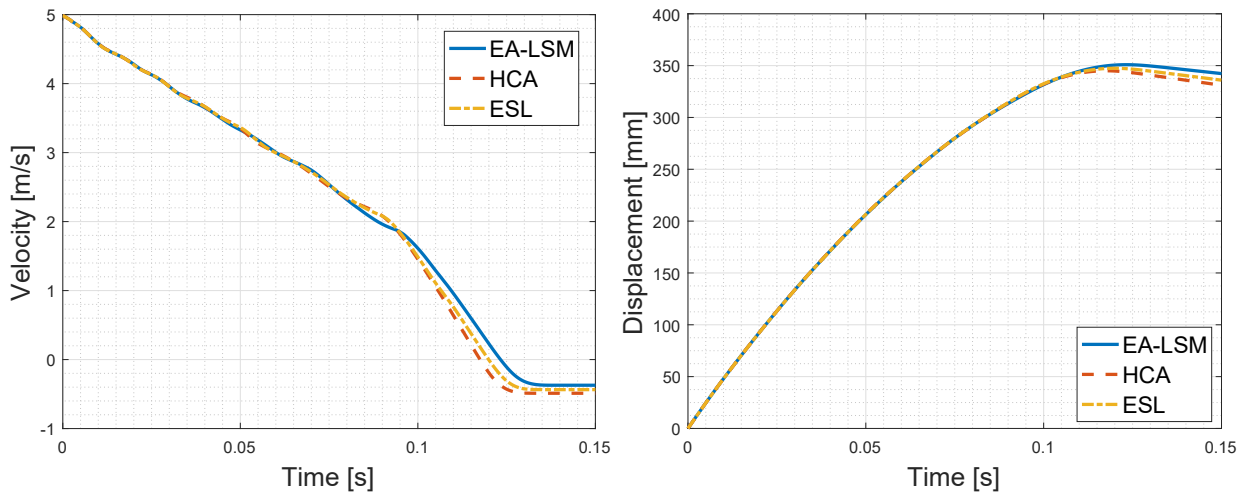


Figure 9.14 Velocity and displacement (in the longitudinal direction) of the rigid wall during the crash event for the designs optimized with EA-LSM, HCA and ESL for the mass minimization problem with a displacement constraint. Test for a modified crash scenario with a rigid wall inclined by 5 [deg] relative to the initial configuration (Figure 9.12).

The test case presented above shows the behavior of the structures at the limit case of an impact with a rigid wall inclined by 5 [deg]. However, it cannot be used to assess the robustness of the solutions obtained using different TO methods. To achieve that, for each of the joint designs, 100 crash simulations, with inclination angles ranging between -5 and $+5$ [deg] according to the Monte Carlo sampling, were performed. Figure 9.15 shows the distribution of the maximal nodal displacements at the joint for different inclination angles and summarizes them with corresponding box plots. In Table 9.9 the means, medians, and standard deviations of maximal displacements are compared.

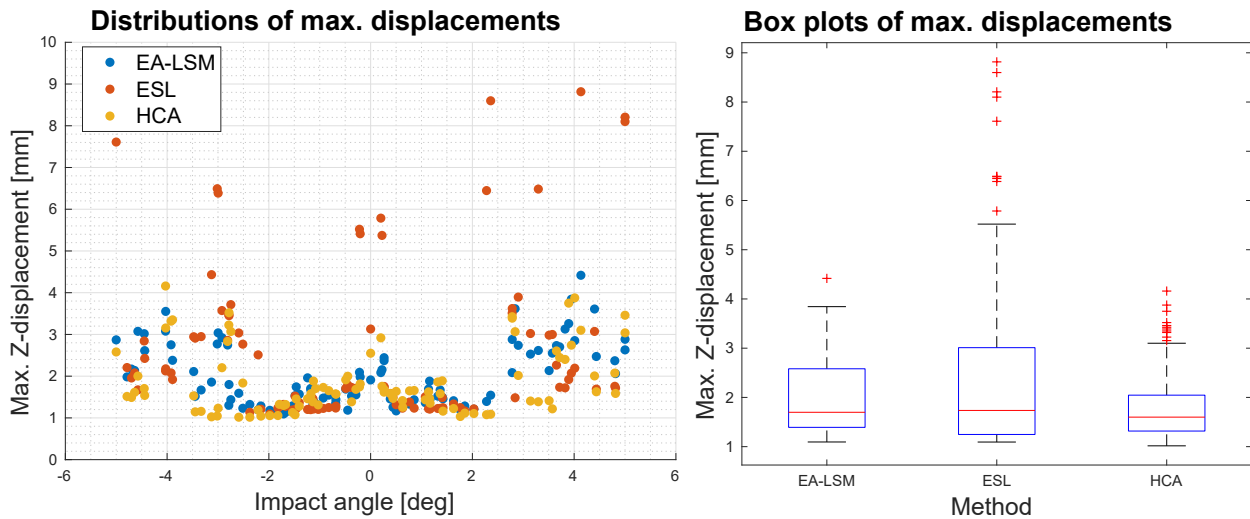


Figure 9.15 Distributions of the maximal displacements of the joint in the longitudinal direction for varying impact angles. Each point in the left plot corresponds to a single FE crash simulation. The plot on the right-hand side summarizes the distributions of the maximal displacements in form of box plots. On each box, the middle red line indicates the median, and the bottom and top edges of the blue box indicate the 25th (q_1) and 75th (q_2) percentiles of the normalized compliance values from the 100 simulations. The whiskers (dash black lines) extend to the most distant data points excluding outliers. The outliers are plotted individually using the red "+" symbol. Points are classified as outliers if they are greater than $q_3 + 1.5(q_3 - q_1)$ or smaller than $q_1 - 1.5(q_3 - q_1)$.

Table 9.9 Key statistics for the robustness study of structures generated using EA-LSM, ESL, and HCA in the mass minimization problem with a displacement constraint.

Property	EA-LSM	ESL	HCA
Mean max. z-displacement	1.99 [mm]	2.57 [mm]	1.86 [mm]
Median max. z-displacement	1.70 [mm]	1.74 [mm]	1.60 [mm]
Std. dev. of max. z-displacement	0.74 [mm]	1.88 [mm]	0.77 [mm]

Surprisingly, the joint optimized using HCA method achieves the lowest mean displacement, equal to 1.86 [mm]. In turn, the EA-LSM structure keeps the mean maximal displacement just below 2 [mm], and the ESL design leads to the mean maximal displacement of 2.57 [mm]. In terms of median values for the designs optimized with EA-LSM, ESL, and HCA, they are equal to 1.70, 1.74, and 1.60 [mm], respectively. However, a statistical evaluation of the data using the Wilcoxon rank sum test on the significance level of 1% did not allow for rejection of the null hypothesis assuming equality of medians. As a result, no statistically significant conclusions can be made regarding the medians of the displacement distributions for the three optimized structures.

Both the means and the plots in Figure 9.15 reveal an interesting tendency – initially, for very low inclination angles (between -0.5 and $+0.5$ [deg]), the maximal displacements grow considerably, which is particularly severe for ESL, to drop below 2 [mm] for all joints for moderate inclination

angles (between -2.0 and -0.5 [deg] as well as $+0.5$ and $+2.0$ [deg]). Most of the times, this is a better performance than for the corresponding cases without wall inclination. It turns out, that in those cases, the small inclination leads to higher stress concentrations in the contact region and better initiation of crash, without building strong asymmetry in the folding pattern (as shown in Table 9.7). As a result, the peak force at the beginning is reduced and the energy absorption in the first phase of the crash simulation increases, leading to a weaker loading in the second phase. Finally, the smaller forces acting on the structure result in smaller displacements of the joints compared to the reference design without wall inclination.

Although the differences between medians are not statistically significant, the standard deviations of the displacement distributions differ considerably, which is confirmed by statistical tests. Both the Levene's test (Levene (1961)) as well as the Brown-Forsythe test³ (Brown and Forsythe (1974)) reject the null hypothesis that the variance of the distributions corresponding to EA-LSM and ESL are equal at 1% significance level. The tests did not allow for rejection of the variance-equality hypothesis for the EA-LSM and HCA data sets. Indeed, especially for inclination angles below -2.0 and above $+2.0$ [deg], the ESL approach shows very high sensitivity to small variations of boundary conditions.

All in all, EA-LSM tends to result in solutions less sensitive to small variations of loading conditions than the state-of-the-art ESL approach, which is crucial in case of crashworthiness optimization. However, the structure optimized using the HCA method is also very robust. In principle, HCA method gradually adapts the design by adding mass in the regions with internal energy concentrations, subject to changing loading conditions during the crash event, which implicitly tests the robustness of the design. Still, the EA-LSM design offers comparable performance to HCA in terms of robustness, while being over 34% better in the test without wall inclination.

9.5. Mass minimization under an acceleration constraint

The following section presents the optimization results for the second test case described in Section 9.2. Similarly to the optimization problem presented in Section 9.3, the optimization objective is to minimize the mass of the joint with a constraint requiring the impactor to bounce back at the end of the simulation. However, instead of considering the intrusion as the second constraint, the maximal absolute acceleration⁴ of the impactor during the crash event is required to be kept below a certain threshold a_{req} . This problem formulation implicitly requires the structure of the joint to provide a stable support for the thin-walled tube, while addressing explicitly the criterion relevant from the perspective of the occupant protection.

³ Brown–Forsythe test uses median instead of mean (as the Levene's test does), which is recommended since it results in a better robustness against many kinds of non-normal distributions, while having a good statistical power (Derrick et al. (2018)).

⁴ No numerical filters on the accelerations were used during the optimization or in the final evaluation of the structural performance of the designs presented in this chapter.

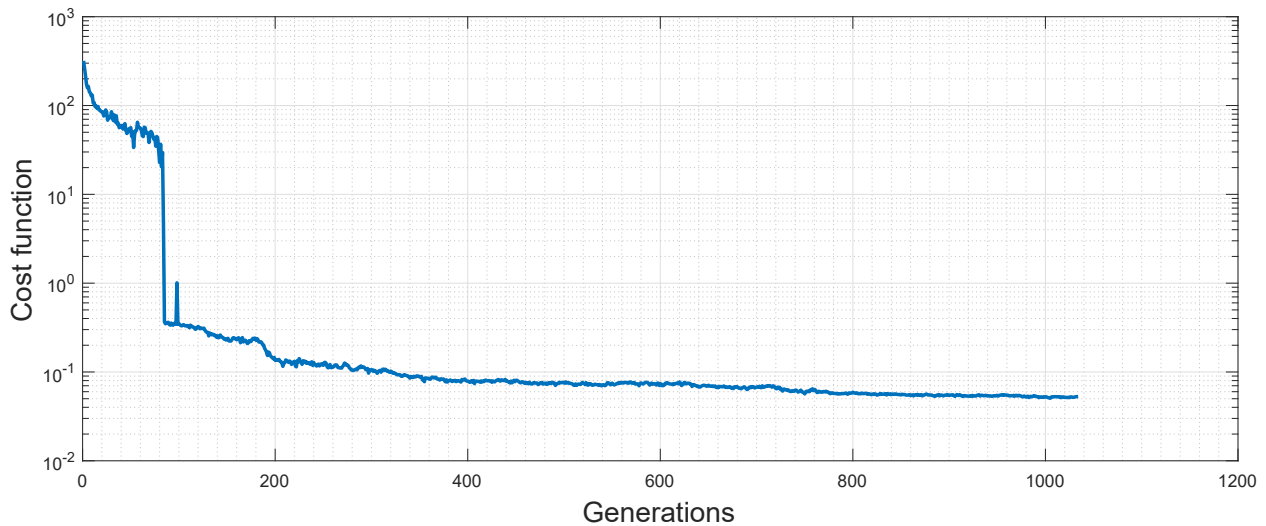


Figure 9.16 Plot of the minimized cost function for the mass minimization problem with a constraint on the maximal acceleration, addressed using EA-LSM. Acceleration constraint is set 10% lower than the maximal acceleration in a scenario with a joint occupying the entire design space.

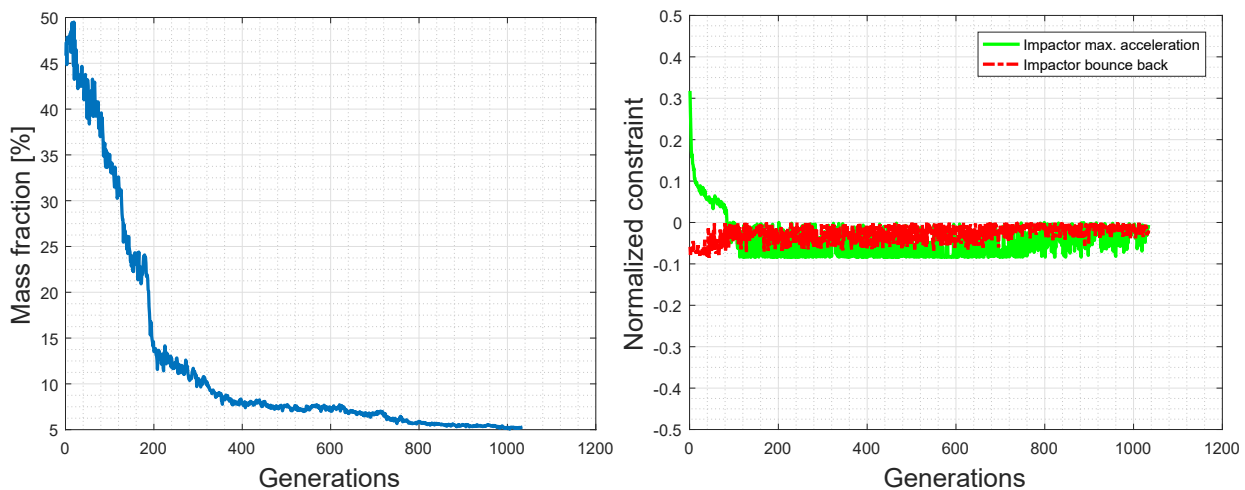
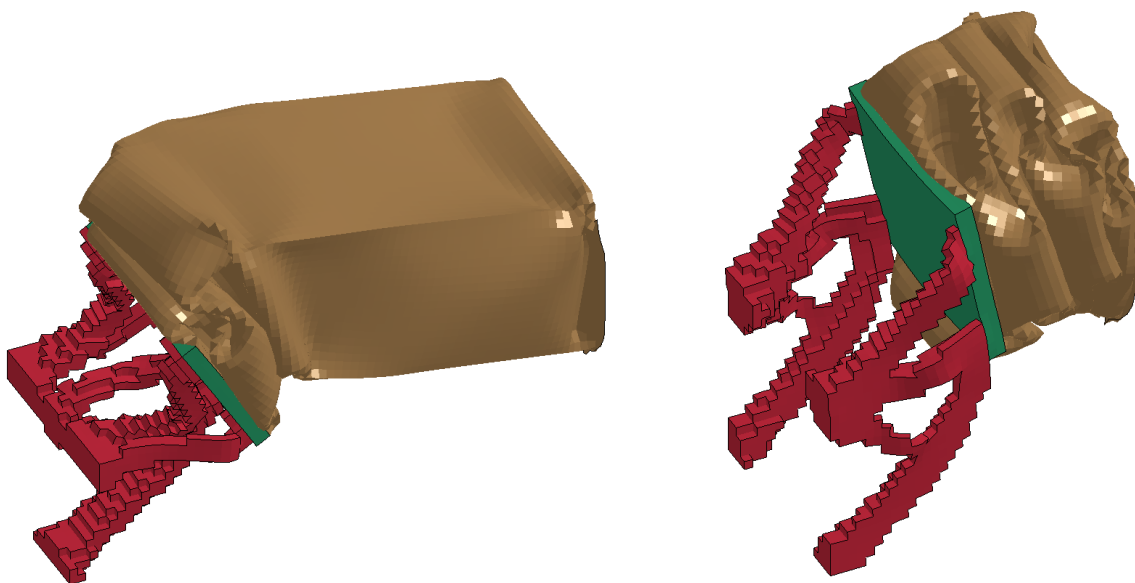


Figure 9.17 Plots of the mass fraction (left) and the normalized constraints (right) for the mass minimization problem with a constraint on the maximal acceleration, addressed using EA-LSM. The constraint on the maximal acceleration of the impactor in the longitudinal direction (green solid line) is calculated as $\frac{a_{max} - a_{req}}{a_{init}}$, where a_{max} is the maximal acceleration for the best design in a given generation and $a_{req} = 84.6[\text{m/s}^2]$ is its maximal allowed value. The second constraint (red dashed line), requiring the impactor to bounce back at the end of the simulation, or equivalently, to absorb the whole kinetic energy of the impactor, is computed as $\frac{v_{end} - 0}{v_{init}}$, where v_{end} is the longitudinal component of velocity of the impactor (initially positive) at the end of the simulation and $v_{init} = 5[\text{m/s}]$ is the initial velocity used to normalize the constraint. A constraint is violated if it is larger than 0.

The first optimization using EA-LSM resulted in a joint with very low mass, corresponding to only 5.44% of mass of the full design space. The maximal acceleration for this case was restricted to 90% of the peak acceleration for the crash case with a joint occupying the entire design domain. The convergence plot of the cost function during the optimization is presented in Figure 9.16, while the evolution of the objective and the constraints is shown in Figure 9.17. It is worth mentioning that although the initial design violates strongly the maximal acceleration constraint, the use of the exterior penalty method (Chapter 5) allows for finding a design satisfying both of the constraints already after ca. 100 generations. At that point, the second constraint, requiring the impactor

to bounce back at the end of the simulation, becomes active, which is understandable since the reduction of the peak force results in a longer duration of the crash event, while the energy to be absorbed remains unchanged. Once both constraints are satisfied, the optimizer further reduces the mass of the joint and the optimization is stopped after ca. 1000 generations. One should note a much more oscillatory character of the convergence plots compared to the ones presented in Section 9.3, which results from the high noisiness of the numerical estimate of the acceleration. However, even for this challenging optimization problem, EA-LSM is able to find a feasible design with a considerably reduced mass. The resulting structure is shown in Table 9.10 (EA-LSM 1).

Since both joint structures obtained using ESL and HCA for a mass constraint below 6% led to an unstable crash behavior, illustrated in Figure 9.18, another optimization using EA-LSM was performed in hope for discovering a different design concept with a higher mass fraction, which could be used for comparison with the state-of-the-art methods. To achieve that, a less stringent acceleration constraint was applied, corresponding to 95% of the peak acceleration for the crash case with a joint occupying the entire design space. The convergence of the cost function for the second optimization is depicted in Figure 9.19. The optimization starts again in an infeasible region and finds a feasible design after ca. 70 generations. After ca. 400 generations, the search stagnates, finding a completely different design concept, illustrated in Table 9.10 (EA-LSM 2). The mass of the optimized design corresponds to 6.96% of the mass of the full design space of the joint.



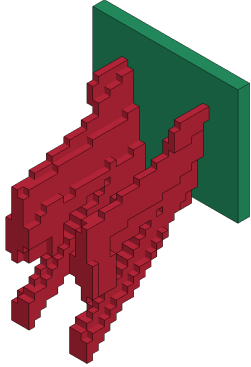
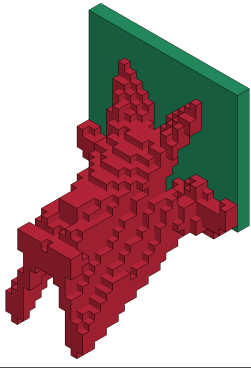
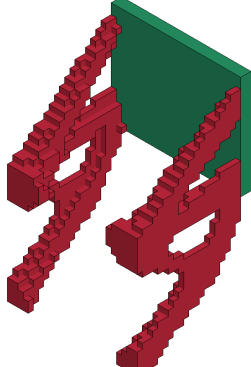
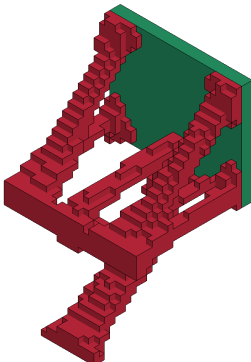
(a) Loss of stability for joint with volume fraction of 6.96%, optimized with HCA.

(b) Loss of stability for joint with volume fraction of 6.05%, optimized with ESL.

Figure 9.18 Illustration of the problems with stability of the joints optimized with HCA and ESL for very low volume fractions. The joint optimized with EA-LSM remains stable even for volume fraction of 5.44%.

Both of the structures differ considerably, showing the ability of EA-LSM to find different design

Table 9.10 Comparison of the joint structures obtained using EA-LSM, ESL, and HCA for the 3D joint mass minimization problem with an acceleration constraint. The values in parentheses express the percentage growth of maximal acceleration magnitude w.r.t. the EA-LSM 1 design.

Method	Design	Volume fraction	Max. acceleration [m/s^2]
EA-LSM 1		5.44%	83.62 (+0%)
EA-LSM 2		6.96%	89.38 (+7%)
ESL		6.97%	93.33 (+12%)
HCA		7.01%	FAIL

concepts. The first design (EA-LSM 1) supports only a small portion of the interface plate, with most of the material located in the bottom area. The design is relatively simple and composed of two large, inclined plates connected to long, diagonal bars. In contrast, the second joint design (EA-LSM 2) is considerably more complex, with four radial beams creating a strong connection to the interface plate. The structural differences between designs result in a considerably different crash behavior of both hybrid S-rails, presented in Table 9.11. As expected, the first joint undergoes substantially larger deformations, resulting in a progressive folding of the thin-walled tube close to the interface plate. On the other hand, the second design provides much more stable support, with a non-standard folding pattern of the tube.

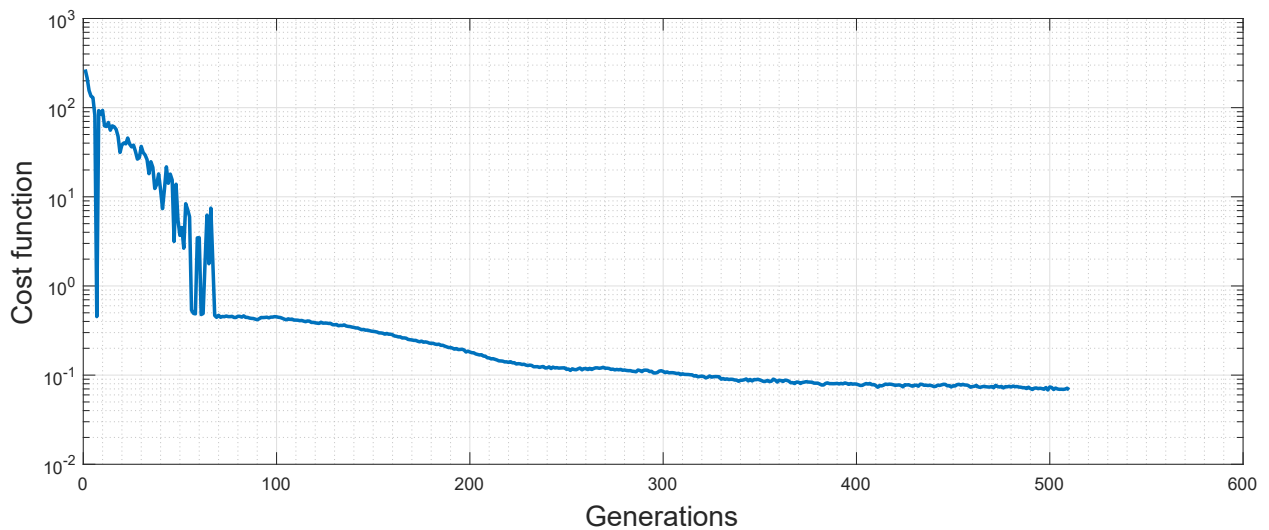


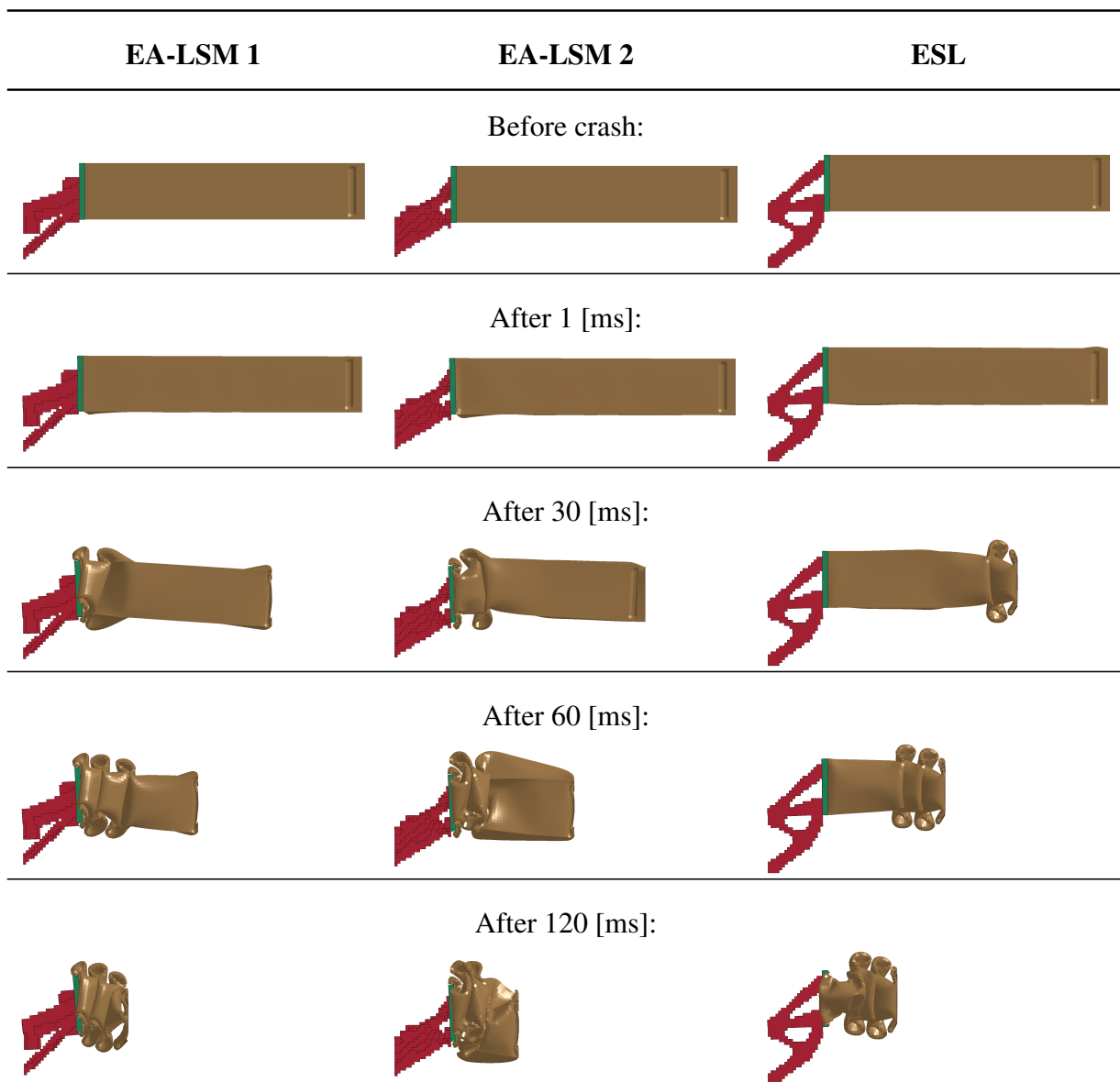
Figure 9.19 Plot of the minimized cost function for the mass minimization problem with a constraint on the maximal acceleration. Second optimization with EA-LSM. Acceleration constraint set 5% lower than the maximal acceleration in a scenario with a joint occupying the entire design space.

After the second optimization run with EA-LSM, the experiments with ESL and HCA were repeated for a mass constraint corresponding to the mass of the second design (mass fraction 6.96%). The optimized designs are presented in Table 9.10. This time, the ESL method was able to develop a structure providing a stable support for the thin-walled tube. In contrast, the design obtained using HCA approach was still unstable for the higher mass fraction (Figure 9.18). As a consequence, the behavior of the structures optimized using EA-LSM were compared with the ESL result only.

One should note that unlike EA-LSM, both ESL and HCA are not able to address the specified optimization problem directly. Namely, EA-LSM explicitly minimizes the mass of the structure under the constraint on maximal absolute acceleration, while ESL minimizes the compliance for the equivalent static case for a given mass fraction constraint. Similarly, HCA homogenizes the energy absorption under the constraint on the total mass. As a result, EA-LSM can offer considerable benefits for the optimization problems, which cannot be easily approximated using the above mentioned criteria.

Figure 9.20 shows a comparison of the absolute acceleration of the impacting rigid wall for both of the structures optimized using EA-LSM as well as the design obtained with ESL. The design optimized using state-of-the-art ESL approach results in a 12% higher peak acceleration than the lighter EA-LSM design and 7% higher peak acceleration than the heavier EA-LSM structure. The differences in the key performance metrics are reflected in deformation modes presented in Table 9.11. Additionally, Figure 9.21 shows the velocity and displacement history of the rigid wall impacting the hybrid structures with different types of joints.

Table 9.11 Comparison of the crash behavior of the hybrid S-rails with joints optimized using EA-LSM and ESL for the mass minimization problem with an acceleration constraint (HCA failed to find a stable structure for a similar mass fraction). Side view.



The results show that the EA-LSM can considerably outperform the HCA in the test scenario considered in this section, but offers only up to 12% reduction of maximal acceleration compared to ESL. It seems that the peak acceleration itself is very difficult to be reduced any further due to

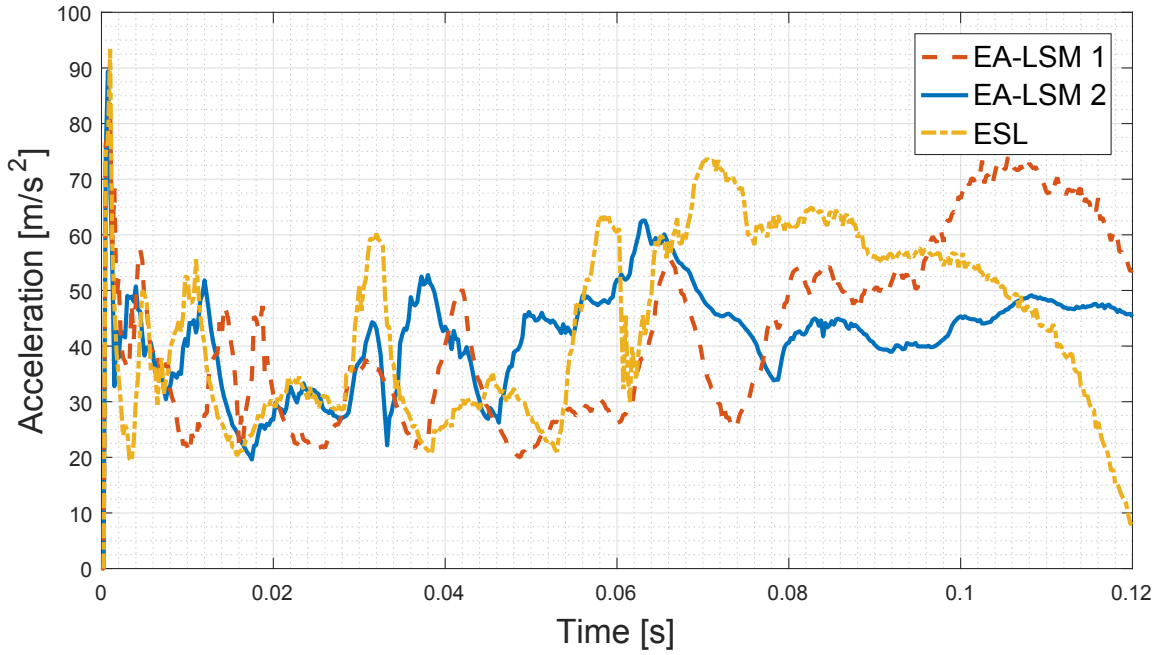


Figure 9.20 Absolute acceleration history (in the longitudinal direction) of the rigid wall impacting the hybrid S-rail structures with joints optimized with EA-LSM, HCA, and ESL. Mass minimization problem with a constraint on the maximal acceleration.

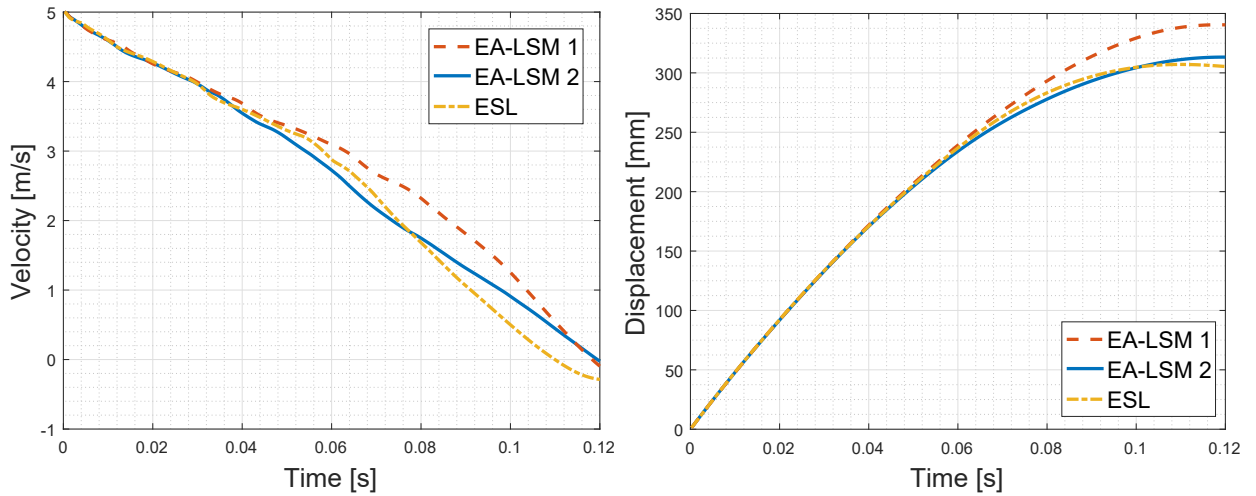


Figure 9.21 Velocity and displacement history (in the longitudinal direction) of the rigid wall impacting the hybrid S-rail structures with joints optimized with EA-LSM, HCA, and ESL. Mass minimization problem with a constraint on the maximal acceleration.

the physical limitations requiring the joint to provide a stable support for the thin-walled tube in the first phase of the crash. On the other hand, the obtained structures show a great potential of EA-LSM in addressing other objectives, such as HIC. The HIC values are calculated according to the following formula:

$$HIC = \max_{t_1, t_2} \left\{ \left[\frac{1}{t_2 - t_1} \int_{t_1}^{t_2} a(t) dt \right]^{2.5} (t_2 - t_1) \right\}, \quad (9.3)$$

where t_1 is the initial and t_2 the final time (given in [s]) maximizing the expression, while $a(t)$ is the acceleration expressed in [g], being the standard gravity acceleration ($g = 9.81[\text{m/s}^2]$). The time duration $t_2 - t_1$ is restricted usually to 36 [ms] (for HIC_{36}) or 15 [ms] (for HIC_{15}). HIC expresses the probability of the head injury during an impact and is a commonly used performance metric in crashworthiness design. Although HIC is normally computed for the center of gravity of the head of a dummy or a human model, for the test case studied in this section, it is calculated based on the acceleration of the impacting rigid wall. Obviously, by the third Newton's law of motion, both approaches would yield results scaled by the ratio of the masses of the vehicle and the wall.

Table 9.12 Head Injury Criterion (HIC) and corresponding maximal average accelerations for the designs obtained using EA-LSM and ESL.

Quantity	EA-LSM 1	EA-LSM 2	ESL
HIC_{36}	1.827	3.510	3.534
HIC_{15}	1.038	2.033	1.740
Avg. acc. for HIC_{36} [g]	4.810	6.246	6.263
Avg. acc. for HIC_{15} [g]	5.446	7.126	6.695

Table 9.12 presents the values of the HIC_{36} and HIC_{15} as well as the corresponding maximal average accelerations. The first design obtained with EA-LSM achieves almost two times smaller value of HIC_{36} than the design produced by ESL. The corresponding average absolute acceleration, calculated within t_1 and t_2 maximizing HIC_{36} , is reduced by over 23% compared to ESL design, which is a significant improvement. The second EA-LSM design offers comparable performance in terms of HIC_{36} to the ESL joint. Of course, in none of the cases discussed above the HIC value was minimized directly, but the obtained results demonstrate the potential of EA-LSM in developing structures considerably better for this criterion than the ones offered by ESL or HCA. In particular, in EA-LSM, HIC can be defined directly as the objective function or a constraint, which could potentially result in structures outperforming the state-of-the-art methods to even a greater extent.

All in all, this section shows the capability of EA-LSM to address challenging optimization problems, involving very noisy responses, such as accelerations. The method is able to develop designs significantly better than the ones obtained with state-of-the-art methods – HCA, failing to provide a stable support for the energy-absorbing tube, and ESL, offering designs with up to 14% higher peak accelerations than the structures obtained with EA-LSM. Finally, the evaluation of the HIC values for the optimized structures shows possibly even a higher potential of EA-LSM in delivering

better structures than the state-of-the-art methods when addressing this criterion directly.

9.6. Summary

This chapter demonstrates the usefulness of EA-LSM in TO of real-world crash structures, involving hundreds of design variables. The axial impact load cases considered in this chapter belong to one of the most challenging design problems in crashworthiness, concerning high nonlinearities, bifurcations and discontinuous responses. As shown in the studies, EA-LSM can be useful for generation of complex joint structures for novel hybrid design concepts involving metal AM. Unlike the state-of-the-art approaches, it can address the optimization problem directly by minimizing arbitrary quantifiable objectives. As a result, for the test cases considered in this chapter, the method is able to generate considerably better designs than the state-of-the-art methods, such as ESL method or HCA. In particular, EA-LSM was always able to provide mechanically stable designs, while structures generated using HCA method exhibited considerable stability problems for the second of the considered test cases. In both of the optimization problems, EA-LSM was able to handle more than one constraint, which is a clear advantage compared to the HCA method. It was also capable of providing feasible solutions for infeasible initial designs, which is an important property in case of industrial optimization problems, being usually highly constrained. Finally, the mutation operator used in ESs acts as a sensitivity tester, resulting in robust designs w.r.t. changing loading conditions. All in all, the study shows that the proposed method is capable of addressing complex crash optimization problems, close to the industrial cases of component design, and provides robust solutions superior to the designs obtained with the state-of-the-art methods.

Chapter 10

Conclusions and future work

10.1. Summary

This thesis proposes topology optimization approaches integrating methods from the domain of computational intelligence, in particular evolutionary computation and machine learning. The main application area as well as the key driving factor for the development of the proposed methods is vehicle crashworthiness, which is a field of engineering of very high importance and complexity, resulting from the strongly nonlinear character of the involved physical phenomena. As a consequence, for those problems, the utilization of the standard gradient-based approaches becomes prohibitive and alternative approaches have to be found. However, instead of focusing on problem-specific solutions, in this thesis, we aim for the development of topology optimization methods of a generic character, which would be potentially also applicable to a wide range of problems in structural mechanics, in particular when gradient-based methods cannot be used.

The main contribution of this dissertation is the development of the Evolutionary Level Set Method (EA-LSM), using Evolutionary Algorithms (EAs) to optimize structures based on a low-dimensional representation inspired by Level Set Methods (LSMs). EAs have demonstrated their applicability to a wide range of difficult real-world optimization problems, involving highly nonlinear, multi-modal, noisy, and discontinuous objective functions and constraints. Being non-gradient optimization methods, they can address any quantifiable optimization criteria, which can be evaluated based on black-box simulation software. As such, they are suitable for solving optimization problems in structural crashworthiness, based on high-fidelity commercial crash simulation software. In par-

ticular, we propose to use Evolution Strategies (ESs), including state-of-the-art Covariance Matrix Adaptation Evolution Strategy (CMA-ES), which are very efficient for parametrizations based on continuous design variables. Since the performance of EAs strongly depends on the dimensionality of the optimization problem, we represent the designs with a limited number of geometric components, referred also to as Moving Morphable Components (MMCs), reducing the number of design variables from thousands to tens, compared to the standard grid representations. Thanks to the utilized representation and the underlying problem-specific evolutionary optimization approach, including specialized repair operators, EA-LSM was shown to be capable of generating complex topologies for 2D and 3D optimization problems.

In order to validate the proposed method, its properties are thoroughly studied on established linear elastic static problems with known optima. EA-LSM using 16 MMCs, corresponding to 40 design variables when symmetry is enforced, is found to yield qualitatively similar designs to the ones obtained with state-of-the-art gradient-based approach using a grid representation with 2500–7500 density parameters. In contrast to the density-based methods, EA-LSM results in structures with clear material-void boundaries, which is visible especially for the test cases with high volume fractions. However, due to the non-deterministic character of EAs, each optimization run with EA-LSM can result in a different design concept, corresponding to a different optimum. We see it as an interesting property of the method rather than a disadvantage, since in the early design phases, the optimization criteria are frequently not precisely defined, and having the possibility to choose between different design concepts becomes much more important, which is reflected in a growing popularity of set-based optimization approaches. Finally, the experiments investigating the influence of such aspects as the type of the used initial design, or finite element mesh resolution, show that EA-LSM is robust w.r.t. the variations of the related parameters. The studies on the representation level, evaluating the influence of the number of MMCs on the performance and complexity of the obtained structures show that in EA-LSM, both computational costs and the granularity of the designs can be easily controlled. Last but not least, we show that EA-LSM is capable of addressing very high-dimensional optimization problems, involving up to 144 design variables, and resulting from 3D structural topology optimization problems for linear elasticity. Again, the method provides multiple design concepts, including topologies consistent with the ones obtained using gradient-based methods.

The evaluation of EA-LSM on nonlinear crash cases, based on explicit high-fidelity crash simulations, proves the usefulness of the method in case of difficult optimization problems, where standard gradient-based approaches are not applicable. For 2D and 3D transverse bending problems, frequently studied in the literature and targeting minimization of the intrusion of the impactor into the design space under a mass constraint, EA-LSM yields most of the time superior results to the state-of-the-art crash topology optimization methods – Hybrid Cellular Automata (HCA) and Equivalent Static Loads (ESL). However, for this optimization problem formulation, the differences between EA-LSM and HCA are often very small. As a consequence, the studies

demonstrate mostly the importance of considering the inertial and nonlinear effects in the optimization process. The true advantages of EA-LSM are clearly visible for the cases with more than one constraint, involving such criteria as the maximum absolute acceleration of the impactor. For those cases, EA-LSM can result in over 30% better performance than HCA, for the structures of identical mass.

Finally, the application of EA-LSM to a demanding real-world optimization of a 3D-printed metal joint in a hybrid S-rail structure shows its usefulness in the automotive industry. For this challenging, 288-dimensional optimization problem, EA-LSM can produce significantly better designs than ESL and HCA. In fact, for problem formulations involving injury-related criteria, EA-LSM finds designs of a very low mass, providing stable support to the energy-absorbing thin-walled tube. For the same mass fraction, ESL and HCA yield structures which collapse already in the initial phase of the crash event. For the problem of mass minimization with a displacement constraint of the joint, which is much more consistent with the assumptions of ESL and HCA, EA-LSM provides solutions with 64% and 34% lower displacements for the same joint mass, respectively. At the end, we evaluate the capability of EA-LSM for providing robust solutions. The rationale behind is that the mutation operator used in evolutionary optimization methods can itself act as a robustness tester since it introduces artificial noise similar to the one used in the standard robust design optimization techniques. Indeed, small variations of the impact angle lead to much smaller variations in terms of performance than for the design optimized using ESL approach. However, HCA is also able to provide a relatively robust design thanks to consideration of the changing loading conditions of the joint during the crash event.

The main disadvantage of EA-LSM remains its relatively high computational cost compared to the state-of-the-art gradient-based or heuristic topology optimization methods. However, thanks to a very good parallel scalability of evolutionary optimization methods, the number of optimization iterations, corresponding to the total time of the optimization with EA-LSM, can be reduced to similar levels, provided that sufficient computational resources are available. Nevertheless, due to demanding computational requirements of the crash simulations themselves, this might be not always feasible and it is very important to develop methods to reduce the number of costly finite element evaluations. For that reason, the second focus area of this dissertation is a development of optimization techniques based on machine learning to enhance the computational efficiency of EA-LSM.

At first, we propose to use an approximate gradient information to increase the local search capabilities of EA-LSM. The concept of hybridizing evolutionary search with gradient-based methods is known in the literature and sometimes such methods are referred to as the memetic algorithms. The method proposed in this work, Hybrid Evolutionary Level Set Method (H-EA-LSM), uses approximate gradients to improve randomly selected individuals from the mutated offspring population by moving them in the steepest descent direction. Subsequently, all of the individuals are

evaluated using the finite element software, and the standard non-elitist selection operator is applied both to the improved as well as the non-improved individuals. As a result, H-EA-LSM can work successfully even when the gradient information is very inaccurate. This property is demonstrated in this work both for static as well as crash load cases. For crash load cases, we propose to obtain gradient approximations either from simplified physical models, such as equivalent static cases, or from machine learning models of sensitivities, trained on previously acquired data. The second option is a generic solution and can be used for criteria where no suitable physical surrogates are available. Based on the experiments on 2D linear elastic and nonlinear crash structures, we show that even relatively simple linear regression models provide accuracy sufficient to increase significantly the convergence velocity in terms of optimization iterations compared to EA-LSM.

The second proposal follows the ideas from the field of surrogate-assisted evolutionary optimization. In case of very expensive crash simulation models, and when a coarse representation of the topology with a limited number of MMCs is sufficient, an attractive alternative is to use machine learning for approximating the objectives and constraints directly. Although approaches utilizing that concept, such as Efficient Global Optimization (EGO), are well-established in design optimization, to the best of our knowledge, they have never been applied to structural topology optimization problems before due to the high number of design variables in conventional representations. In this work, we describe the Kriging-Guided Level Set Method (KG-LSM) using EGO and a low-dimensional MMC-based representation, with problem-specific modifications for carrying out topology optimization. Based on 2D static cases, we show that the method is able to provide solutions consistent with the literature. In the experiments, the dimensionality of the optimization problems is gradually increased from 2 to 15, showing the limits of the method due to the curse of dimensionality, for the particular case of compliance minimization. Finally, we show that for a 15-dimensional intrusion minimization crash problem, the method is able to significantly reduce the number of finite element simulations compared to EA-LSM. In the considered test case, EA-LSM requires on average ca. 500 evaluations to reach similar intrusion values as KG-LSM after 150 simulations. Based on the experiments for 15-dimensional static and crash cases, one should note that the maximal dimensionality of the problems that can be addressed efficiently with KG-LSM can strongly depend on a particular characteristics of the considered objective functions and constraints.

Finally, we present the third, alternative, approach for improving the performance of EA-LSM, taking advantage of machine-learning-based predictions of favorable structural changes. Inspired by the concept of adaptive representations, which gradually increase the dimensionality of the used representation during the optimization process, we propose the Adaptive Evolutionary Level Set Method (A-EA-LSM). To allow for an easy and efficient handling of topological variations, A-EA-LSM uses a graph representation of topology, which is mapped to a corresponding set of MMCs. The method starts with a very few MMCs, which significantly increases the convergence velocity of evolutionary optimization. Once the optimization converges, the representation

is extended by adding new MMCs in the structure. In order to let the modified individuals to optimize so that they are able to compete with simpler designs, a niching-like technique is used and results in several topology types coexisting together in the population. The designs optimized in a lower-dimensional space are assumed to be good approximations of the optimal structures in higher-dimensional spaces. Since it is not clear where to add the new MMCs, we propose a learning-based topology variation operator, which predicts potentially the best locations of new structural connections. The operator uses a neural network classifier that compares different candidates for topological variations based on the structural state of the design. As a result, based on the automatically sampled data, it can learn topology variation rules for specific optimization objectives prior to the optimization. The main motivation behind this approach is the existence of engineering rules for part design, including the ones for crashworthiness, which can be potentially learned from the data. Hence, the proposed solution has a generic character and proves to be significantly better than random variations in the presented experiments. In the test for linear elasticity, we are able to reproduce the reference solutions known from the literature. However, for the non-linear crash case, A-EA-LSM yields a design close, but not identical to the solutions obtained with EA-LSM. Thus, although the performance of the design improves during the optimization, it is not clear if the adaptive process can always lead to the same optima as for the optimization carried out directly in the high-dimensional search space. To investigate that and to prove the usefulness of A-EA-LSM in general case, more research in this direction is needed.

All of the machine learning enhancements of EA-LSM can be seen as modules, which can be potentially combined together. Obviously, the underlying assumptions are sometimes contradicting, e.g. it might be not efficient to use A-EA-LSM together with KG-LSM, since all of the data samples collected for lower-dimensional representations would be useless for the extended representation. As a result, the interactions between the enhancements should be studied carefully. A good example of the synergistic effects coming from such a hybridization is described in our recent publication (Raponi et al. (2019b)), where KG-LSM was combined with EA-LSM. Another interesting aspect, not discussed in this thesis, but highlighting the potential of the proposed method, is the incorporation of manufacturing constraints into EA-LSM. We have demonstrated how to embed such constraints in the approach for generating self-supporting structures for additive manufacturing of crash components (Bujny et al. (2017c)). Finally, the concepts of EA-LSM were also used in an evolutionary approach for optimization of thin-walled extrusions, based on shell finite element meshes (Bujny et al. (2016a)). Since this is a problem of high practical relevance, with most of the car body components being sheet metal structures, an integration of the methods discussed here to this problem seems to be very interesting. The further ideas for promising future research directions are given in the next section.

10.2. Outlook

Beside the application of EA-LSM and its extensions to the optimization of thin-walled extrusions, the research on the design representations compatible with the proposed methods would potentially allow for addressing many interesting real-world structural optimization problems. In particular, using higher-order MMCs, e.g. with curved skeletons (Guo et al. (2016)), might increase the structural attainability of the method. It would be also very interesting to investigate if it is better to use less components of higher complexity or many more simple MMCs. Using EA-LSM for topometry optimization or joint shape and topology optimization seems to be also very interesting in the context of the optimization of e.g. stamped parts, and some initial experiments carried out in this direction were already very promising. As another important application, one could think about using the proposed methods for optimization of thin-walled structures by utilizing the hollow MMC approach (Bai and Zuo (2020)). One of the main shortcomings of the standard density-based methods is the fact that they generate topologies hard to interpret as thin-walled structures, which strongly limits the use of these methods in practice. Finally, using conforming mapping techniques based on the level-set field would allow for elimination of the unrealistic stress concentrations present in the standard density-based methods. Potentially, re-meshing techniques could be used to generate shell meshes from the level-set fields to address the layout optimization of thin-walled profiles.

Due to the interdisciplinary and stochastic character of the most of the real-world optimization problems, another future research direction could focus on using known techniques from the field of evolutionary optimization to address multi-objective problems (Deb (2001); Deb et al. (2002)) or robust design optimization (Beyer and Sendhoff (2007); Aspenberg (2011)). Using dimensionality reduction techniques such as principal component analysis (Wold et al. (1987)) or techniques relying on decoupling of optimization problems, e.g. cooperative coevolution (Potter and Jong (2000); Yang et al. (2008)), could be also very promising to further reduce the computational costs of EA-LSM.

Regarding the machine-learning enhancements of EA-LSM, especially further extensions of KG-LSM would be very interesting. For instance, one could combine KG-LSM with EA-LSM, via initialization of the parameters of CMA-ES based on the data collected by KG-LSM, as proposed by Mohammadi et al. (2015). However, the main limitation of KG-LSM is the low dimensionality of the optimization problems it can address. Thus, using techniques targeting high-dimensional surrogate-assisted optimization (Kyriacou et al. (2014); Wang et al. (2016); Bouhlef et al. (2016, 2018)) could substantially increase the applicability of KG-LSM.

Finally, the application of the proposed methods to the fields other than crashworthiness would show the true benefits coming from their generic character. For instance, the problems in manufacturing (Tavakoli and Davami (2009); Kor et al. (2009)), soft robotics (Laschi et al. (2016); Rieffel

et al. (2009); Hiller and Lipson (2012); Zhang et al. (2017a)), or selected types of compliant mechanism design problems (Kaminakis and Stavroulakis (2012); Tummala et al. (2013)) might pose difficulties to standard gradient-based optimizers, and, consequently, could take advantage of the methods proposed in this work.

Appendix A

Methods

This section of the appendix contains an additional information about the methods proposed in this work.

A.1. Features for learning-based topology variations

In Table A.1, the structural state features used by the topology variation predictor (Section 6.3.3) for the crash cases are given.

Table A.1 Structural state features used for prediction of favorable topological variations in the nonlinear crash cases. Since the field variables change over time in the transient crash simulation, only the maximal values of the quantities are taken into account.
*The characteristic sections are specified at 25%, 50%, and 75% of the length of the graph edge.

ID	Feature
5	Max. resultant displacement in the neighborhood of radius 1 around the graph node.
6	Max. strain energy in the neighborhood of radius 1 around the graph node.
7	Max. von Mises stress in the neighborhood of radius 1 around the graph node.
8-10	Max. resultant displacement at the 3 characteristic sections* along the edge.
11-13	Max. strain energy at the 3 characteristic sections* along the edge.
14-16	Max. von Mises stress at the 3 characteristic sections* along the edge.
17-19	Normal force at the 3 characteristic sections* along the edge.
20-22	Transverse force at the 3 characteristic sections* along the edge.
23-25	Bending moment at the 3 characteristic sections* along the edge.
26-28	Max. velocity at the 3 characteristic sections* along the edge.
29-31	Max. acceleration moment at the 3 characteristic sections* along the edge.

Appendix B

Algorithm settings

In this section of the appendix, the parameter settings for selected algorithms used in this work is given.

B.1. Parameter settings for SIMP

To validate the results obtained with EA-LSM in the 2D linear elastic cases, the state-of-the-art density-based method, utilizing the SIMP approach as well as a gradient-based Optimality Criteria (OC) optimizer, is used. In particular, we use the efficient MATLAB implementation by Andreassen et al. (2011), based on the previous work by Sigmund (2014). SIMP is utilized also within the ESL approach for the 3D crash cases, where an in-house Python code extending the approach described by Sigmund (2014) is used. The exact parameter settings for the SIMP used in the cases mentioned above is given in Table B.1.

Table B.1 Settings of the SIMP approach used in this thesis. The definitions of the parameters can be found in the paper by Sigmund (2014).

Parameter	Value
Filter radius (divided by the element size)	3
Penalization power	3
Filtering type	Sensitivity

B.2. Parameter settings for CMA-ES

In all of the EA-LSM optimization runs utilizing CMA-ES (Hansen and Kern (2004)), the hyperparameters of the algorithm are defined according to the rules specified in Table B.2, used by default in the Python implementation of CMA-ES by Hansen (2016).

Table B.2 Selected rules for defining the settings of CMA-ES(μ, λ) used in this work. The hyperparameters of the algorithm are defined based on the dimensionality of the optimization problem (n).

Hyperparameter	Rule
Number of offspring individuals	$\lambda = 4 + \lfloor 3 \ln(n) \rfloor$
Number of parent individuals	$\mu = \lfloor \frac{\lambda}{2} \rfloor$
Maximum number of iterations	$iter_{max} = \lfloor \frac{100+50(n+3)^2}{\lambda^{0.5}} \rfloor$
Termination if no improvement over $iter_{stag}$ iterations	$iter_{stag} = \lfloor 100 + 100 \frac{n^{1.5}}{\lambda} \rfloor$

The rest of the hyperparameters are set on the default levels, as well. To great extent, the rules used by Hansen (2016), e.g. for defining the number of parents, μ , and offspring, λ , are justified in the work by Hansen and Ostermeier (2001).

Appendix C

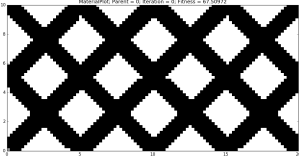
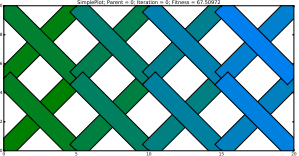
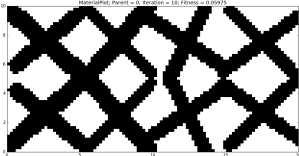
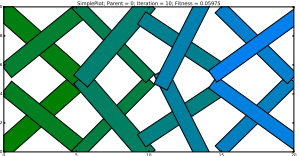

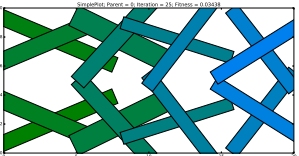

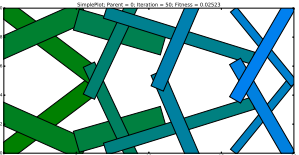

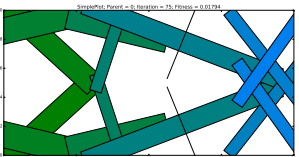

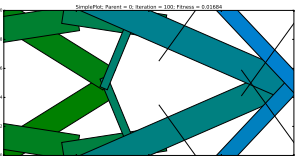

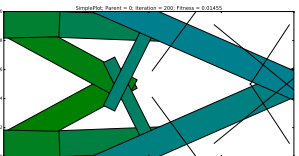

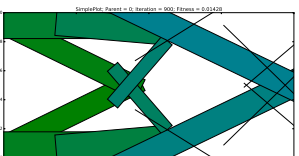
Results

This section of the appendix contains a supplementary material from the numerical experiments discussed in this thesis.

C.1. Design evolution in EA-LSM for the cantilever beam case

Table C.1 shows the evolution of the design represented with 16 MMCs during one of the optimization runs with EA-LSM for the linear elastic static cantilever beam case of compliance minimization with 50% volume constraint and aspect ratio 1:2, considered in Chapter 7. As an underlying optimization algorithm, CMA-ES(7,15) is used. For each generation, the material distribution and the layout of MMCs for the best individual in the population are shown. The initial design violates the 50% volume constraint. As a consequence, at the beginning, mainly the thickness of the MMCs on the right-hand side is reduced. After 75 generations, the most important structural connections are already established, and only some redundant material on the right-hand side, represented by four MMCs, has still to be removed. Between 200 and 900 generations, only minimal changes of the material distribution can be observed. In this phase, the compliance of the structure reduces by less than 2%. The final design is consistent with the solution obtained with the state-of-the-art SIMP approach (Table 7.2, 5th row).

Table C.1 Evolution of the best individual in EA-LSM for the cantilever beam test case with 1:2 aspect ratio and 50% volume constraint.

Topology		
Generation	Material distribution	MMC layout
0		
10		
25		
50		
75		
100		
200		
900		

Bibliography

- Aage N, Andreassen E, Lazarov BS (2015) Topology optimization using PETSc: An easy-to-use, fully parallel, open source topology optimization framework. *Structural and Multidisciplinary Optimization* 51(3):565–572, DOI 10.1007/s00158-014-1157-0
- Ahmed F, Bhattacharya B, Deb K (2013) Constructive solid geometry based topology optimization using evolutionary algorithm. In: *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012)*, no. 201 in *Advances in Intelligent Systems and Computing*, Springer India, pp 227–238
- Allaire G, Jouve F, Toader AM (2004) Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics* 194(1):363–393, DOI 10.1016/j.jcp.2003.09.032
- Allaire G, De Gournay F, Jouve F, Toader A (2005) Structural optimization using topological and shape sensitivity via a level set method. *Control and Cybernetics* 34(1):59
- Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O (2011) Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization* 43(1):1–16, DOI 10.1007/s00158-010-0594-7
- Apostol TM (1974) *Mathematical Analysis*. Addison-Wesley, Boston, MA, USA
- Arciszewski T, Michalski RS, Wnek J (1995) Constructive induction: the key to design creativity. In: *Proceedings of the Third International Round-Table Conference on Computational Models of Creative Design*, Heron Island, Queensland, Australia
- Arora JS (2012) *Introduction to Optimum Design (Third Edition)*. Academic Press, Boston, MA, USA
- Aspenberg D (2011) *Robust optimisation of structures: evaluation and incorporation of variations in simulation based design*. PhD Thesis, Linköping University, Linköping, Sweden
- Aulig N (2017) *Generic Topology Optimization Based on Local State Features*. PhD Thesis, Technische Universität Darmstadt, Darmstadt, Germany
- Aulig N, Olhofer M (2014a) Neuro-evolutionary topology optimization of structures by utilizing local state features. In: *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, ACM, Vancouver, BC, Canada, pp 967–974
- Aulig N, Olhofer M (2014b) Topology Optimization by Predicting Sensitivities Based on Local State Features. In: *11th World Congress on Computational Mechanics*, Barcelona, Spain

- Aulig N, Olhofer M (2016a) Evolutionary computation for topology optimization of mechanical structures: An overview of representations. In: 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, Vancouver, BC, Canada, pp 1948–1955, DOI 10.1109/CEC.2016.7744026
- Aulig N, Olhofer M (2016b) State-based representation for structural topology optimization and application to crashworthiness. In: 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, Vancouver, BC, Canada, pp 1642–1649, DOI 10.1109/CEC.2016.7743985
- Aulig N, Nutwell E, Menzel S, Detwiler D (2015) Towards Multi-objective Topology Optimization of Structures subject to Crash and Static Load Cases. In: 4th International Conference on Engineering Optimization, Lisbon, Portugal
- Aulig N, Nutwell E, Menzel S, Detwiler D (2018) Preference-based topology optimization for vehicle concept design with concurrent static and crash load cases. *Structural and Multidisciplinary Optimization* 57(1):251–266, DOI 10.1007/s00158-017-1751-z
- Autodesk (2018) What is Generative Design | Tools & Software | Autodesk. URL <https://www.autodesk.com/solutions/generative-design>, accessed: 2020-01-27
- Bai J, Zuo W (2020) Hollow structural design in topology optimization via moving morphable component method. *Structural and Multidisciplinary Optimization* 61(1):187–205, DOI 10.1007/s00158-019-02353-0
- Bandi P, Schmiedeler JP, Tovar A (2013) Design of Crashworthy Structures With Controlled Energy Absorption in the Hybrid Cellular Automaton Framework. *Journal of Mechanical Design* 135(9):091,002–091,002, DOI 10.1115/1.4024722
- Banga S, Gehani H, Bhilare S, Patel SJ, Kara LB (2018) 3D Topology Optimization Using Convolutional Neural Networks ArXiv:1808.07440
- Bäck T (1996) *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, Oxford, UK
- Bäck T (2014) Introduction to Evolution Strategies. In: *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, ACM, Vancouver, BC, Canada.
- Bäck T, Schwefel HP (1993) An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation* 1(1):1–23, DOI 10.1162/evco.1993.1.1.1
- Bäck T, Foussette C, Krause P (2013) *Contemporary Evolution Strategies*. Springer, New York, NY, USA
- Belytschko T, Xiao SP, Parimi C (2003) Topology optimization with implicit functions and regularization. *International Journal for Numerical Methods in Engineering* 57(8):1177–1196, DOI 10.1002/nme.824

- Bendsøe MP (1989) Optimal shape design as a material distribution problem. *Structural Optimization* 1(4):193–202, DOI 10.1007/BF01650949
- Bendsøe MP, Sigmund O (2004) *Topology Optimization*. Springer Berlin Heidelberg, Germany
- Beyer HG, Sendhoff B (2007) Robust optimization – A comprehensive survey. *Computer Methods in Applied Mechanics and Engineering* 196(33):3190–3218, DOI 10.1016/j.cma.2007.03.003
- Bishop C (2007) *Pattern Recognition and Machine Learning*. Springer, New York, NY, USA
- Bouhleb MA, Martins JRRA (2019) Gradient-enhanced kriging for high-dimensional problems. *Engineering with Computers* 35(1):157–173, DOI 10.1007/s00366-018-0590-x
- Bouhleb MA, Bartoli N, Otsmane A, Morlier J (2016) Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction. *Structural and Multidisciplinary Optimization* 53(5):935–952, DOI 10.1007/s00158-015-1395-9
- Bouhleb MA, Bartoli N, Regis RG, Otsmane A, Morlier J (2018) Efficient global optimization for high-dimensional constrained problems by using the Kriging models combined with the partial least squares method. *Engineering Optimization* 50(12):2038–2053, DOI 10.1080/0305215X.2017.1419344
- Brackett D, Ashcroft I, Hague R (2011) Topology optimization for additive manufacturing. In: *Proceedings of the Solid Freeform Fabrication Symposium*, Austin, TX, USA, pp 348–362
- Branke J (1998) Creating robust solutions by means of evolutionary algorithms. In: *Parallel Problem Solving from Nature — PPSN V*, Springer Berlin Heidelberg, Amsterdam, The Netherlands, *Lecture Notes in Computer Science*, pp 119–128
- Brown MB, Forsythe AB (1974) Robust Tests for the Equality of Variances. *Journal of the American Statistical Association* 69(346):364–367, DOI 10.2307/2285659
- Bruggi M, Duysinx P (2012) Topology optimization for minimum weight with compliance and stress constraints. *Structural and Multidisciplinary Optimization* 46(3):369–384, DOI 10.1007/s00158-012-0759-7
- Buhl T, Pedersen CBW, Sigmund O (2000) Stiffness design of geometrically nonlinear structures using topology optimization. *Structural and Multidisciplinary Optimization* 19(2):93–104, DOI 10.1007/s001580050089
- Bujny M (2015) *Development of a Hybrid Evolutionary Approach for Level Set Topology Optimization*. Master’s thesis, Technical University of Munich, Munich, Germany
- Bujny M, Aulig N, Olhofer M, Duddeck F (2016a) Evolutionary Crashworthiness Topology Optimization of Thin-Walled Structures. In: *11th ASMO UK / ISSMO / NOED2016: International Conference on Numerical Optimisation Methods for Engineering Design*, Munich, Germany

- Bujny M, Aulig N, Olhofer M, Duddeck F (2016b) Evolutionary Level Set Method for Crashworthiness Topology Optimization. European Community on Computational Methods in Applied Sciences, Crete Island, Greece
- Bujny M, Aulig N, Olhofer M, Duddeck F (2016c) Hybrid Evolutionary Approach for Level Set Topology Optimization. In: IEEE 2016 Congress on Evolutionary Computation (CEC), IEEE, Vancouver, Canada, pp 5092–5099, DOI 10.1109/CEC.2016.7748335
- Bujny M, Aulig N, Olhofer M, Duddeck F (2017a) Identification of optimal topologies for crashworthiness with the evolutionary level set method. *International Journal of Crashworthiness* 23(4):395–416, DOI 10.1080/13588265.2017.1331493
- Bujny M, Aulig N, Olhofer M, Duddeck F (2017b) Topology optimization of crash structures with the hybrid evolutionary level set method. In: 12th World Congress of Structural and Multidisciplinary Optimisation, Braunschweig, Germany
- Bujny M, Olhofer M, Duddeck F (2017c) Optimal structures for crash by additive manufacturing. In: 1st ECCOMAS Thematic Conference on Simulation for Additive Manufacturing, Munich, Germany
- Bujny M, Aulig N, Olhofer M, Duddeck F (2018) Learning-based Topology Variation in Evolutionary Level Set Topology Optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference, ACM, Kyoto, Japan, pp 825–832, DOI 10.1145/3205455.3205528
- Canisius M (2017) Adding Value with 3D Printing – Requirements for User, Design and Simulation. In: 2017 European Altair Technology Conference, Frankenthal, Germany
- Cavazzuti M, Baldini A, Bertocchi E, Costi D, Torricelli E, Moruzzi P (2010) High performance automotive chassis design: a topology optimization based approach. *Structural and Multidisciplinary Optimization* 44(1):45–56, DOI 10.1007/s00158-010-0578-7
- Cervera E, Trevelyan J (2005) Evolutionary structural optimisation based on boundary representation of NURBS. Part I: 2D algorithms. *Computers & Structures* 83(23–24):1902–1916, DOI 10.1016/j.compstruc.2005.02.016
- Challis VJ (2010) A discrete level-set topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization* 41(3):453–464, DOI 10.1007/s00158-009-0430-0
- Chapman CD, Saitou K, Jakiela MJ (1994) Genetic Algorithms as an Approach to Configuration and Topology Design. *Journal of Mechanical Design* 116(4):1005–1012, DOI 10.1115/1.2919480
- Chen BC, Kikuchi N (2001) Topology optimization with design-dependent loads. *Finite Elements in Analysis and Design* 37(1):57–70, DOI 10.1016/S0168-874X(00)00021-4

- Christensen J (2015) Topology optimisation of structures exposed to large (non-linear) deformations. PhD Thesis, Coventry University, Coventry, UK
- Christensen J, Bastien C, Blundell MV (2012) Effects of roof crush loading scenario upon body in white using topology optimisation. *International Journal of Crashworthiness* 17(1):29–38, DOI 10.1080/13588265.2011.625640
- Coello Coello CA (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering* 191(11–12):1245–1287, DOI 10.1016/S0045-7825(01)00323-1
- Dalcín L, Paz R, Storti M (2005) MPI for Python. *Journal of Parallel and Distributed Computing* 65(9):1108–1115, DOI 10.1016/j.jpdc.2005.03.010
- Dalcín L, Paz R, Storti M, D’Elía J (2008) MPI for Python: Performance improvements and MPI-2 extensions. *Journal of Parallel and Distributed Computing* 68(5):655–662, DOI 10.1016/j.jpdc.2007.09.005
- Dalcín LD, Paz RR, Kler PA, Cosimo A (2011) Parallel distributed computing using Python. *Advances in Water Resources* 34(9):1124–1139, DOI 10.1016/j.advwatres.2011.04.013
- Darwin C (1859) *On the origin of species*. John Murray, London, UK
- Deb K (2001) *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, New York, NY, USA
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2):182–197, DOI 10.1109/4235.996017
- Derrick B, Ruck A, Toher D, White P (2018) Tests for equality of variances between two samples which contain both paired observations and independent observations 13(2):36–47
- Dhondt G (2004) *The Finite Element Method for Three-Dimensional Thermomechanical Applications*. John Wiley & Sons, Munich, Germany
- Dijk NPv, Langelaar M, Keulen Fv (2012) Explicit level-set-based topology optimization using an exact Heaviside function and consistent sensitivity analysis. *International Journal for Numerical Methods in Engineering* 91(1):67–97, DOI 10.1002/nme.4258
- Dijk Nv, Maute K, Langelaar M, Keulen Fv (2013) Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization* 48(3):437–472, DOI 10.1007/s00158-013-0912-y
- Dirac PAM (1930) *The Principles of Quantum Mechanics*. Oxford University Press, Oxford, UK

- DTU (2018) Interactive 3D TopOpt App - TopOpt. URL <http://www.topopt.mek.dtu.dk/Apps-and-software/Interactive-3D-TopOpt-App>, accessed: 2020-01-26
- Du J, Olhoff N (2004) Topological optimization of continuum structures with design-dependent surface loading – Part I: new computational approach for 2D problems. *Structural and Multidisciplinary Optimization* 27(3):151–165, DOI 10.1007/s00158-004-0379-y
- Duddeck F (2008) Multidisciplinary optimization of car bodies. *Structural and Multidisciplinary Optimization* 35(4):375–389, DOI 10.1007/s00158-007-0130-6
- Duddeck F, Volz K (2012) A new topology optimization approach for crashworthiness of passenger vehicles based on physically defined equivalent static loads. In: *Proceedings ICRASH conference, Milan, Italy*
- Duddeck F, Hunkeler S, Lozano P, Wehrle E, Zeng D (2016) Topology optimization for crashworthiness of thin-walled structures under axial impact using hybrid cellular automata. *Structural and Multidisciplinary Optimization* 54(3):415–428, DOI 10.1007/s00158-016-1445-y
- Duysinx P, Bendsøe MP (1998) Topology optimization of continuum structures with local stress constraints. *International Journal for Numerical Methods in Engineering* 43(8):1453–1478, DOI 10.1002/(SICI)1097-0207(19981230)43:8<1453::AID-NME480>3.0.CO;2-2
- Eiben AE, Smith JE (2003) *Introduction to Evolutionary Computing*. Natural Computing Series, Springer Berlin Heidelberg, Germany
- Erhart A, Schumacher P, Lazarov N, Müllerschön H (2012) Topology optimization with LS-TaSC and Genesis/ESL for crash-loading. In: *11th LS-DYNA German users forum, Ulm, Germany*
- Eschenauer HA, Schumacher A (1997) Topology and Shape Optimization Procedures Using Hole Positioning Criteria. In: *Topology Optimization in Structural Mechanics, International Centre for Mechanical Sciences, Springer, Vienna*, pp 135–196, DOI 10.1007/978-3-7091-2566-3_4
- Eschenauer HA, Kobelev VV, Schumacher A (1994) Bubble method for topology and shape optimization of structures. *Structural Optimization* 8(1):42–51, DOI 10.1007/BF01742933
- Fang H, Rais-Rohani M, Liu Z, Horstemeyer MF (2005) A comparative study of meta-modeling methods for multiobjective crashworthiness optimization. *Computers & Structures* 83(25):2121–2136, DOI 10.1016/j.compstruc.2005.02.025
- Fang J, Sun G, Qiu N, Kim NH, Li Q (2016) On design optimization for structural crashworthiness and its state of the art. *Structural and Multidisciplinary Optimization* pp 1–29, DOI 10.1007/s00158-016-1579-y

- Fender G, Marburg S, Duddeck F (2016) Identification of a Set of Candidate Solutions for Optimal Positioning of Damping Layers. *SAE International Journal of Passenger Cars - Mechanical Systems* 9(3):987–994, DOI 10.4271/2016-01-1778
- Fender J, Duddeck F, Zimmermann M (2017) Direct computation of solution spaces. *Structural and Multidisciplinary Optimization* 55(5):1787–1796, DOI 10.1007/s00158-016-1615-y
- Forrester A, Sobester A, Keane A (2008) *Engineering Design via Surrogate Modelling: A Practical Guide*, 1st edn. Wiley, Chichester, West Sussex, England; Hoboken, NJ
- Fredricson H, Johansen T, Klarbring A, Petersson J (2003) Topology optimization of frame structures with flexible joints. *Structural and Multidisciplinary Optimization* 25(3):199–214, DOI 10.1007/s00158-003-0281-z
- Gaynor AT, Guest JK (2016) Topology optimization considering overhang constraints: Eliminating sacrificial support material in additive manufacturing through design. *Structural and Multidisciplinary Optimization* 54(5):1157–1172, DOI 10.1007/s00158-016-1551-x
- Gibbons JD, Chakraborti S (2011) Nonparametric Statistical Inference. In: *International Encyclopedia of Statistical Science*, Springer, Berlin, Heidelberg, pp 977–979, DOI 10.1007/978-3-642-04898-2_420
- Gibson I, Rosen D, Stucker B (2015) *Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*, 2nd edn. Springer-Verlag, New York, NY, USA
- Gil A, Segura J, Temme NM (2007) *Numerical Methods for Special Functions*. SIAM, Philadelphia, PA, USA
- Gilbert M, Tyas A (2003) Layout optimization of large-scale pin-jointed frames. *Engineering Computations* 20(8):1044–1064, DOI 10.1108/02644400310503017
- Goetz J, Tan H, Renaud J, Tovar A (2012) Two-material optimization of plate armour for blast mitigation using hybrid cellular automata. *Engineering Optimization* 44(8):985–1005, DOI 10.1080/0305215X.2011.624182
- Gomes AA, Suleman A (2006) Application of spectral level set methodology in topology optimization. *Structural and Multidisciplinary Optimization* 31(6):430–443, DOI 10.1007/s00158-006-0005-2
- Grobler C, Kok S, Wilke DN (2018) Simple Intuitive Multi-objective Parallelization of Efficient Global Optimization: SIMPLE-EGO. In: *Advances in Structural and Multidisciplinary Optimization*, Springer International Publishing, Braunschweig, Germany, pp 205–220

- Guirguis D, Aly M, Hamza K, Hegazi H (2015) Image Matching Assessment of Attainable Topology via Kriging-Interpolated Level-Sets. In: ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers Digital Collection, Buffalo, NY, USA, DOI 10.1115/DETC2014-34622
- Guo T, Lohan DJ, Cang R, Ren MY, Allison JT (2018) An Indirect Design Representation for Topology Optimization Using Variational Autoencoder and Style Transfer. In: 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, American Institute of Aeronautics and Astronautics, Kissimmee, FL, USA, DOI 10.2514/6.2018-0804
- Guo X (2014) Doing topology optimization explicitly and geometrically: a new moving morphable components based framework. In: *Frontiers in Applied Mechanics*, Imperial College Press, London, UK, pp 31–32
- Guo X, Zhang W, Zhang J, Yuan J (2016) Explicit structural topology optimization based on moving morphable components (MMC) with curved skeletons. *Computer Methods in Applied Mechanics and Engineering* 310:711–748, DOI 10.1016/j.cma.2016.07.018
- Hagishita T, Ohsaki M (2009) Topology optimization of trusses by growing ground structure method. *Structural and Multidisciplinary Optimization* 37(4):377–393, DOI 10.1007/s00158-008-0237-4
- Hammer V, Olhoff N (2000) Topology optimization of continuum structures subjected to pressure loading. *Structural and Multidisciplinary Optimization* 19(2):85–92, DOI 10.1007/s001580050088
- Hamza K, Aly M, Hegazi H (2013) A Kriging-Interpolated Level-Set Approach for Structural Topology Optimization. *Journal of Mechanical Design* 136(1):011,008, DOI 10.1115/1.4025706
- Han YH, Witowski K, Lazarov N, Anakiev K (2015) Topometry and Shape Optimization of a Hood. In: 10th European LS-DYNA Conference, Würzburg, Germany
- Hansen N (2016) CMA Evolution Strategy Source Code. URL <http://www.cmap.polytechnique.fr/~nikolaus.hansen/html-pythoncma/frames.html>, accessed: 2016-08-02
- Hansen N, Kern S (2004) Evaluating the CMA Evolution Strategy on Multimodal Test Functions. In: *Parallel Problem Solving from Nature - PPSN VIII*, no. 3242 in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, Germany, pp 282–291
- Hansen N, Ostermeier A (1996) Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In: *Proceedings of IEEE International Conference on Evolutionary Computation*, Nagoya, Japan, pp 312–317, DOI 10.1109/ICEC.1996.542381

- Hansen N, Ostermeier A (2001) Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation* 9(2):159–195, DOI 10.1162/106365601750190398
- Hansen N, Müller S, Koumoutsakos P (2003) Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation* 11(1):1–18, DOI 10.1162/106365603321828970
- Hassani B, Hinton E (1998) A review of homogenization and topology optimization I—homogenization theory for media with periodic structure. *Computers & Structures* 69(6):707–717, DOI 10.1016/S0045-7949(98)00131-X
- Hiller J, Lipson H (2012) Automatic Design and Manufacture of Soft Robots. *IEEE Transactions on Robotics* 28(2):457–466, DOI 10.1109/TRO.2011.2172702
- Hörmander L (2003) *The Analysis of Linear Partial Differential Operators I: Distribution Theory and Fourier Analysis*. Springer, Berlin, Heidelberg, Germany
- Huang X, Xie Y (2010) *Evolutionary Topology Optimization of Continuum Structures: Methods and Applications*. John Wiley & Sons Ltd., Chichester, West Sussex, UK
- Huang X, Xie YM, Lu G (2007) Topology optimization of energy-absorbing structures. *International Journal of Crashworthiness* 12(6):663–675, DOI 10.1080/13588260701497862
- Hunkeler S (2013) *Topology Optimisation in Crashworthiness Design via Hybrid Cellular Automata for Thin-walled Structures*. PhD Thesis, Queen Mary University of London, London, UK
- Iserles A (2008) *A First Course in the Numerical Analysis of Differential Equations* by Arieh Iserles. Cambridge University Press, Cambridge, UK
- Jeong S, Obayashi S (2005) Efficient global optimization (EGO) for multi-objective problem and data mining. In: 2005 IEEE Congress on Evolutionary Computation, Edinburgh, UK, vol 3, pp 2138–2145 Vol. 3, DOI 10.1109/CEC.2005.1554959
- Jin Y (2011) Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation* 1(2):61–70, DOI 10.1016/j.swevo.2011.05.001
- Jones DR, Schonlau M, Welch WJ (1998) Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization* 13(4):455–492, DOI 10.1023/A:1008306431147
- Kaminakis NT, Stavroulakis GE (2012) Topology optimization for compliant mechanisms, using evolutionary-hybrid algorithms and application to the design of auxetic materials. *Composites Part B: Engineering* 43(6):2655–2668, DOI 10.1016/j.compositesb.2012.03.018

- Kanter JLCGd (2006) Energy Absorption of Monolithic and Fibre Reinforced Aluminium Cylinders. PhD Thesis, Delft University of Technology, Delft, The Netherlands
- Karev A, Harzheim L, Immel R, Erzgräber M (2018) Comparison of Different Formulations of a Front Hood Free Sizing Optimization Problem Using the ESL-Method. In: *Advances in Structural and Multidisciplinary Optimization*, Springer International Publishing, Braunschweig, Germany, pp 933–951, DOI 10.1007/978-3-319-67988-4_71
- Karev A, Harzheim L, Immel R, Erzgräber M (2019) Free sizing optimization of a front hood using the ESL method: overcoming challenges and traps. *Structural and Multidisciplinary Optimization* 60(4):1687–1707, DOI 10.1007/s00158-019-02285-9
- Kim DH, Jung KH, Kim DJ, Park SH, Kim DH, Lim J, Nam BG, Kim HS (2017) Improving pedestrian safety via the optimization of composite hood structures for automobiles based on the equivalent static load method. *Composite Structures* 176:780–789, DOI 10.1016/j.compstruct.2017.06.016
- Kor J, Chen X, Sun Z, Hu H (2009) Casting Design through Multi-objective Optimization. In: *2009 Second International Conference on Future Information Technology and Management Engineering*, Milan, Italy, pp 604–608, DOI 10.1109/FITME.2009.156, iSSN: null
- Kresslein J, Haghghi P, Park J, Ramnath S, Sutradhar A, Shah JJ (2018) Automated cross-sectional shape recovery of 3D branching structures from point cloud. *Journal of Computational Design and Engineering* 5(3):368–378, DOI 10.1016/j.jcde.2017.11.010
- Krige DG (1951) A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy* 52(6):119–139
- Krischer L (2018) Gradient Approximation for Crash Topology Optimization. Master's thesis, Technical University of Munich, Munich, Germany
- Kudela L, Zander N, Bog T, Kollmannsberger S, Rank E (2015) Efficient and accurate numerical quadrature for immersed boundary methods. *Advanced Modeling and Simulation in Engineering Sciences* 2(1):10, DOI 10.1186/s40323-015-0031-y
- Kyriacou SA, Asouti VG, Giannakoglou KC (2014) Efficient PCA-driven EAs and metamodel-assisted EAs, with applications in turbomachinery. *Engineering Optimization* 46(7):895–911, DOI 10.1080/0305215X.2013.812726
- Langelaar M (2016) Topology optimization of 3D self-supporting structures for additive manufacturing. *Additive Manufacturing* 12:60–70, DOI 10.1016/j.addma.2016.06.010
- Laschi C, Mazzolai B, Cianchetti M (2016) Soft robotics: Technologies and systems pushing the boundaries of robot abilities. *Science Robotics* 1(1), DOI 10.1126/scirobotics.aah3690

- Leary M, Merli L, Torti F, Mazur M, Brandt M (2014) Optimal topology for additive manufacture: A method for enabling additive manufacture of support-free optimal structures. *Materials & Design* 63:678–690, DOI 10.1016/j.matdes.2014.06.015
- LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444, DOI 10.1038/nature14539
- Lee KH, Kang DH (2007) Structural optimization of an automotive door using the kriging interpolation method. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 221(12):1525–1534, DOI 10.1243/09544070JAUTO403
- Lehmann H, Menzel S (2012) Evolvability as concept for the optimal design of free-form deformation control volumes. In: 2012 IEEE Congress on Evolutionary Computation, Brisbane, QLD, Australia, pp 1–8, DOI 10.1109/CEC.2012.6256510
- Lei X, Liu C, Du Z, Zhang W, Guo X (2018) Machine Learning Driven Real Time Topology Optimization under Moving Morphable Component (MMC)-Based Framework. *Journal of Applied Mechanics* 86(1), DOI 10.1115/1.4041319
- Levene H (1961) Robust tests for equality of variances. *Contributions to probability and statistics Essays in honor of Harold Hotelling* pp 279–292
- Lim D, Ong YS, Jin Y, Sendhoff B (2007) A Study on Metamodeling Techniques, Ensembles, and Multi-surrogates in Evolutionary Computation. In: *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, ACM, London, England, pp 1288–1295, DOI 10.1145/1276958.1277203
- Lim D, Jin Y, Ong YS, Sendhoff B (2010) Generalizing Surrogate-Assisted Evolutionary Computation. *IEEE Transactions on Evolutionary Computation* 14(3):329–355, DOI 10.1109/TEVC.2009.2027359
- Liu J, Ma Y (2016) A survey of manufacturing oriented topology optimization methods. *Advances in Engineering Software* 100:161–175, DOI 10.1016/j.advengsoft.2016.07.017
- Liu K (2018) Cluster-Based Structural Optimization and Applications to Crashworthiness. PhD Thesis, Purdue University, West Lafayette, IN, USA
- Liu K, Tovar A, Nutwell E, Detwiler D (2015) Thin-Walled Compliant Mechanism Component Design Assisted by Machine Learning and Multiple Surrogates. *SAE Technical Paper* 2015-01-1369 DOI 10.4271/2015-01-1369
- Liu K, Detwiler D, Tovar A (2017) Metamodel-Based Global Optimization of Vehicle Structures for Crashworthiness Supported by Clustering Methods. In: *Advances in Structural and Multidisciplinary Optimization*, Springer, Cham, pp 1545–1557, DOI 10.1007/978-3-319-67988-4_116

- Liu Y (2008) Crashworthiness design of multi-corner thin-walled columns. *Thin-Walled Structures* 46(12):1329–1337, DOI 10.1016/j.tws.2008.04.003
- LSTC (2011) LS-TaSC - Topology and Shape Computations for LS-DYNA, v2.0
- LSTC (2014) LS-DYNA KEYWORD USER'S MANUAL, Volume II - Material Models
- LZN (2017) A car body inspired by nature. The flexible car body out of a 3D-Printer. Tech. rep., Laser Zentrum Nord, Hamburg, Germany
- Maaranen H, Miettinen K, Penttinen A (2007) On initial populations of a genetic algorithm for continuous optimization problems. *Journal of Global Optimization* 37(3):405, DOI 10.1007/s10898-006-9056-6
- Matejka J, Glueck M, Bradner E, Hashemi A, Grossman T, Fitzmaurice G (2018) Dream Lens: Exploration and Visualization of Large-Scale Generative Design Datasets. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ACM, New York, NY, USA, CHI '18, pp 369:1–369:12, DOI 10.1145/3173574.3173943
- Maute K, Schwarz S, Ramm E (1998) Adaptive topology optimization of elastoplastic structures. *Structural Optimization* 15(2):81–91, DOI 10.1007/BF01278493
- Menzel S (2011) Evolvable free-form deformation control volumes for evolutionary design optimization. In: *2011 IEEE Congress of Evolutionary Computation (CEC)*, New Orleans, LA, USA, pp 1388–1395, DOI 10.1109/CEC.2011.5949778
- Michalewicz Z (1996) *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Berlin Heidelberg, Berlin, Heidelberg, Germany
- Michell A (1904) The limits of economy of material in frame-structures. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 8(47):589–597, DOI 10.1080/14786440409463229
- Müller SD, Hansen N, Koumoutsakos P (2002) Increasing the Serial and the Parallel Performance of the CMA-Evolution Strategy with Large Populations. In: *Parallel Problem Solving from Nature — PPSN VII*, no. 2439 in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, Germany, pp 422–431, DOI 10.1007/3-540-45712-7_41
- Müllerschön H, Erhart A, Anakiev K, Schumacher P, Kassegger H (2013) Application of the equivalent static load method for impact problems with GENESIS and LS-DYNA. In: *Automotive CAE Grand Challenge*, Hanau, Germany
- Mohammadi H, Le Riche R, Touboul E (2015) Making EGO and CMA-ES Complementary for Global Optimization. In: *Learning and Intelligent Optimization*, vol 8994, Springer International Publishing, Basel, Germany, pp 287–292, DOI 10.1007/978-3-319-19084-6_29

- Mozumder C, Renaud JE, Tovar A (2012) Topometry optimisation for crashworthiness design using hybrid cellular automata. *International Journal of Vehicle Design* 60(1/2):100, DOI 10.1504/IJVD.2012.049160
- Mozumder CK (2010) Topometry optimization of sheet metal structures for crashworthiness design using hybrid cellular automata. PhD Thesis, University of Notre Dame, Notre Dame, IN, USA
- Mulard A (2018) Crash Topology Optimisation for Additive Manufacturing Structures. Master's thesis, Technical University of Munich, Munich, Germany
- Myers JL, Well A (2003) *Research design and statistical analysis*, 2nd edn. Lawrence Erlbaum Associates, Mahwah, NJ, USA
- Neri F, Cotta C (2012) Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation* 2:1–14, DOI 10.1016/j.swevo.2011.11.003
- Neves MM, Rodrigues H, Guedes JM (1995) Generalized topology design of structures with a buckling load criterion. *Structural Optimization* 10(2):71–78, DOI 10.1007/BF01743533
- Nocedal J, Wright SJ (2006) *Numerical optimization*. Springer Science+Business Media, New York, NY, USA
- Norato JA, Bendsøe MP, Haber RB, Tortorelli DA (2007) A topological derivative method for topology optimization. *Structural and Multidisciplinary Optimization* 33(4):375–386, DOI 10.1007/s00158-007-0094-6
- Norato JA, Bell BK, Tortorelli DA (2015) A geometry projection method for continuum-based topology optimization with discrete elements. *Computer Methods in Applied Mechanics and Engineering* 293:306–327, DOI 10.1016/j.cma.2015.05.005
- Novotny AA, Sokołowski J (2013) *Topological Derivatives in Shape Optimization. Interaction of Mechanics and Mathematics*, Springer Berlin Heidelberg, Germany
- Nutwell E, Aulig N, Detwiler D (2017) Topology Optimization of a Bumper Beam System Considering a Nonlinear Design Requirement. In: *NAFEMS World Congress 2017*, Stockholm, Sweden
- Oh S, Jung Y, Kim S, Lee I, Kang N (2019) Deep Generative Design: Integration of Topology Optimization and Generative Models ArXiv:1903.01548
- Olhofer M, Jin Y, Sendhoff B (2001) Adaptive encoding for aerodynamic shape optimization using evolution strategies. In: *Proceedings of the 2001 Congress on Evolutionary Computation*, Seoul, South Korea, vol 1, pp 576–583

- Ortmann C (2015) Entwicklung eines graphen- und heuristikbasierten Verfahrens zur Topologieoptimierung von Profilquerschnitten für Crashlastfälle. PhD Thesis, Bergische Universität Wuppertal, Wuppertal, Germany
- Ortmann C, Schumacher A (2013) Graph and heuristic based topology optimization of crash loaded structures. *Structural and Multidisciplinary Optimization* 47(6):839–854, DOI 10.1007/s00158-012-0872-7
- Park GJ (2010) Technical overview of the equivalent static loads method for non-linear static response structural optimization. *Structural and Multidisciplinary Optimization* 43(3):319–337, DOI 10.1007/s00158-010-0530-x
- Patel NM (2007) Crashworthiness Design Using Topology Optimization. PhD Thesis, University of Notre Dame, Notre Dame, IN, USA
- Patel NM, Penninger CL, Renaud JE (2009) Topology Synthesis of Extrusion-Based Nonlinear Transient Designs. *Journal of Mechanical Design* 131(6), DOI 10.1115/1.3116255
- Pearson K (1905) The Problem of the Random Walk. *Nature* 72(1865):294, DOI 10.1038/072294b0
- Pedersen CBW (2003) Topology optimization design of crushed 2D-frames for desired energy absorption history. *Structural and Multidisciplinary Optimization* 25(5-6):368–382, DOI 10.1007/s00158-003-0282-y
- Pedersen N (2000) Maximization of eigenvalues using topology optimization. *Structural and Multidisciplinary Optimization* 20(1):2–11, DOI 10.1007/s001580050130
- Pedro HTC, Kobayashi MH (2011) On a cellular division method for topology optimization. *International Journal for Numerical Methods in Engineering* 88(11):1175–1197, DOI 10.1002/nme.3218
- Pehne I, Vučina D, Vlak F (2016) Evolutionary topology optimization using parameterized b-spline surface. In: VII European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS Congress 2016), p 8283
- Potter MA, Jong KAD (2000) Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents. *Evolutionary Computation* 8(1):1–29, DOI 10.1162/106365600568086
- Raeisi S, Tapkir P, Tovar A, Mozumder C, Xu S (2019) Multi-Material Topology Optimization for Crashworthiness Using Hybrid Cellular Automata. In: WCX SAE World Congress Experience, Detroit, MI, pp 2019–01–0826, DOI 10.4271/2019-01-0826

- Rahnamayan S, Tizhoosh HR, Salama MMA (2007) A novel population initialization method for accelerating evolutionary algorithms. *Computers & Mathematics with Applications* 53(10):1605–1614, DOI 10.1016/j.camwa.2006.07.013
- Ramnath S, Aulig N, Bujny M, Menzel S, Gandikota I, Horner K (2019) Load Case Preference Patterns based on Parameterized Pareto-Optimal Vehicle Design Concept Optimization. In: *European LS-DYNA Conference 2019*, Koblenz, Germany
- Rao SS (2009) *Engineering Optimization: Theory and Practice*. John Wiley & Sons, Hoboken, NJ, USA
- Raponi E (2017) *Development of Surrogate Modeling Techniques for Level Set Topology Optimization*. Master's thesis, University of Camerino, Camerino, Italy
- Raponi E, Bujny M, Olhofer M, Aulig N, Boria S, Duddeck F (2017) Kriging-guided level set method for crash topology optimization. In: *7th GACM Colloquium on Computational Mechanics for Young Scientists from Academia and Industry*, GACM, Stuttgart, Germany
- Raponi E, Bujny M, Olhofer M, Aulig N, Boria S, Duddeck F (2019a) Kriging-assisted topology optimization of crash structures. *Computer Methods in Applied Mechanics and Engineering* 348:730–752, DOI 10.1016/j.cma.2019.02.002
- Raponi E, Bujny M, Olhofer M, Boria S, Duddeck F (2019b) Hybrid Kriging-assisted Level Set Method for Structural Topology Optimization. In: *11th International Conference on Evolutionary Computation Theory and Applications*, Vienna, Austria
- Ratle A (2001) Kriging as a surrogate fitness landscape in evolutionary optimization. *AI EDAM* 15(1):37–49, DOI 10.1017/S0890060401151024
- Rechenberg I (1971) *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. PhD Thesis, Technical University of Berlin, Berlin, Germany
- Richter A, Botsch M, Menzel S (2015) Evolvability of representations in complex system engineering: A survey. In: *2015 IEEE Congress on Evolutionary Computation (CEC)*, Sendai, Japan, pp 1327–1335, DOI 10.1109/CEC.2015.7257042, iSSN: 1089-778X, 1941-0026
- Richter A, Achenbach J, Menzel S, Botsch M (2016) Evolvability as a quality criterion for linear deformation representations in evolutionary optimization. In: *2016 IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, Canada, pp 901–910, DOI 10.1109/CEC.2016.7743886
- Richter A, Dresselhaus S, Menzel S, Botsch M (2018) Orthogonalization of Linear Representations for Efficient Evolutionary Design Optimization. In: *Genetic and Evolutionary Computation Conference*, Kyoto, Japan, DOI 10.1145/3205455.3205568

- Rieffel J, Saunders F, Nadimpalli S, Zhou H, Hassoun S, Rife J, Trimmer B (2009) Evolving soft robotic locomotion in PhysX. In: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, ACM, Montreal, Québec, Canada, pp 2499–2504, DOI 10.1145/1570256.1570351
- Rothlauf F (2006) Representations for Genetic and Evolutionary Algorithms. Springer, Berlin, Heidelberg, Germany
- Roux W (2016) The LS-TaSC™ Multipoint Method for Constrained Topology Optimization. In: 14th International LS-DYNA Users Conference, Dearborn, MI, USA, p 15
- Rozvany GIN (2008) A critical review of established methods of structural topology optimization. *Structural and Multidisciplinary Optimization* 37(3):217–237, DOI 10.1007/s00158-007-0217-0
- Rozvany GIN (2011) On symmetry and non-uniqueness in exact topology optimization. *Structural and Multidisciplinary Optimization* 43(3):297–317, DOI 10.1007/s00158-010-0564-0
- Rozvany GIN, Zhou M, Birker T (1992) Generalized shape optimization without homogenization. *Structural Optimization* 4(3-4):250–252, DOI 10.1007/BF01742754
- de Ruiter M, van Keulen F (2000) Topology Optimization: Approaching the Material Distribution Problem using a Topological Function Description. In: The Fifth International Conference on Computational Structures Technology, Leuven, Belgium, pp 111–119, DOI 10.4203/ccp.67.1.13
- de Ruiter M, van Keulen F (2004) Topology optimization using a topology description function. *Structural and Multidisciplinary Optimization* 26(6):406–416, DOI 10.1007/s00158-003-0375-7
- Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and Analysis of Computer Experiments. *Statistical Science* 4(4):409–423
- Salway D, Zeguer T (2013) Multi-disciplinary Topology Optimization for Vehicle Bonnet Design. In: 9th European LS-DYNA Conference, Manchester, UK
- Sandgren E (1990) Topological design of structural components using genetic optimization methods. In: Winter Annual Meeting ASME, Dallas, TX, USA, vol 115, pp 31–43
- Sanfeliu A, Fu KS (1983) A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-13(3):353–362, DOI 10.1109/TSMC.1983.6313167

- Schneider D, Schumacher A (2018) Finding Optimized Layouts for Ribs on Surfaces Using the Graph and Heuristic Based Topology Optimization. In: *Advances in Structural and Multidisciplinary Optimization*, Springer International Publishing, Braunschweig, Germany, pp 1615–1628, DOI 10.1007/978-3-319-67988-4_121
- Schneider D, Link S, Schumacher A, Ortmann C (2018) Graph and Heuristic Based Topology Optimization of Crashworthiness Composite Profile Structures Manufactured by Vacuum Infusion and Gluing. In: *Proceeding of the 13th World Congress on Computational Mechanics (WCCM XIII) and 2nd Pan American Congress on Computational Mechanics (PANACM II)*, Braunschweig, Germany, DOI <https://doi.org/10.1007/s00158-012-0872-7>
- Schneider D, Schumacher A, Donhauser T, Huf A, Schmeer S (2019) Flexible Graph Syntax for the Topology Optimization of Crashworthiness Profile Structures Made from Thermoplastic Composites. *Key Engineering Materials* 809:493–499, DOI 10.4028/www.scientific.net/KEM.809.493
- Schumacher A (2005) Parameter-based topology optimization for crashworthiness structures. In: *6th World Congress on Structural and Multidisciplinary Optimisation*, Rio de Janeiro, Brazil
- Schumacher A, Ortmann C (2013) Rule generation for optimal topology changes of crash-loaded structures. In: *10th World Congress on Structural and Multidisciplinary Optimization*, Orlando, FL, United States
- Schwefel HP (1977) *Numerische Optimierung von Computer-Modellen mittels der Evolutionstrategie*. Birkhäuser, Basel, Germany
- Schwefel HP (1987) Collective phenomena in evolutionary systems. In: *Problems of Constancy and Change - The Complementarity of Systems Approaches to Complexity*, Budapest, Hungary
- Schwefel HP (1995) *Evolution and Optimum Seeking*. Wiley VCH, New York, NY, USA
- Sethian JA, Wiegmann A (2000) Structural Boundary Design via Level Set and Immersed Interface Methods. *Journal of Computational Physics* 163(2):489–528, DOI 10.1006/jcph.2000.6581
- Sigmund O (1997) On the Design of Compliant Mechanisms Using Topology Optimization. *Mechanics of Structures and Machines* 25(4):493–524, DOI 10.1080/08905459708945415
- Sigmund O (2011) On the usefulness of non-gradient approaches in topology optimization. *Structural and Multidisciplinary Optimization* 43(5):589–596, DOI 10.1007/s00158-011-0638-7
- Sigmund O (2014) A 99 line topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization* 21(2):120–127, DOI 10.1007/s001580050176
- Sigmund O, Maute K (2013) Topology optimization approaches. *Structural and Multidisciplinary Optimization* 48(6):1031–1055, DOI 10.1007/s00158-013-0978-6

- Sosnovik I, Oseledets I (2017) Neural networks for topology optimization ArXiv:1709.09578
- Stanley KO, Miikkulainen R (2002) Evolving Neural Networks Through Augmenting Topologies. *Evolutionary Computation* 10(2):99–127, DOI 10.1162/106365602320169811
- van Stein B, Wang H, Kowalczyk W, Emmerich M, Bäck T (2017) Cluster-based Kriging Approximation Algorithms for Complexity Reduction ArXiv:1702.01313
- Steiner T, Jin Y, Sendhoff B (2008) A Cellular Model for the Evolutionary Development of Lightweight Material with an Inner Structure. In: 10th Annual Conference on Genetic and Evolutionary Computation, Atlanta, GA, USA, DOI 10.1145/1389095.1389260
- Storn R, Price K (1997) Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization* 11(4):341–359, DOI 10.1023/A:1008202821328
- Sun G, Song X, Baek S, Li Q (2014) Robust optimization of foam-filled thin-walled structure based on sequential Kriging metamodel. *Structural and Multidisciplinary Optimization* 49(6):897–913, DOI 10.1007/s00158-013-1017-3
- Suzuki K, Kikuchi N (1991) A homogenization method for shape and topology optimization. *Computer Methods in Applied Mechanics and Engineering* 93(3):291–318, DOI 10.1016/0045-7825(91)90245-2
- Tang Z, Liu S, Zhang Z (2013) Analysis of energy absorption characteristics of cylindrical multi-cell columns. *Thin-Walled Structures* 62:75–84, DOI 10.1016/j.tws.2012.05.019
- Tavakoli R, Davami P (2009) Optimal riser design in sand casting process with evolutionary topology optimization. *Structural and Multidisciplinary Optimization* 38(2):205–214, DOI 10.1007/s00158-008-0282-z
- Torstenfelt B, Klarbring A (2007) Conceptual optimal design of modular car product families using simultaneous size, shape and topology optimization. *Finite Elements in Analysis and Design* 43(14):1050–1061, DOI 10.1016/j.finel.2007.06.005
- Tovar A (2004) Bone remodeling as a hybrid cellular automaton optimization process. PhD Thesis, University of Notre Dame, Notre Dame, IN, USA
- Tummala Y, Frecker MI, Wissa AA, Jr JEH (2013) Design and optimization of a bend-and-sweep compliant mechanism. *Smart Materials and Structures* 22(9):094,019, DOI 10.1088/0964-1726/22/9/094019
- Ulu E, Zhang R, Yumer ME, Kara LB (2014) A Data-Driven Investigation and Estimation of Optimal Topologies under Variable Loading Configurations. In: *Computational Modeling of Objects*

- Presented in Images. Fundamentals, Methods, and Applications, Springer, Cham, Lecture Notes in Computer Science, pp 387–399, DOI 10.1007/978-3-319-09994-1_38
- Van Miegroet L, Duysinx P (2007) Stress concentration minimization of 2D filets using X-FEM and level set description. *Structural and Multidisciplinary Optimization* 33(4):425–438, DOI 10.1007/s00158-006-0091-1
- Vanderplaats (2019) GENESIS. URL <http://www.vrand.com/products/genesis/>, accessed: 2019-10-03
- Vatanabe SL, Lippi TN, Lima CRd, Paulino GH, Silva EC (2016) Topology optimization with manufacturing constraints: A unified projection-based approach. *Advances in Engineering Software* 100:97–112, DOI 10.1016/j.advengsoft.2016.07.002
- Wang MY, Wang X, Guo D (2003) A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering* 192(1):227–246, DOI 10.1016/S0045-7825(02)00559-5
- Wang S, Wang MY (2006) Radial basis functions and level set method for structural topology optimization. *International Journal for Numerical Methods in Engineering* 65(12):2060–2090, DOI 10.1002/nme.1536
- Wang Z, Hutter F, Zoghi M, Matheson D, De Freitas N (2016) Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research* 55(1):361–387
- Wehrle E (2015) Design optimization of lightweight space-frame structures considering crashworthiness and parameter uncertainty. PhD Thesis, Technical University of Munich, Munich, Germany
- Wehrle EJ, Han YH, Duddeck F (2015) Topology optimization of transient nonlinear structures—A comparative assessment of methods. In: 10th European LS-DYNA Conference, Würzburg, Germany
- Weider K, Schumacher A (2018) A Topology Optimization Scheme for Crash Loaded Structures Using Topological Derivatives. In: *Advances in Structural and Multidisciplinary Optimization*, Springer International Publishing, Braunschweig, Germany, pp 1601–1614, DOI 10.1007/978-3-319-67988-4_120
- Wein F, Dunning P, Norato JA (2019) A Review on Feature-Mapping Methods for Structural Optimization ArXiv:1910.10770
- Wilson B, Cappelleri D, Simpson TW, Frecker M (2001) Efficient Pareto Frontier Exploration using Surrogate Approximations. *Optimization and Engineering* 2(1):31–50, DOI 10.1023/A:1011818803494

- Wold S, Esbensen K, Geladi P (1987) Principal component analysis. *Chemometrics and Intelligent Laboratory Systems* 2(1):37–52, DOI 10.1016/0169-7439(87)80084-9
- Wolfram S (2002) *A New Kind of Science*. Wolfram Media Inc., Champaign, IL, USA
- Xie YM, Steven GP (1997) *Evolutionary Structural Optimization*. Springer-Verlag, London, UK, DOI 10.1007/978-1-4471-0985-3
- Yang XY, Xie YM, Steven GP, Querin OM (1999) Bidirectional Evolutionary Method for Stiffness Optimization. *AIAA Journal* 37(11):1483–1488, DOI 10.2514/2.626
- Yang Z, Tang K, Yao X (2008) Large scale evolutionary optimization using cooperative coevolution. *Information Sciences* 178(15):2985–2999, DOI 10.1016/j.ins.2008.02.017
- Yi SI, Lee JY, Park GJ (2012) Crashworthiness design optimization using equivalent static loads. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 226(1):23–38, DOI 10.1177/0954407011411638
- Yin H, Wen G, Wu X, Qing Q, Hou S (2014) Crashworthiness design of functionally graded foam-filled multi-cell thin-walled structures. *Thin-Walled Structures* 85:142–155, DOI 10.1016/j.tws.2014.08.019
- Yoshimura M, Shimoyama K, Misaka T, Obayashi S (2016) Topology optimization of fluid problems using genetic algorithm assisted by the Kriging model. *International Journal for Numerical Methods in Engineering* 109(4):514–532, DOI 10.1002/nme.5295
- Yu Y, Hur T, Jung J, Jang IG (2018) Deep learning for determining a near-optimal topological design without any iteration. *Structural and Multidisciplinary Optimization* DOI 10.1007/s00158-018-2101-5
- Zeng D (2018) *Enhanced Hybrid Cellular Automata for Crashworthiness Topology Optimization of Thin-walled Structures*. PhD Thesis, Technical University of Munich, Munich, Germany
- Zeng D, Duddeck F (2017) Improved hybrid cellular automata for crashworthiness optimization of thin-walled structures. *Structural and Multidisciplinary Optimization* 56(1):101–115, DOI 10.1007/s00158-017-1650-3
- Zhang H, Wang MY, Chen F, Wang Y, Kumar AS, Fuh JYH (2017a) Design and development of a soft gripper with topology optimization. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, Canada, pp 6239–6244, DOI 10.1109/IROS.2017.8206527
- Zhang W, Zhong W, Guo X (2015) Explicit layout control in optimal design of structural systems with multiple embedding components. *Computer Methods in Applied Mechanics and Engineering* 290:290–313, DOI 10.1016/j.cma.2015.03.007

Zhang W, Yuan J, Zhang J, Guo X (2016) A new topology optimization approach based on Moving Morphable Components (MMC) and the ersatz material model. *Structural and Multidisciplinary Optimization* 53(6):1243–1260, DOI 10.1007/s00158-015-1372-3

Zhang W, Zhou J, Zhu Y, Guo X (2017b) Structural complexity control in topology optimization via moving morphable component (MMC) approach. *Structural and Multidisciplinary Optimization* 56(3):535–552, DOI 10.1007/s00158-017-1736-y

Zhang Y, Chen A, Peng B, Zhou X, Wang D (2019) A deep Convolutional Neural Network for topology optimization with strong generalization ability ArXiv:1901.07761

Zhou M, Rozvany GIN (1991) The COC algorithm, Part II: Topological, geometrical and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering* 89(1–3):309–336, DOI 10.1016/0045-7825(91)90046-9