Article Identifier
DOI: 10.22032/dbt.58937
URN: urn:nbn:de:gbv:ilm1-2023isc-138:9

60th ILMENAU SCIENTIFIC COLLOQUIUM
Technische Universität Ilmenau, 4 – 8 September 2023
URN: urn:nbn:de:gbv:ilm1-2023isc:1

AN AI-DRIVEN DESIGN METHOD AS BASIS FOR TEAMING

Alexander Hasse (1), Lucio Flavio Campanile (1), Stephan Husung (2)
(1) Technische Universität Chemnitz, (2) Technische Universität Ilmenau

ABSTRACT

The product development process could benefit from a synergistic human-machine teaming, potentially shortening product development cycles and improving product performance and sustainability. However, there is a lack of available methods to achieve this goal. A technical product has to satisfy numerous requirements. Due to the variety and complexity of these requirements, the design process is challenging for human engineers. While engineers are supported by various tools (e.g. FEM) for analyzing product properties, tools for computer-aided synthesis of product properties considering the corresponding requirements are still only available in exceptional cases. However, such synthesis capabilities are necessary to qualify a computer-aided tool for productive teaming with engineers. Special methods based on artificial intelligence show a high potential for general computer-aided synthesis methods. This contribution presents an innovative approach in this direction based on topology optimization techniques.

Index Terms - *productive teaming, ai-driven design, topology optimization*

1. INTRODUCTION

Technical products are developed to meet existing or expected needs of different stakeholders [1]. In the development process, various constraints must be taken into account that result from the context of the product's use or from legal and organizational regulations, etc. The requirements and constraints lead to a variety of required product properties. The required properties must be implemented as product characteristics and verified during development (see Figure 1). Not all required properties can be implemented equally well, so compromises must always be made. For the implementation of the individual required properties in product characteristics within the scope of the synthesis as well as for their verification, extensive knowledge (e.g., about physical interrelationships, technological processes, etc.) is required, which must be applied in a problem-specific fashion, e.g., via design guidelines or dimensioning calculations. The synthesis has to be performed down to the detailed level of the product (e.g. down to the detailed design of the components including material specifications or surface finishes). The definition of characteristics for relevant required properties always has an impact on other properties, which may change, or new, previously unknown properties may emerge (so-called emergent properties).

© 2023 by the authors. – Licensee Technische Universität Ilmenau, Deutschland.

This is an Open Access article distributed under the terms of the Creative Commons Attribution ShareAlike-4.0 International License, (https://creativecommons.org/licenses/by-sa/4.0/).

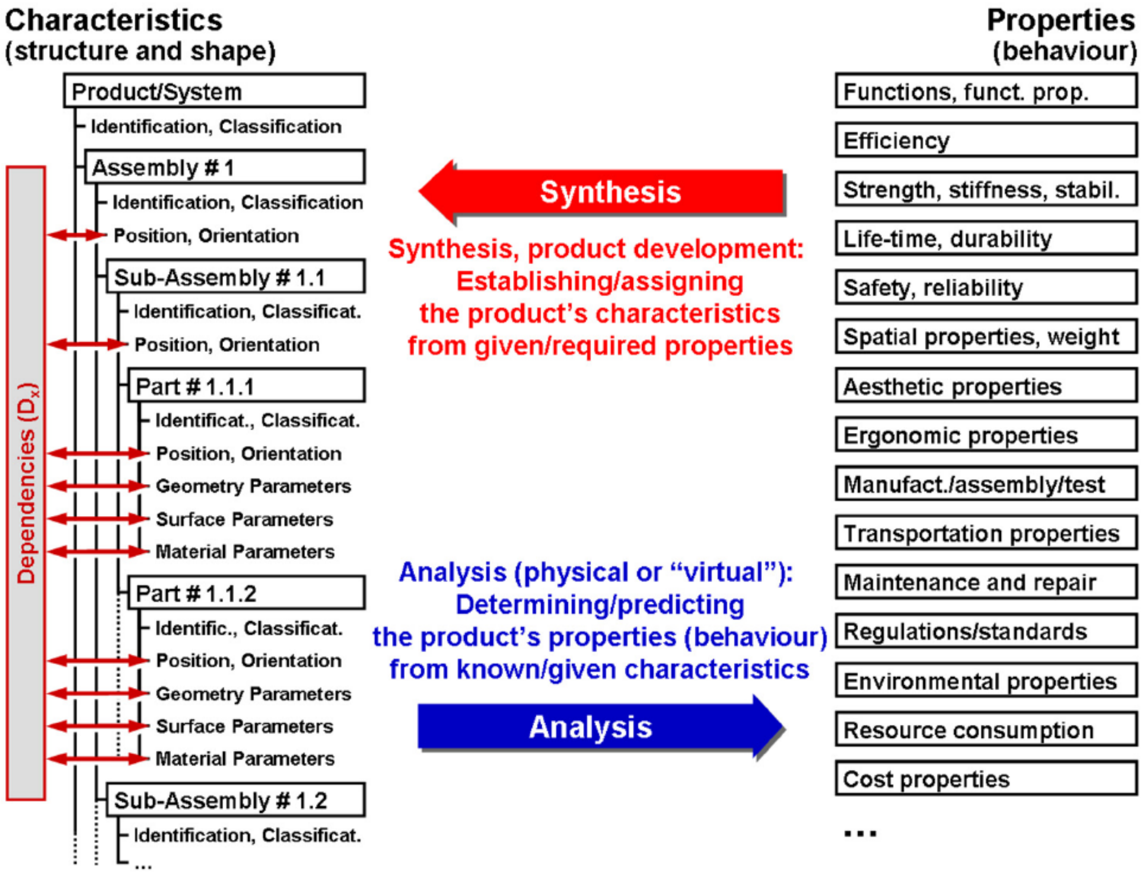

Figure 1 Characteristics and properties, and their two main relationships [2]

In order to meet the challenges described above, problem-solving procedures and methods have been developed for product design that provide engineers with means and make use of people's skills and cognitive abilities (see also Table 1). Especially in product development, flexibility and abstraction of problems or existing solutions and adaptation of goals to changes are essential.

Due to the large number of relevant required properties, the complex relationships between properties and characteristics, and the associated parameters, computer-based tools (here referred to as machines) can demonstrate their strengths in the context of product development. Machines are able to process large amounts of data in a reproducible manner with problem-specific procedures and can recognize relevant patterns in the data (see also Table 1).

Table 1 Complementary strengths of humans and machines (quotation from [3])

Human skills	**Machine skills**
Flexibility & Transfer	Pattern Recognition
Empathy & Creativity	Probabilistic
Annotate Arbitrary Data	Consistency
Common Sense	Speed & Efficiency

Since humans and machines have complementary capabilities, the question of purposeful collaboration between humans and machines is increasingly arising in the context of development. Historically, this cooperation has existed for a very long time. Supporting solutions exist, among others, for information storage and supply [4, 5] or analysis of as-is properties (e.g. for numerical simulations [6, 7] or simulations using surrogate models [8, 9]).

© 2023 by the authors. – Licensee Technische Universität Ilmenau, Deutschland.

Previous approaches usually assume separate tasks in which the human uses the machine specifically for certain purposes. The progress of AI technologies already shows extended possibilities of collaboration between humans and machines [10], so that teaming between humans and machines will become more and more possible in the future. Teaming in this context means that humans and machines act as partners, develop common goals and complementary activities. According to Johnson and Bradshaw [10], teaming is characterized by the three features observability, predictability, and directability (OPD).
The synthesis of mechanical products and in general of technical products is probably the best area in which the idea of teaming can be developed. As opposed to analysis, synthesis is a task of much higher complexity. Its translation into algorithms is made difficult, among others, by the absence of uniqueness in the solution: while the determination of a product's behavior as a function of its layout has, by nature, a unique solution, the uniqueness of the inverse problem, i.e. finding a layout which shows a given behavior, is usually not given. There is a large range of feasible solutions and the choice of the final design includes, in most cases, some degree of arbitrariness. Up to now, formalizing design into algorithmic procedures was made possible by translating the design problem into an optimization problem, which requires a forced uniqueness of the formulation and somehow distorts the original issue. By nature, this not eliminate arbitrariness but just proposes one individual way of proceeding to the solution. Computer-aided analysis is not likely to lead to a constructive interaction between human and machine, since it just represents a tool which passively support the human specialist and cannot engage in a sort of cooperation at eye level. In design, there is a multitude of approaches to identify feasible solutions and choose a suitable representative for final design. Relying on an optimization formulation is just one possible option, which is a typical machine-oriented one. Classical engineering based on intuition and experience is a completely different process, which will be probably never transferred to machines as a whole. A ML-based design procedure like the one presented later on, which evolves on an optimization-related problem definition but does not solve the optimization directly is a third of several possible ways. The coexistence of alternative philosophies which cannot be made equivalent to another and require active cooperation to be harmonized with another (sort of "points of view") and the above mentioned degree of arbitrariness which is inherent to the synthesis task makes this issue to the perfect terrain to develop teaming.

2. STATE OF THE ART

Before presenting the approach discussed in this paper, the historical outline and state of the art of research will be briefly presented.
The wish to automate synthesis existed for a very long time. With the development of computer technology in the last century, the first approaches to automation were developed. Initially, these were mainly analytical designs of individual product features within a defined range (dimensional optimizations). With the progress of computer technology and in particular of computer-aided optimization, the focus of automated design moved to optimization-based approaches. In those approaches, one particular feature of the searched design is chosen as a quantity to be minimized or maximized, while a set of restrictions are to be fulfilled. If more than one quantity is to be maximized or minimized, a multi-objective problem is created, in which the targets are mixed in some way. Optimization-based approaches distinguish between deterministic methods (hill-climbing [11] or gradient methods) and stochastic methods (evolutionary algorithms [12, 13], neural networks, swarm techniques, etc.) [14].
Among optimization-based approaches, topology optimization (TO) deals with the search of optimal distribution of material over a given design domain. In mono-material topology

© 2023 by the authors. – Licensee Technische Universität Ilmenau, Deutschland.

optimization the material of which the structure is to be build is a constant of the problem, and the geometry remains unknown.

In stiffness optimization, the function to be minimized is usually the scalar measure of structural compliance. In addition, a condition on material quantity (degree of filling) must be fulfilled. This material quantity corresponds to the fraction of the maximum possible amount of material The minimization of compliance results in maximizing the stiffness [15, 16]. Typically considered restrictions are the available design domain, the static and kinematic boundary conditions for the regarded load cases as well as strength thresholds.

There are numerous possible approaches to TO [17]. According to the “Solid Isotropic Material with Penalization” (SIMP) approach of Bendsøe [15], a subdivision of the design domain into elements takes place. A factor, yet to be determined (density), scales the contribution of each element to the overall stiffness of the structure. Today, engineers commonly use topology optimization during product development to generate an initial design [18, 19]. However, the traditional iterative calculation method is time-consuming and requires significant computational power. Some newer approaches use artificial intelligence (AI) to speed up the process [20], but they still depend on the availability of pre-optimized data in sufficient quality and quantity.

3. APPROACH

This contribution proposes an alternative AI-based design method that does not rely on pre-optimized data. Instead, an artificial neural network (the predictor) generates designs based on input data such as boundary conditions and degree of filling. During the training phase, the predictor is fed by random input data and learns how to create optimized designs by supervision operated by so-called evaluators. They assess the predictor’s output according to given criteria and provide the error function which controls the learning process. Once training is complete, the AI-assisted design method produces designs that are comparable to those generated by conventional topology optimizers but with a fraction of the computational effort.

4. THE PEN-METHOD

The PEN-Method (PEN is an acronym for Predictor-Evaluator-Network) is an AI-based method devised for synthetizing optimal topologies. In its original form, the method was published in [20]. The underlying optimization problem consists in the search of the geometry of maximum stiffness within a given design domain and under fulfilment of a volume constraint. The geometry is parameterized by subdividing the design domain into small regions (elements) and by scaling the stiffness of each element by an individual factor (density) which can take, for a real structure, the values 0 (no material) or 1 (full material). Intermediate values have no physical meaning and are accepted for algorithmic reason, but made unfavorable by a proper penalization technique. After the optimization is completed, the intermediate values are eliminated during a post-processing step. When performed by a classic optimization procedure, the synthesis process accepts, for a given design domain and a given discretization, the boundary conditions (fixed displacements and given forces) as well as the so-called degree of filling (which percentage of the available volume can be filled by material) and provides the density values of the single elements as results.

An AI-based approach aims to model this input-output relationship by a parametric algorithm (e.g. an artificial neural network – ANN), whose parameter are determined by proper training based on available data. The most obvious way of realizing such an objective is to provide a large number of data sets (input values with the corresponding optimal geometry) by

© 2023 by the authors. – Licensee Technische Universität Ilmenau, Deutschland.

conventional optimization and use them for the training process [9]. Examples of applications of this principle were already present in the literature at the time at which the PEN method was developed. The method adopted an innovative strategy which avoids the cumbersome and somehow problematic task of providing pre-optimized data.

The method is based on the interaction between a trainable ANN named *predictor*, which is in charge of generating the optimal geometries based on data sets, and a proper number of *evaluators*, which are in charge of supervising the predictor's training. Each evaluator assesses the outputs of the predictor with respect to a certain criterion and returns a corresponding scalar value as measure of the criterion's fulfilment. The predictors' outputs are combined in a single value which serves as error function. During the training, the predictor is fed by randomly generated data sets and its parameter are optimized with the error function as objective.

4.1 **Predictor**

The predictor is based on a Deep-Learning (DL) architecture consisting of multiple hidden layers, convolutional layers and output layers. Different activation functions were used in the different layers. the trainable parameter of the predictor are the weights and the bias of the single layers as well as the parameters of the activation functions. In the output layer, the sigmoid function is used, which provides results in the interval (0,1) and therefore makes the predictor's output directly suitable to describe the densities.

4.2 **Compliance evaluator**

The task of the compliance evaluator is the computation of the global mean compliance which provides a scalar measure of the structure's stiffness. It performs this calculation on a FEM-based procedure. The global mean compliance c is defined as follows:

$$c = \mathbf{u}^T \mathbf{k} \mathbf{u} = \mathbf{u}^T \mathbf{f} \tag{1}$$

where $\mathbf{k}$ is the stiffness matrix, $\mathbf{f}$ the force vector and $\mathbf{u}$ the displacement vector.

4.3 **Degree-of-filling evaluator**

The task of this evaluator is to determine the deviation of the degree of filling M_{is} from the target value M_{tar} as follows:

$$M = \left| M_{tar} - M_{is} \right| \tag{2}$$

By considering the filling degree's deviation M in the objective function, the predictor is penalized by the extent that it deviates from the target degree of filling M_{tar}.

4.4 **Filter evaluator**

The filter evaluator searches for checkerboard patterns in the geometry and outputs a scalar value $F \in [0,1]$ that measures the amount and extent of checkerboard patterns detected. Checkerboard patterns consist of alternating high and low density values of the geometry. They

© 2023 by the authors. – Licensee Technische Universität Ilmenau, Deutschland.

are undesirable because they do not reflect the optimal material distribution and are difficult to transfer to real parts. This is realized by discrete convolution.

4.5 Uncertainty evaluator

When calculating the density values of the geometry, the predictor should, as far as possible, focus on the limit values 0 and 1 and penalize intermediate values. The deviation from this goal is expressed by the uncertainty evaluator with the output:

$$P = \frac{1}{N_e} \sum_{i=1}^{N_e} e^{-a\left(x_i - \frac{1}{2}\right)^2} \tag{3}$$

where N_e is the number of elements, x_i the density of the i-th element and a a process parameter.

4.6 Objective function

The quality function

$$f_Q = (\alpha c + 1)(\beta M + 1)(\gamma F + 1)(\delta P + 1) \tag{4}$$

with α, β, γ and δ as process parameters. In order to speed up the process and avoid numerical instabilities, the quality function is computed for a large number of geometries and averaged, which leads to the objective function:

$$J = \frac{1}{b_n} \sum_{i=1}^{b_n} (\alpha c_i + 1)(\beta M_i + 1)(\gamma F_i + 1)(\delta P_i + 1) \tag{5}$$

4.7 Results

The PEN method was implemented in the programming language Python and tested with conventional optimized geometries as benchmarks. These geometries were obtained by the “88 lines of code“ (top88), made available by Andreassen et al. [21]. The training processed approximately 7.6 million randomly generated training data sets.

The results were validated using 100 randomly generated input data sets (validation data), which were not part of the training. The corresponding optimized geometries were conventionally calculated by the top88 algorithm. The results of the comparison are shown in Figure 2.

© 2023 by the authors. – Licensee Technische Universität Ilmenau, Deutschland.

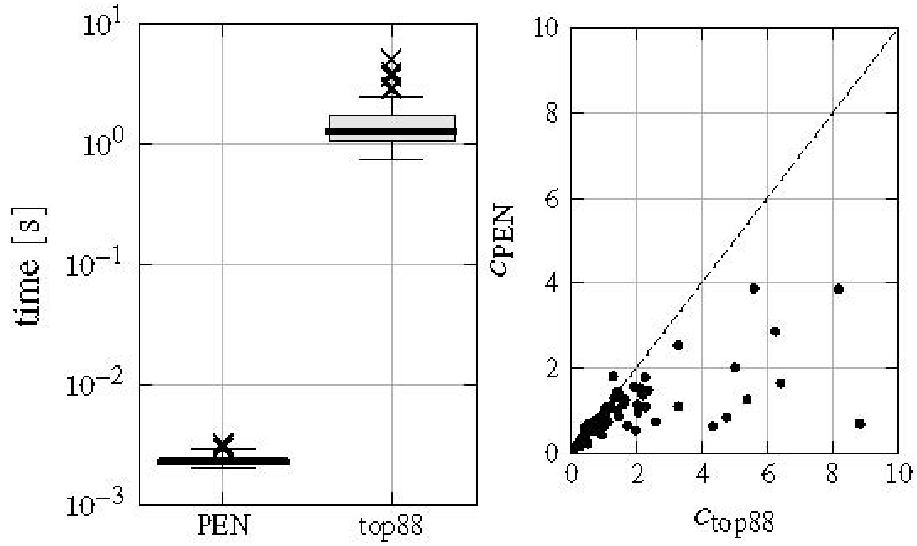

Figure 2 Computing time (left) and compliance (right) comparison [20].

On average, the PEN method delivers similar results to the ones obtained by conventional optimization in about 7.3 ms, while the conventional optimizer requires, on average, 1.9 s (and is, therefore, roughly 259 times slower); see Figure 2 left. It can also be seen that the majority of geometries generated by PEN have a compliance that is close to the geometries generated by top88; see Figure 2 right.

In Figure 2 a comparison between geometries obtained by the PEN method and top88 is shown. Some deviations can be seen (e.g. column four or five). The results can be improved by an enhanced choice of layers or hyperparameters of the predictor and by adapting the objective function. For all sample geometries in Figure 3, the compliance is reported under the geometry diagram.

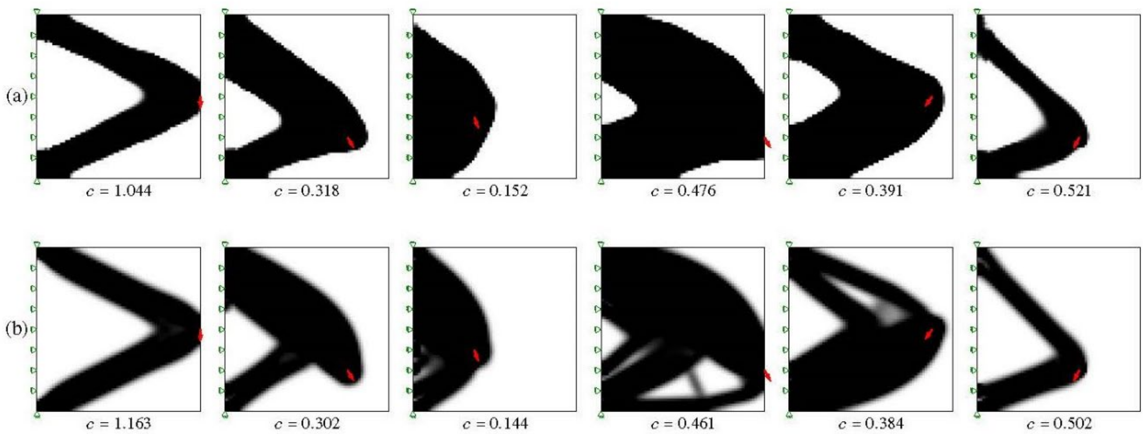

Figure 3 Sample geometries: (a) PEN (b) top-88. [20]

5. CONCLUSION

The PEN method is able to generate topology-optimized geometries without any need of pre-optimized data for the training. The generated geometries are, in most cases, very similar to the results of conventional topology optimization.
The PEN-based optimization is much faster than the conventional one, due to the fact that the computing-intensive part is shifted into the training.
The method was tested for the 2D case up to an output resolution of 64 × 64. This choice is not a limitation of the method and can be improved by using better hardware for training or by high-performance computing.
In cases in which conventional optimization is possible, we expect that the PEN method is able to deliver comparable solutions, like in the analyzed application. However, the PEN method could prove superior in handling applications and optimization problems of higher complexity, such as those involving stress limitations or those dealing with the design of compliant mechanisms. This expectation is related to the fact that no optimized data are needed. All methods that process pre-optimized data suffer from the difficulties encountered by conventional optimization while managing the above-mentioned problems. Because the PEN method works without optimized data, it could also be applied to problems that have no optimal solutions or solutions that are hard to calculate.
The PEN method shows the potential of AI-based solutions for teaming in product development. Further research activities address the development of common and aligned goals as well as the extension of the approaches to other synthesis tasks.

REFERENCES

[1] G. Pahl, W. Beitz, J. Feldhusen, and K.-H. Grote, *Engineering design: A systematic approach,* 3rd ed. London: Springer, 2007.
[2] C. Weber and S. Husung, “Virtualisation of product development/design - seen from Design Theory and Methodology,” in *18th International Conference on Engineering Design (ICED 11)*, Kopenhagen, vol. 2, 2011, pp. 226–235.
[3] D. Dellermann, P. Ebel, M. Söllner, and J. M. Leimeister, “Hybrid Intelligence,” *Bus Inf Syst Eng*, vol. 61, no. 5, pp. 637–643, 2019, doi: 10.1007/s12599-019-00595-2.
[4] T. Brix, U. Döring, and J.-C. Fauroux, “Web-based maintenance and enhancement of the IFToMM dictionary and its use for the information retrieval in DMG-Lib,” in *IFToMM World Congress*, 2015.
[5] P. Kügler, F. Dworschak, B. Schleich, and S. Wartzack, “The evolution of knowledge-based engineering from a design research perspective: Literature review 2012–2021,” *Advanced Engineering Informatics*, vol. 55, p. 101892, 2023, doi: 10.1016/j.aei.2023.101892.
[6] P. Fritzson and V. Engelson, “Modelica - A unified object-oriented language for system modeling and simulation,” in *12th European Conference on Object-Oriented Programming - ECOOP’98*, vol. 1445, G. Goos, J. Hartmanis, J. van Leeuwen, and E. Jul, Eds., Berlin, Heidelberg: Springer, 1998, pp. 67–90.
[7] T. Zerwas *et al.,* “Model Signatures for the Integration of Simulation Models into System Models,” *Systems*, vol. 10, no. 6, p. 199, 2022, doi: 10.3390/systems10060199.
[8] J. Kudela and R. Matousek, “Recent advances and applications of surrogate models for finite element method computations: a review,” *Soft Comput*, vol. 26, no. 24, pp. 13709–13733, 2022, doi: 10.1007/s00500-022-07362-8.

© 2023 by the authors. – Licensee Technische Universität Ilmenau, Deutschland.

[9] M. Steck and S. Husung, “Systematic Optimisation Process for an eBike Drive Unit in a Highly Variable Environment,” *Proc. Des. Soc.*, vol. 3, pp. 3305–3314, 2023, doi: 10.1017/pds.2023.331.
[10] M. Johnson and J. M. Bradshaw, “How Interdependence Explains the World of Teamwork,” in *Engineering Artificially Intelligent Systems* (Lecture Notes in Computer Science), W. F. Lawless, J. Llinas, D. A. Sofge, and R. Mittu, Eds., Cham: Springer International Publishing, 2021, pp. 122–146.
[11] T. Weise, “Global Optimization Algorithms – Theory and Application –,” 2009.
[12] N. Noilublao and S. Bureerat, “Simultaneous topology, shape and sizing optimisation of a three-dimensional slender truss tower using multiobjective evolutionary algorithms,” *Computers & Structures*, vol. 89, 23-24, pp. 2531–2538, 2011, doi: 10.1016/j.compstruc.2011.08.010.
[13] K. Deb, “Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction,” in *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*, L. Wang, A. H. C. Ng, and K. Deb, Eds., London: Springer London, 2011, pp. 3–34.
[14] S. Vajna, C. Weber, K. Zeman, P. Hehenberger, D. Gerhard, and S. Wartzack, *CAx für Ingenieure: Eine praxisbezogene Einführung,* 3rd ed. Berlin, Germany: Springer Vieweg, 2018.
[15] M. P. Bendsøe and O. Sigmund, *Topology Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.
[16] N. Aulig, *Generic Topology Optimization Based on Local State Features*. VDI Verlag, 2017.
[17] O. Sigmund and K. Maute, “Topology optimization approaches,” *Structural and Multidisciplinary Optimization*, vol. 48, no. 6, pp. 1031–1055, 2013, doi: 10.1007/s00158-013-0978-6.
[18] A. Hasse and L. F. Campanile, “Design of compliant mechanisms with selective compliance,” *Smart Mater. Struct.*, vol. 18, no. 11, p. 115016, 2009, doi: 10.1088/0964-1726/18/11/115016.
[19] S. Kirmse, L. F. Campanile, and A. Hasse, “Synthesis of compliant mechanisms with selective compliance – An advanced procedure,” *Mechanism and Machine Theory*, vol. 157, p. 104184, 2021, doi: 10.1016/j.mechmachtheory.2020.104184.
[20] A. Halle, L. F. Campanile, and A. Hasse, "An Artificial Intelligence–Assisted Design Method for Topology Optimization without Pre-Optimized Training Data." *Applied Sciences* 11.19 (2021): 9041.
[21] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund, “Efficient topology optimization in MATLAB using 88 lines of code,” *Structural and Multidisciplinary Optimization*, vol. 43, no. 1, pp. 1–16, 2011, doi: 10.1007/s00158-010-0594-7.

CONTACTS

Alexander Hasse	alexander.hasse@mb.tu-chemnitz.de ORCID: https://orcid.org/0000-0002-6737-5152
Lucio Flavio Campanile	flavio.campanile@mb.tu-chemnitz.de ORCID: https://orcid.org/0000-0002-3553-1667
Stephan Husung	stephan.husung@tu-ilmenau.de ORCID: https://orcid.org/0000-0003-0131-5664

© 2023 by the authors. – Licensee Technische Universität Ilmenau, Deutschland.