SoftwareX 27 (2024) 101797

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

MorphoGen: Topology optimization software for Extremely Modular Systems

Piotr Tauzowski, Bartlomiej Blachowski, Ela Zawidzka, Machi Zawidzki *

Institute of Fundamental Technological Research, Polish Academy of Sciences, Pawińskiego 5B, 02-106 Warsaw, Poland

ARTICLE INFO

Keywords:
Stress Constrained Topology Optimization
Extremely Modular System
Object-oriented software architecture
MATLAB-based array programming
First Order Reliability Analysis

ABSTRACT

This paper introduces MorphoGen — an integrated reliability-based topology optimization and nonlinear finite element analysis system for 2D and 3D domains. The system's key innovation is its seamless prototyping of scientific formulations for computational problems in topology optimization. Its layered and object-oriented architecture, based on the template method design pattern, facilitates effortless modifications of algorithms and the introduction of new types of finite elements, materials, and analyses. MorphoGen also offers flexible handling of objective functions and constraints during topological optimization, enhancing its adaptability. It empowers researchers and practitioners to explore a wide range of engineering challenges, fostering a deeper understanding of complex structural behaviors and efficient design solutions.

There are many topology optimization software and open source codes, especially based on the classical SIMP method. Unlike these codes our package is freely distributed among users and since it is distributed on the MIT licence, which allows for its easy modification depending on the particular needs of the users. For this purpose, we use the topology optimization algorithm proposed for the first time in our previous paper (Blachowski et al., 2020). The algorithm is based on a fully stress design-based optimality criteria and can be applied for topology optimization of either linearly elastic and elastoplastic structures. Additionally, the novelty of the proposed system is related to its ability of solving optimal topology under various constraints such as displacement, stresses and fatigue in both deterministic and probabilistic cases.

Another application are modular structures, which reduce design complexity and manufacturing costs as well as rapid reconfiguration. However, in the realm of structural optimization, modular systems are more challenging due to various: modes of operation of the modules and the stresses configurations. Moreover, this area of research is dramatically less explored.

Thus the effectiveness of MorphoGen for structural engineering is demonstrated with examples of topological shape optimization of two Extremely Modular Systems: a planar robotic manipulator Arm-Z and spatial free-form ramp Truss-Z.

Code metadata

Current code version	1.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-23-00754
Permanent link to Reproducible Capsule	
Legal Code License	MIT License
Code versioning system used	git
Software code languages, tools, and services used	MATLAB
Compilation requirements, operating environments & dependencies	MATLAB 2023a
If available Link to developer documentation/manual	
Support email for questions	ptauzow@ippt.pan.pl

* Corresponding author.
E-mail address: zawidzki@ippt.pan.pl (Machi Zawidzki).

https://doi.org/10.1016/j.softx.2024.101797
Received 8 November 2023; Received in revised form 23 April 2024; Accepted 12 June 2024

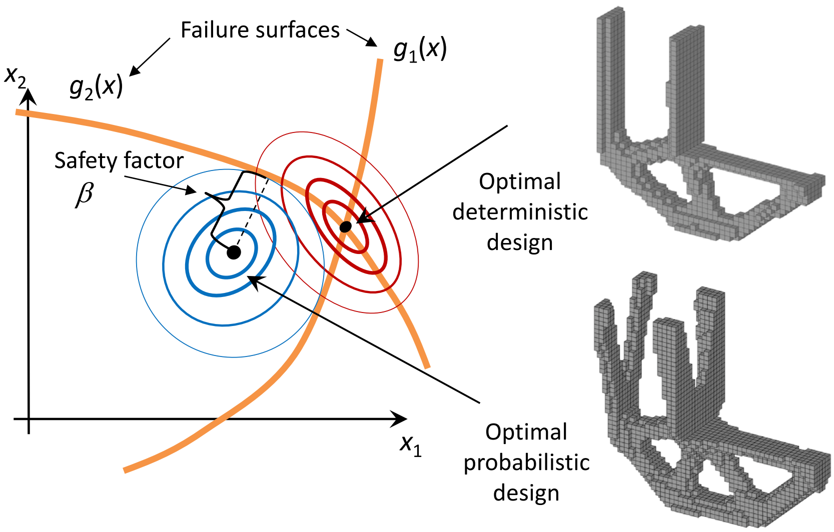

Fig. 1. Reliability versus deterministic design optimization implemented in MorphoGen software.

1. Introduction

1.1. Motivation and significance

Optimal design is a fundamental challenge in engineering, and topology optimization plays a pivotal role in achieving it. This process entails systematic reduction of material within design space while preserving essential structural characteristics, such as serviceability limits and stress requirements. Moreover, topological optimization can encompass complex factors like failure probability.

The pursuit of economical, aesthetically pleasing, robust, and reliable designs has driven extensive research in topological optimization. To facilitate the development of novel optimization formulations and expedite their validation, the need for versatile software tools is evident. These tools should offer flexible architecture that permits comprehensive modifications, thereby enabling efficient numerical testing of innovative approaches. One such instrumental approach is the use of object-oriented programming (see Fig. 1). Reliability-based computational morphogenesis is an approach to find optimal topology of a structure in presence of uncertainty related to either material properties or loading conditions.

MorphoGen enables potential users to easily formulate, solve and present in graphical way results of reliability-based topology optimization, formulated as follows:

$$\begin{array}{ll} \text{find} & \boldsymbol{\rho} \\ \text{min} & F_{obj}(\boldsymbol{\rho}) \\ \text{subject to} & F_g(g_i(\boldsymbol{\rho},\mathbf{x})) \geq 0 \quad i = 1,2,\ldots,N_c \\ & \rho_{\min} \leq \boldsymbol{\rho} \leq \rho_{\max}, \end{array} \tag{1}$$

where $\boldsymbol{\rho}$ is density vector, F_{obj} is an objective function (e.g. volume fraction or compliance), g_i is ith constraints function (e.g. compliance, volume fraction or number of low-cycle fatigue), N_c is number of constraints functions. Additional function F_g is formulated as follows : $F_g = \Pr[g_i(\boldsymbol{\rho},\mathbf{x}) \geq 0] - \Phi(\beta_i^t) \quad i = 1,2,\ldots,N_c$. Pr[.] is probability of failure, $g_i(.)$ is in this case ith performance function, $\mathbf{x}$ is vector of random variables, $\Phi(.)$ is standard normal cumulative distribution function, β_i^t is the target reliability index and $\rho_{\min} = 10^{-3}, \rho_{\max} = 1.0$ are box constraints.

Efficient implementation of topological optimization requires appropriate algorithms. Access to packages equipped with pre-built elements allows for relatively rapid and efficient prototyping of innovative solutions.

MorphoGenintegrates data structures representing classes, algorithms for topological optimization and classes tailored for the finite element method, which assess the structural stress distribution. While our primary focus is on addressing static mechanical problems, our object-oriented architecture ensures adaptability to diverse physical problems. This means that the system can be extended seamlessly to handle other physical domains, such as heat flow or magnetic field distribution.

MorphoGenprovides a platform for rapid prototyping and exploration of new topological optimization concepts. It has been validated through extensive testing and have already found application (please refer to benchmark results presented below).

The preliminary results have been presented in [1], where the reliability based constraints were applied by using *First Order Reliability Method*. Ref. [2] deals with topology optimization of elastoplastic bodies with yield stress as a natural constraints executed by return mapping algorithm.

MorphoGenprovides a user-friendly graphical interface in MATLAB environment, which allows to define: initial design space, structural constraints, and set optimization objectives. Users can also select from a range of predefined material properties and load conditions or customize them according to their specific requirements. The mesh resolution can be easily adjusted, which enables the use of sparse meshes in the initial project phase of creating the model and selecting parameters. Calculations speed is crucial during this phase, especially when the analysis is performed multiple times. Once the model is ready, the user can then set the desired target resolution.

1.2. Related work

Modern numerical methods for finding optimal topology commenced in 1988 [3]. The next milestone was the 99-line Matlab code presented in [4], which certainly promoted the topology optimization methodology among scientists and engineers. The 88 line version of that code presented later in Ref. [5] was a substantial computational speedup. Other codes have been developed involving extension to 3D problems [6,7], material design [8], level–set parametrizations [9], and use of advanced discretization techniques [10]. The latest version of the 99 line code has been published in [11]. An interesting approach was presented by the authors [12] in their package IbIPP, in which the data for the topology optimization task are read from the graphic file, the geometry of the design domain, the boundary conditions as well as load loads.

Modularity provides the benefits of mass-production and re-configurability and better quality control [13]. Moreover it enables a trade-off between the computational performance of a periodic unit cell and the efficacy of non-uniform designs in multi-scale material optimization [14].

In order to demonstrate the effectiveness of our optimization approach, we chose relatively unexplored area of modular structural design, which poses specific challenges:

- Spatial configurations of modules are usually unknown beforehand,
- The irregular module geometry often leads to geometric diversity of the global structures in result.
- Global geometries are often free-form and can be determined during construction on site.

In the literature there is only a limited number of publications on structural optimization of modular constructions.

A typical area of modularity is industrial product design.

In the automotive industry, [15] presents families of related products, where certain modules are shared between specific products. A proof-of-concept framework is proposed for product optimization and illustrated in a case study of a car frame family that is comprised of the base model and its seven-seat and pickup versions, which all share two frame fragments.

In the aeronautical industry, a typical examples of modular structures are airborne shelves used in transport aircraft. They are widely used owing to their transportability, reconfigurability, low manufacturing and service costs. Their topology optimization with multiple assemblies that consist of repeated standard modules and optional reinforcements are presented in [16].

Another natural area for modular design is the construction, in particular — bridge industry. For example, spatial orientation and topology optimization of modular trusses is presented in [17]. However, authors bypass there discrete aspect optimization due to modularity by encoding the spatial orientation of the modules using continuous variables. Moreover, additional constraints are imposed in order to ensure the geometric compatibility of converged solution.

Ref. [18] presents the concept of designing modular bridges including layout optimization and component re-usability. Authors simultaneously optimize the internal topology and in-place spatial orientation of rectangular truss panels comprising girder-type modular bridges.

In more recent paper [14], a modular-topology optimization using Wang tilings [19] has been proposed for truss structures. The approach presented there is two-level, and facilitates simultaneous optimization of the topologies and placement of sixteen types of truss modules.

A heuristic sequential framework for designing modular structures and mechanisms with free material design and clustering also based on Wang tiling has been presented in [20].

Related studies of topology optimization consider *periodic* structures, e.g. [21,22]. The common trait there is the congruency of modules or unit cells. Moreover, the units in periodic structures, unlike general modular structures, are arranged in pre-designed regular grid with topology repetition.

Certain analogues to modular structure optimization can be found in multi-scale (and often multi-physical) material modeling with representative volume elements (RVE) used at the micro-scale level [23–25] or lattice-structured materials with meso-scale cells [26,27]. However, typical optimization objectives express selected properties of homogenized material itself, and usually do not involve global-level properties as typical in structural optimization and structural mechanics. A relatively rare exception is the research reported in [28,29], where the global structure is optimized concurrently with the cell micro-structure using the homogenization approach, and the objectives at the global level are respectively the minimum compliance and the maximum fundamental natural frequency.

Multi-scale concurrent topology optimization for cellular structures based on ordered *SIMP*: solid isotropic material with penalization [4, 30] has been presented in [31].

The *kriging* meta-model has been developed for predictions in mining engineering and geo-statistics [32]. This method has been also applied in the field of the mechanical engineering [33–35]. More recent paper [36] also demonstrates the reduced computational cost in topology optimization. This approach has been recently presented for topology optimization for cellular structures with nonuniform micro-structures in [37].

Extremely Modular Systems introduced in [38] represent particularly interesting types of modularity due to their conceptual purity. Namely, they are comprised of as few types of modules a possible, ideally — just one.

As illustrative examples showing the flexibility and efficiency of MorphoGen, structural optimization of modules for two such systems: Arm-Z - a hyper-redundant robotic manipulator [39] (as an industrial product design scale), and Truss-Z for self-supporting pedestrian ramps [40] (as a larger, construction scale) have been used. Topology optimization of *non-uniform* parameterized lattice micro-structures has been presented in [41].

For more information on research content on topology optimization in the field of extreme modularity see respective subsections: 3.2 Structural Optimization of the Arm-Z module, and 3.3 Structural Optimization of the Truss-Z module.

A very important issue of code efficiency was raised in [42], where the MATLAB implementation prioritizes efficiency through comprehensive vectorization techniques. In that paper the code optimizes computational speed by decomposing the tangent stiffness matrix into three sparse matrices, with only one requiring updating in each Newton iteration. Finally, MorphoGenin addition provides MATLAB codes for 2D and 3D finite element implementations, accommodating von Mises and Drucker–Prager yield criteria, and offers flexibility in selecting finite elements and numerical quadrature rules to further enhance efficiency.

2. Software description

This section gives description of MorphoGenarchitecture. Early attempts its development relied on functor oriented implementation [43]. The current version utilizes hybrid approach involving layered architecture and object-oriented programming.

MorphoGenhas layered software architecture, with the individual layers hierarchy:

- **application layer** — here users find ready-to-use example scripts showcasing the implementation and use of the software.
- **design layer** implements algorithms for topology optimization such as derivative-free approach called stress intensity driven topology optimization, as well as sensitivity-based approach known as Method of Moving Asymptotes.
- **analysis layer** contains algorithms dedicated to solving specific problems for e.g.: linear elastic analysis with sensitivity, elasto-plastic analysis, etc. Additionally, reliability algorithms such as FORM (First-Order Reliability Method), AMV (Advanced Mean Value), and Monte Carlo simulation are stored here.
- **element layer** stores the finite element files, including planar and spatial element definitions.
- **material layer** contains material definitions specific to each finite element type, e.g. isotropic materials are defined for plane stress/strain and solid elements.
- **mesh layer** contains nodes and elements organized into mesh class designed for finite element mesh storage and generation.
- **math layer** provides various mathematical utilities, including Gauss numerical integration and linear equation solver class. They facilitate data manipulation of global matrices and right-hand vectors. Additionally, shape function classes compatible with finite elements are included.

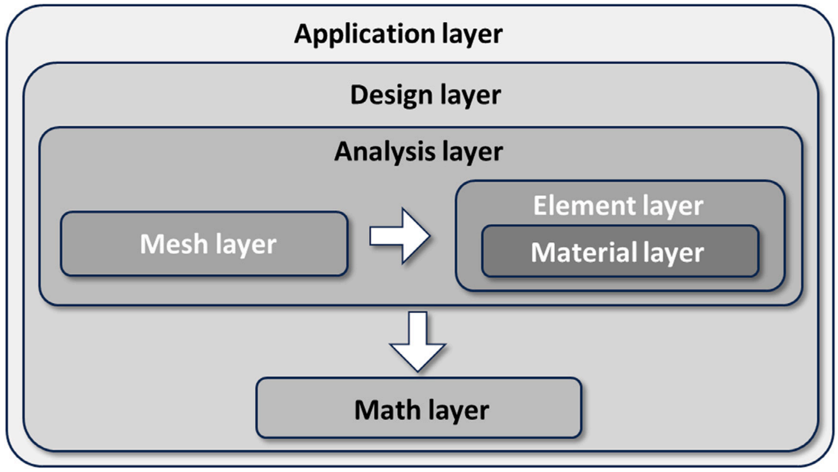

Fig. 2. Layered architecture of MorphoGen system.

(Fig. 2) depicts the layers and a basic cooperation diagram of the objects within each layer. The figure highlights the nesting of layers, showcasing how objects are composed, while arrows depict the cooperation of classes, typically from different layers, not being parts. For example, the 'Topology optimization' object in the Design layer utilizes the 'FEAnalysis' object from the Analysis layer. This, in turn, is composed of objects from the Mesh class and 'FElement' class (Element layer), which can further incorporate a 'Material' class object. Arrows represent the collaboration between objects across different layers.

The 'Mesh' class generates the mesh (geometry), while the 'Element' class handles the physical aspects of the finite element. Elements can have different material properties, hence the cooperation with the 'Material' layer. Finally, all layers interact with the Math layer, which provides essential classes and functions for various tasks across different layers, such as building global matrices, solving linear equations, and performing the MMA optimization procedure used in the Design layer.

These directories provide a comprehensive framework to explore and utilize the software's features and capabilities.

2.1. Software architecture

The MorphoGenarchitecture is composed of several class hierarchies (layers), with functions and abstract classes denoted in *italics*. Specific classes and methods are represented in plain font. Here the key elements of class hierarchy that provide an overview of the system's architecture are highlighted with the focus on selected properties and methods that demonstrate fundamental dependencies. The class diagrams illustrates these dependencies, particularly from the perspective of users interested in implementing their own formulations.

The class hierarchy shown in Fig. 3 represents finite element analysis layer and consists of the base abstract class FEAnalysis.

Nonlinear analysis adds complexity to problem-solving. Our NonlinearAnalysis class, inheriting from FEAnalysis, implements the solve method using the Newton–Raphson algorithm and the 'template method' design pattern. This design makes the algorithm versatile, with specific adaptations achieved by overriding computeTangentMatrix and computeResidualVector methods in derived classes. For instance, in ElastoplasticAnalysis, we have implemented non-linear plasticity analysis, projecting stresses to avoid exceeding yield stress.

Numerous reliability analysis algorithms exist for calculating failure probability. Given that we utilize failure probability as a constraint for structural topological optimization, the efficiency of these methods becomes crucial. One of the fastest methods available is First-Order Reliability Method (FORM). In addition to FORM, we have implemented the Monte Carlo method as a reference method. This allows us to test and verify the accuracy of our solutions by comparing them to results obtained through alternative methods. The topology optimization classes (Fig. 4) represented by a class inheriting from the root topology optimization class, namely TopologyOptimization. In presented approach, the sensitivity information is replaced by the degree of stress intensity, which forms the basis for updating the design variables at each iteration. Individual classes derived from this class represent various constraint formulations. e.g.: StressConstrainedTopologyOptimization which handles reliability constraints, FatigueConstrainedTopologyOptimization which addresses fatigue constraints based on the number of cycles, and ReliabilityConstrainedTopologyOptimization which deals with constraints on probability of failure.

MMA algorithm is the most popular formulation of topological optimization today, therefore we will illustrate the method of implementing topology optimization formulations using an example of a class that implements compliance minimization. To introduce a new formulation into the system, it is sufficient to create a new class inheriting from the GradientBasedTopologyOptimization class. Here we create the MMACompliance class, and override the two abstract methods in the GradientBasedTopologyOptimization class. These methods are: computeObjectiveFunctionAndGradient, which calculates the objective function along with its gradient, and computeConstraintsAndGradient, which calculates the constraint value along with its gradient.

Combining the calculation of the objective function and its gradient into a single function improves the efficiency. Here these values are based on the results of finite element analysis. To calculate the gradient (sensitivity), the analysis results are required.

2.2. Software functionalities

A number of packages are available for structural topological optimization. Particularly relevant package has been described in [42], where the extensive utilization of MATLAB vectorization techniques resulted in substantial computational efficiency. However, this acceleration comes at the cost of code clarity and readability compared to the approach presented here.

The integration of our original Finite Element Method (FEM) solver offers several advantages. It improves problem-solving efficiency, e.g. in the case of a regular mesh, the stiffness matrix can be computed just once for the entire topology optimization process and then aggregated by multiplying it by the density (design variable) for a given finite element. We also considered using the finite element method implemented in MATLAB's Partial Differential Equation (PDE) Toolbox package. However, it has certain drawbacks, such as the absence of rectangular

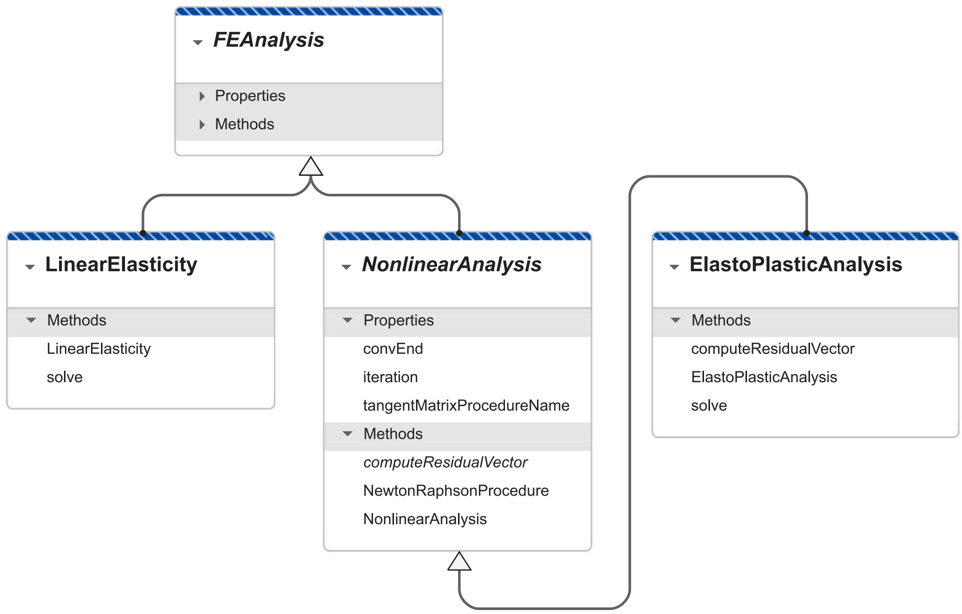

Fig. 3. Class hierarchy for finite element analysis.

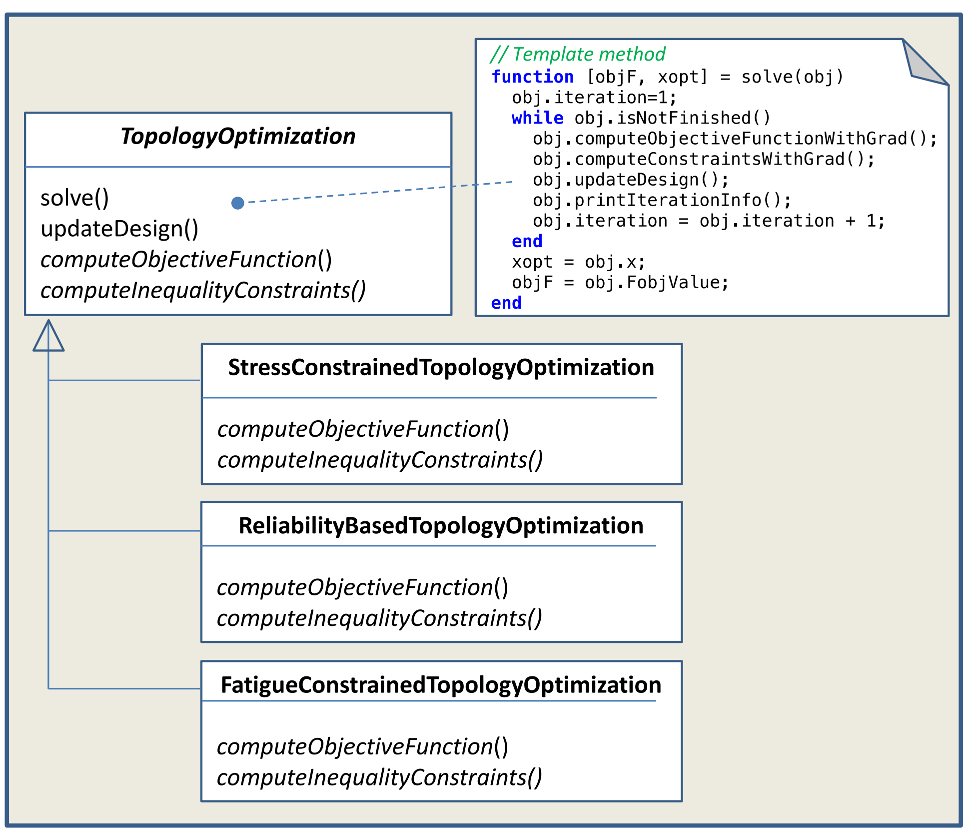

Fig. 4. Template method design pattern applied in MorphoGen.

and cubic elements, which are fundamental for addressing topology optimization problems. Furthermore, comprehensive documentation for all elements of the PDE Toolbox is lacking, making it more of a "black box" solution geared toward problem-solving rather than serving as a versatile framework for users' extensions. Most importantly, it has been indicated by numerical tests, that MorphoGenis more efficient than the MATLAB PDE Toolbox. The effectiveness of our proposed method is inferred from the observed performance metrics in Table 1. Despite the larger memory usage, the significant reduction in execution time underscores the efficiency gains achieved through our approach. Our effective solution can be attributed to profiling conducted using the existing MATLAB profiler and the programming style that maximizes code vectorization.

MATLAB vectorization, characterized by its array programming style, offers notable advantages in computational efficiency and execution speed. By leveraging vectorized operations, MATLAB can perform

Table 1
Comparison of FEM execution times and memory use.

No	Mesh resolution	DOFs	Execution time [s]/Memory usage [GB]		
			PDE toolbox	Vectorized code [42]	Proposed study
1	25 × 10 × 10	9.438	0.08/0.2	0.04/0.2	0.04/0.7
2	50 × 20 × 20	67.473	0.75/0.5	0.32/0.7	0.27/1.2
3	75 × 30 × 30	219.108	3.08/0.8	1.79/3.9	1.05/3.1
4	100 × 40 × 40	509.343	8.32/1.7	5.28/5.8	2.55/5.4
5	125 × 50 × 50	983.178	18.47/3.8	8.56/11.2	4.83/12.0
6	150 × 60 × 60	1.685.613	33.78/5.0	14.85/20.0	8.38/18.4
7	175 × 70 × 70	2.661.648	55.88/7.7	20.04/28.0	13.42/30.0
8	200 × 80 × 80	3.956.283	91.60/11.2	36.43/41.0	20.27/45.8
9	225 × 90 × 90	5.614.518	136.16/17.3	57.61/59.3	28.40/66.2
10	250 × 100 × 100	7.681.353	187.70/28.0	77.00/79.6	39.44/88.0

computations on entire arrays at once, leading to optimized processing and reduced execution times compared to traditional iterative approaches. However, it is essential to recognize that this efficiency gain may come at the expense of increased memory consumption—a well-known trade-off in software optimization. Vectorized operations often necessitate the creation of temporary arrays or intermediate variables, resulting in larger memory footprints, particularly for complex computations or large datasets. For example in our case is the assembly of a stiffness matrix, where storing element matrices of all elements in memory allows us to allocate a sparse global matrix with a single MATLAB command. The entire process of assembling the global matrix is performed efficiently using low-level sparse matrix procedures.

Another significant advantage of our approach is the simplicity which enables users to intuitively formulate the optimization problems. This ease of use is attributed to the software's straightforwardness and the efficiency of the object-oriented approach in describing complexity of a given problem.

2.3. API for reliability-based computational morphogenesis

Listing below presents Application Programming Interface (API) for topology optimization code implementing the compliance minimization task with the volume fraction constraint, as described above (see Fig. 5).

3. Illustrative examples

3.1. Iterative nature of topology optimization

While a detailed explanation of our algorithm falls outside the scope of this paper (having been covered extensively elsewhere, e.g., [1,2,43,44]), we can gain insight into its operation by examining the convergence of key values like the objective function and constraint. Fig. 6 depicts the evolution of the objective function, which in our case represents volume fraction. The second curve shows the evolution of the optimization constraint, normalized displacement.

During the iterative process, the structure evolves from an initially filled workspace (problem domain, green point on the constraint curve). The algorithm progressively reduces material density in the least stressed elements, ultimately reaching the final topology (red point on the constraint curve). This process continues until the constraint becomes active, signifying a limitation on the design.

Fig. 6 also showcases several intermediate topologies generated during the iterations. Two particularly significant points are highlighted: (1) the first instance where the diagonal lacing loses continuity, and (2) the lancing disappearing and the strut is formed. As evident in the constraint variability graph, these events correspond to local extrema on constraint curve. Such extrema, along with sudden jumps in the constraint function, are commonly observed in our topology optimization algorithm, due to significant structural changes between iterations. These changes can manifest as the loss of continuity in load-bearing parts of topology, the formation of holes within the structure, or other significant topological modifications. While our example primarily focuses on the loss of continuity, these additional phenomena can at general also occur.

3.2. Structural optimization of the Arm-Z module

In principle, Extremely Modular Systems [38] are comprised of as few types of modules as possible and allow for creation of free-form objects.

Arm-Z [39] is the concept of an Extremely Modular robotic manipulator. In Arm-Z each module is identical, and the links between modules have exactly one degree of freedom (1-DoF) — the relative twist. Arm-Z manipulator has as many degrees of freedom as the number of modules less of one. This redundancy allows it to perform complicated movements, but also may improve its robustness and fault tolerance [45]. Despite its extreme simplicity, the control of Arm-Z is relatively difficult [39]. Here we use an extremely simple planar Arm-Z composed of 24 identical modules introduced in [46], and shown in Fig. 7.1. Each module has only two discrete positions: left (0) or right (1). Inspection is a typical task for such a manipulator. Fig. 7.2 shows an example environment with five points (A . . . E) to be visited by the manipulator gliding on a flat surface. Fig. 7.3 shows the envelope of all Arm-Z positions during the inspection experiment.

Topology optimization procedure is based on formulation (1). An iterative algorithm was used based on the removal of the most intensive elements until a volume equal to half of the initial volume was achieved. The stress intensity is based on von Mises stress divided by maximal stress found in the structure. For detailed description of the algorithm see [2]. Two types of loads were applied to the segment: a tensile force acting at the connection with adjacent member, and the pressure of the adjacent modules, as shown in Fig. 7.4. The optimization of the pin as a part of the joint needs a separate formulation. Moreover, the potential improvement by mass reduction of the pin is relatively small, thus it has been excluded from the computations. To reduce the computations, the finite element model takes advantage of symmetry of the optimized element.

Fig. 7.5 shows an example of a 6-module Arm-Z comprised of optimized modules.

3.3. Structural optimization of the Truss-Z module

Truss-Z is composed of frame-truss modules (and their reflection, rotation and combination of both), which allows to create free-form ramps, as illustrated in Fig. 8.1.

The optimal assembly of Truss-Z is a hard combinatorial problem. Various deterministic and meta-heuristic computational methods have been applied for designing Truss-Z paths, e.g.: backtracking [40] and evolutionary algorithms [47]. Image processing methods parallelized with GPU have been implemented for effective multi-objective path optimization in [48]. For optimization of multi-branch Truss-Z layouts, evolutionary algorithms have been applied in [49], and a graph-theoretic exhaustive search has been presented in [50].

The first attempt for structural optimization of Truss-Z module has been documented in [51], where a particular outer geometry of the module has been arbitrarily assumed, and the problem of sizing optimization of the module members has been considered. On the other

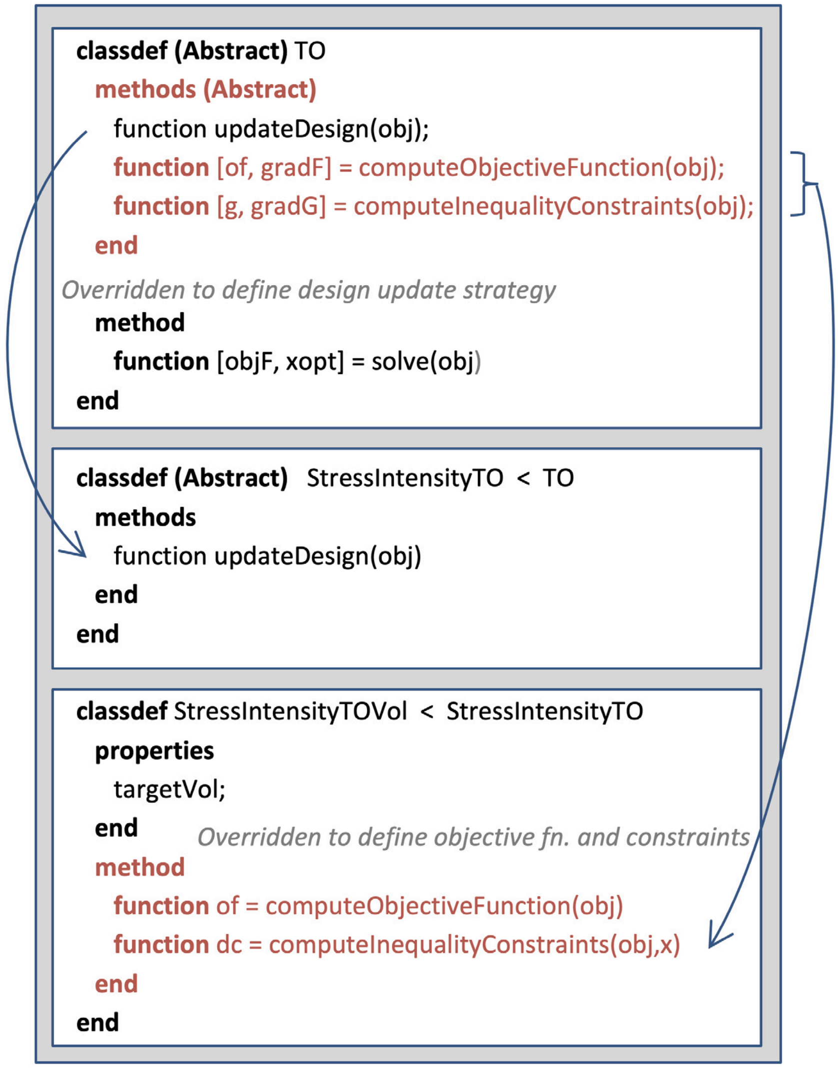

Fig. 5. Illustration of API class for topology optimization.

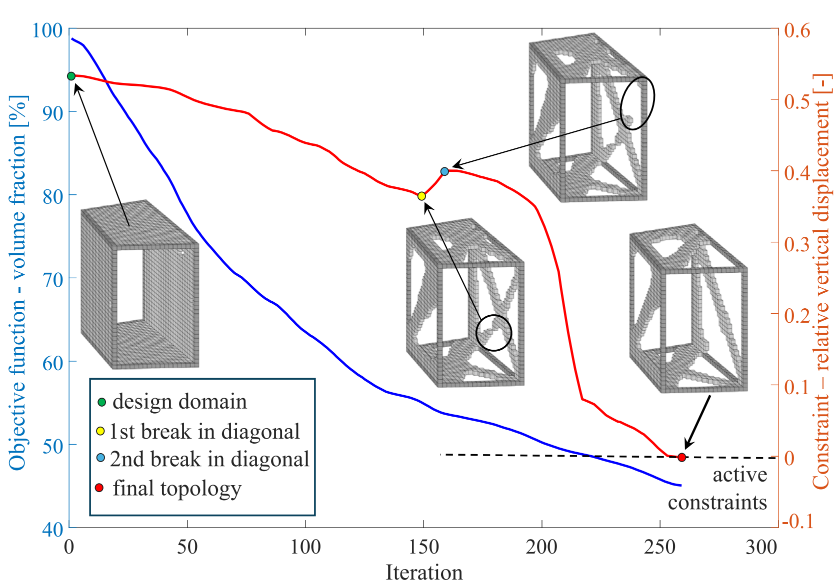

Fig. 6. Evolution of the objective function and constraint.

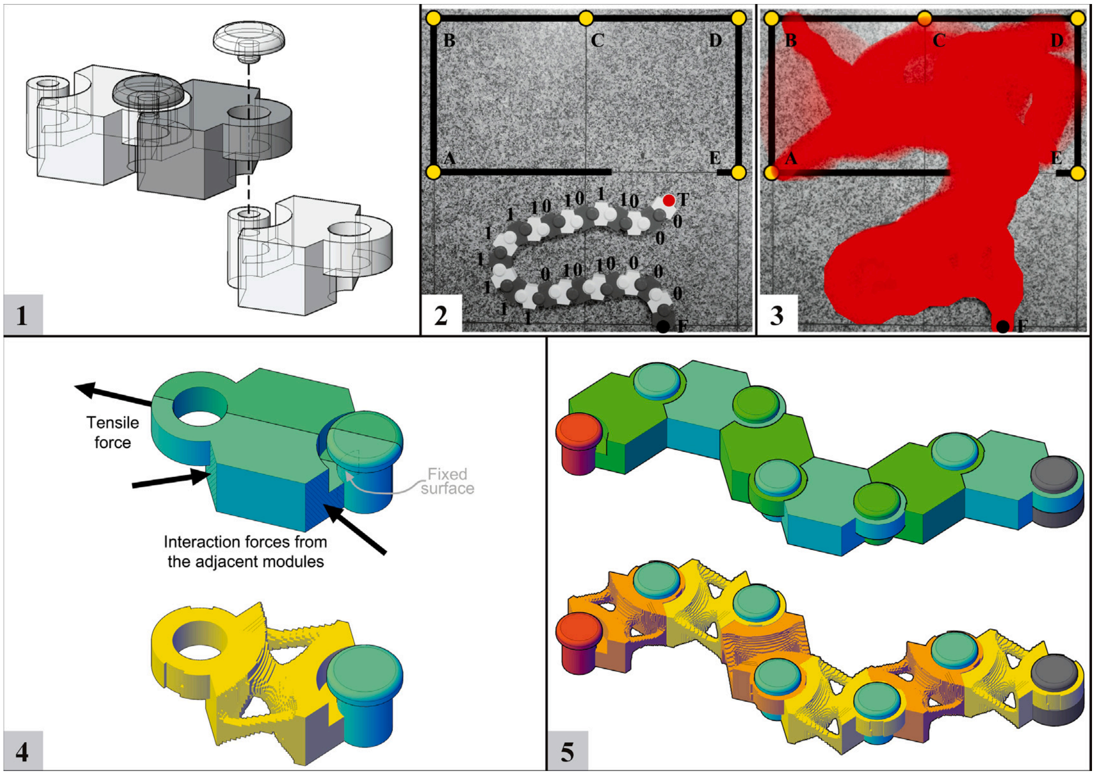

Fig. 7. 1. Two modules linked together and exploded view showing the connection. 2. The inspection experiment setup. 3. Red indicates the envelope of all positions of the manipulator during experiment. 4. Top: boundary conditions and loads imposed on the module, with emphasized symmetry; bottom: optimized module with pin (shown in cyan) excluded from computations. 5. Top: 6-module planar Arm-Z. The tip and fixed base are shown in red and black, respectively; bottom: the optimized Arm-Z. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

hand, the optimal design of the universal module not for a specific spatial arrangement, but rather for all possible scenarios has been proposed in [52].

Here, we demonstrate the usability and efficiency of MorphoGen by conceptual structural optimization of a Truss-Z module subjected to various typical loads: bending, shear and torsion. For a user-friendly simple simulation of a live load by pedestrian crowd see [53].

As Fig. 8.1. indicates, the loads for each module vary, and since the structure is composed of identical units — each module must withstand the globally worst condition. In other words, the basic module is the most strained unit in the entire structure. Similarly to the previous example, an algorithm was used based on the element's effort. The final volume of the topology was set to 40% of the initial volume. The design domain is given by the general envelope of the module, shown in Fig. 8.2. To ensure the rigidity of the structure, the extreme elements form a kind of a cubic frame, which is a permanent part of the structure, not subject to the topological element removal algorithm.

Fig. 8.3 shows the results of Truss-Z module optimization for particular loads and the combined load. Fig. 8.4 shows the results of Truss-Z module optimization for additional three combinations of pairs of loads: bending + shear, bending + torsion and shear + torsion.

4. Impact

MorphoGensoftware is designed as a versatile and flexible tool for solving structural topology optimization problems. Its layered and object-oriented architecture allows adding new functionalities related to design of multi-scale structures and robust design of structures subjected to random loading conditions.

MorphoGenopens new possibilities for solving important scientific problems, e.g. topology optimization in low-cycle fatigue design recently submitted by the authors [44]. Moreover, MorphoGenis able to solve various problem of structural topology optimization without using external libraries and packages.

Users can tackle variety of problems using MorphoGen, ranging from classical topology optimization within the elastic regime with constraints solely on displacements to more intricate reliability-based topology optimization in the plastic regime, incorporating constraints on both stresses and displacements.

MorphoGenis actively promoted at prestigious conferences such as Civil-Comp-Opti (*International Conference on Soft Computing, Machine Learning and Optimization in Civil, Structural and Environmental Engineering*) and CST (*International Conference on Computational Structures Technology*). Moreover, we offer MorphoGenas a numerical tool for courses on topology optimization for Master's and Doctoral students.

Finally, MorphoGenserves as an excellent foundation for development of commercial software. It presents innovative approach well suited for additive manufacturing technology, meeting new and exciting areas in structural engineering.

5. Conclusions

This paper presents MorphoGen— a novel integrated system for reliability-based topology optimization and nonlinear finite element analysis, which development has been inspired by the uncharted territory of topological optimization of Extremely Modular Systems. The layered and object-oriented architecture of MorphoGen, based on the template method design pattern enables intuitive experimentation of scientific formulations for computational problems in topology optimization. MATLAB-profiled code demonstrates very good computational performance and achieves competitive results compared to existing solutions for structural topology optimization.

Flexibility of MorphoGenin handling objective functions and constraints results in unprecedented adaptability, enabling researchers and practitioners to address a wide range of engineering challenges.

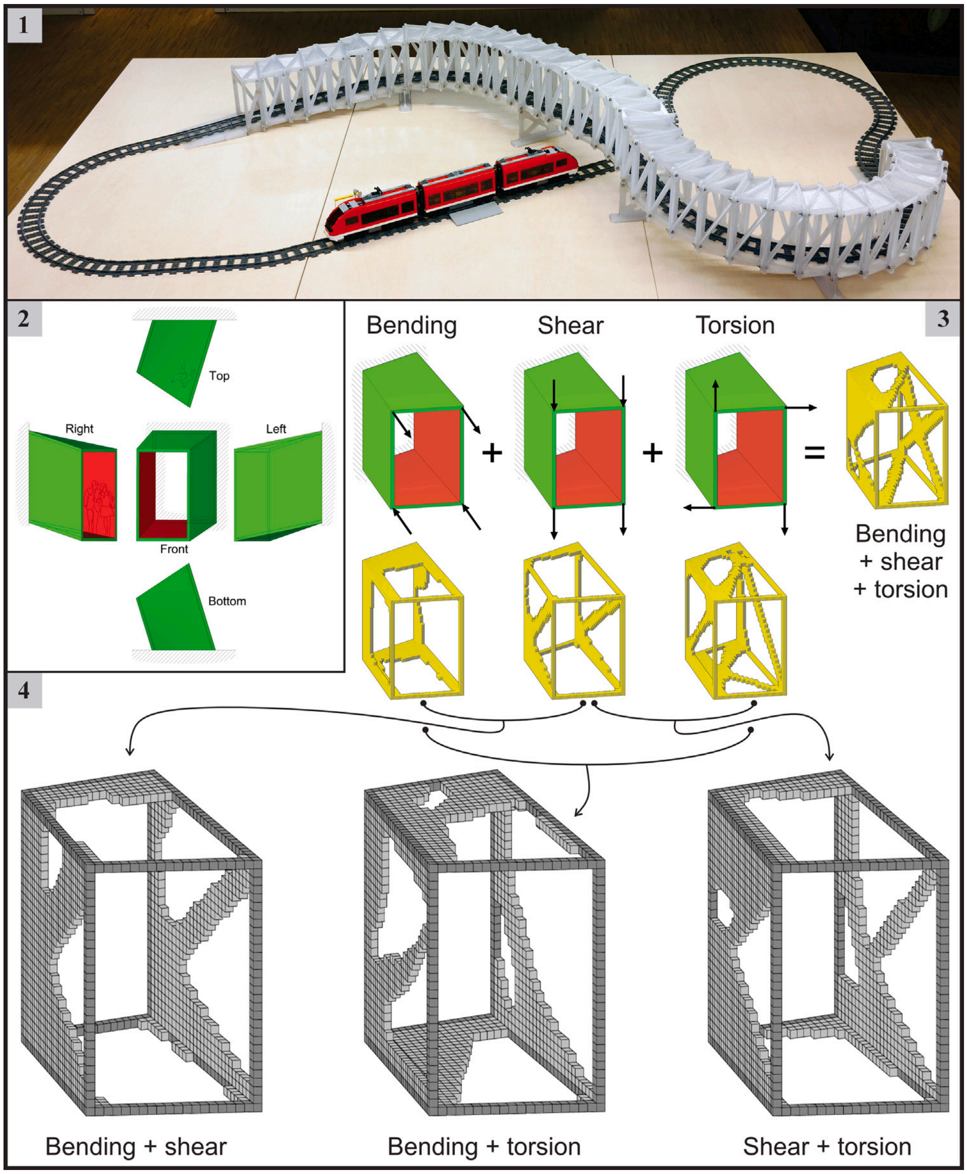

Fig. 8. 1. A free-form railway LEGO® bridge as an example of a Truss-Z structure. 2. Truss-Z module envelope in parallel projections showing the static scheme for optimization. 3. Top: three load schemes: bending, shear and torsion; bottom: the optimized modules for corresponding loads and the result of optimization for the combination of these three loads. 4. Three combinations of pairs of loads.

As we look to the future, our system provides a strong foundation for further exploration and innovation in the fields of topology optimization and finite element analysis. It opens doors to new research questions and practical applications that can facilitate scientific research in the field of structural analysis and engineering design.

While we acknowledge certain limitations in our study, we believe that the contributions made in this work have potential to advance both research and industry practices. We look forward to further investigations in this exciting and dynamic area of study, where the boundaries of structural engineering continue to expand.

CRediT authorship contribution statement

Piotr Tauzowski: Software. **Bartlomiej Blachowski:** Methodology. **Ela Zawidzka:** Investigation. **Machi Zawidzki:** Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Machi Zawidzki reports financial support was provided by National Science Centre Poland. Ela Zawidzka reports financial support was provided by National Science Centre Poland. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This research is a part of the project titled *Arm-Z: an extremely modular hyper redundant low-cost manipulator – development of control methods and efficiency analysis* and funded by *OPUS 17* research grant No. 2019/33/B/ST8/02791 supported by the National Science Centre, Poland.

References

[1] Tauzowski P, Blachowski B, Lógó J. Topology optimization of elasto-plastic structures under reliability constraints: A first order approach. Comput Struct 2021;243:106406.
[2] Blachowski B, Tauzowski P, Lógó J. Yield limited optimal topology design of elastoplastic structures. Struct Multidiscip Optim 2020;61(5):1953–76.
[3] Bendsøe MP, Kikuchi N. Generating optimal topologies in structural design using a homogenization method. Comput Methods Appl Mech Engrg 1988;71(2):197–224.
[4] Sigmund O. A 99 line topology optimization code written in Matlab. Struct Multidiscip Optim 2001;21:120–7.
[5] Andreassen E, Clausen A, Schevenels M, Lazarov B, Sigmund O. Efficient topology optimization in MATLAB using 88 lines of code. Struct Multidiscip Optim 2011;43(1):1–16.
[6] Liu K, Tovar A. An efficient 3D topology optimization code written in Matlab. Struct Multidiscip Optim 2014;50(6):1175–96.
[7] Amir O, Aage N, Lazarov BS. On multigrid-CG for efficient topology optimization. Struct Multidiscip Optim 2014;49(5):815–29.
[8] Xia L, Breitkopf P. Design of materials using topology optimization and energy-based homogenization approach in matlab. Struct Multidiscip Optim 2015;52(6):1229–41.
[9] Challis VJ. A discrete level-set topology optimization code written in Matlab. Struct Multidiscip Optim 2010;41(3):453–64.
[10] Sanders ED, Pereira A, Aguiló MA, Paulino GH. PolyMat: an efficient matlab code for multi-material topology optimization. Struct Multidiscip Optim 2018;58(6):2727–59.
[11] Ferrari F, Sigmund O. A new generation 99 line Matlab code for compliance topology optimization and its extension to 3D. Struct Multidiscip Optim 2020;62(4):2211–28.
[12] Ibhadode O, Zhang Z, Bonakdar A, Toyserkani E. IbIPP for topology optimization—An Image-based Initialization and Post-Processing code written in MATLAB. SoftwareX 2021;14:100701.
[13] Mikkola JH, Gassmann O. Managing modularity of product architectures: toward an integrated theory. IEEE Trans Eng Manag 2003;50(2):204–18.
[14] Tyburec M, Zeman J, Doškář M, Kružík M, Lepš M. Modular-topology optimization with wang tilings: an application to truss structures. Struct Multidiscip Optim 2021;63(3):1099–117.
[15] Torstenfelt BR, Klarbring A. Structural optimization of modular product families with application to car space frame structures. Struct Multidiscip Optim 2006;32(2):133–40.
[16] Jie W, Tong G, Ming L, Jihong Z, Longlong S, ZHANG W. Topology optimization of modular structures with multiple assemblies and applications to airborne shelves. Chin J Aeronaut 2023.
[17] Tugilimana A, Thrall AP, Descamps B, Coelho RF. Spatial orientation and topology optimization of modular trusses. Struct Multidiscip Optim 2017;55(2):459–76.
[18] Tugilimana A, Thrall AP, Coelho RF. Conceptual design of modular bridges including layout optimization and component reusability. J Bridge Eng 2017;22(11):04017094.
[19] Wang H. Proving theorems by pattern recognition I. Commun ACM 1960;3(4):220–34.
[20] Tyburec M, Doškář M, Zeman J, Kružík M. Modular-topology optimization of structures and mechanisms with free material design and clustering. Comput Methods Appl Mech Engrg 2022;395:114977.
[21] Zhou CW, Lainé JP, Ichchou M, Zine AM. Multi-scale modelling for two-dimensional periodic structures using a combined mode/wave based approach. Comput Struct 2015;154:145–62.
[22] Moses E, Fuchs MB, Ryvkin MB. Topological design of modular structures under arbitrary loading. Struct Multidiscip Optim 2002;24(6):407–17.
[23] Chen N, Yu D, Xia B, Liu J, Ma Z. Microstructural topology optimization of structural-acoustic coupled systems for minimizing sound pressure level. Struct Multidiscip Optim 2017;56(6):1259–70.
[24] Chen Y, Zhou S, Li Q. Multiobjective topology optimization for finite periodic structures. Comput Struct 2010;88(11):806–11.
[25] Wang Y, Luo Z, Zhang N, Qin Q. Topological shape optimization of multifunctional tissue engineering scaffolds with level set method. Struct Multidiscip Optim 2016;54(2):333–47.
[26] Augustyniak M. Old materials – new capabilities: lattice materials in structural mechanics. J Theoret Appl Mech 2018;56(1):213–26.
[27] Messner MC. Optimal lattice-structured materials. J Mech Phys Solids 2016;96:162–83.
[28] Liu L, Yan J, Cheng G. Optimum structure with homogeneous optimum truss-like material. Comput Struct 2008;86(13–14):1417–25.
[29] Niu B, Yan J, Cheng G. Optimum structure with homogeneous optimum cellular material for maximum fundamental frequency. Struct Multidiscip Optim 2008;39(2):115–32.
[30] Bendsøe MP, Sigmund O. Material interpolation schemes in topology optimization. Arch Appl Mech 1999;69:635–54.
[31] Zhang Y, Xiao M, Li H, Gao L, Chu S. Multiscale concurrent topology optimization for cellular structures with multiple microstructures based on ordered SIMP interpolation. Comput Mater Sci 2018;155:74–91.
[32] Simpson TW, Mauery TM, Korte JJ, Mistree F. Kriging models for global approximation in simulation-based multidisciplinary design optimization. AIAA J 2001;39(12):2233–41.
[33] Hamza K, Aly M, Hegazi H. A kriging-interpolated level-set approach for structural topology optimization. J Mech Des 2014;136(1):011008.
[34] Sakata S, Ashida F, Zako M. Structural optimization using Kriging approximation. Comput Methods Appl Mech Engrg 2003;192(7–8):923–39.
[35] Simpson TW, Booker AJ, Ghosh D, Giunta AA, Koch PN, Yang R-J. Approximation methods in multidisciplinary analysis and optimization: a panel discussion. Struct Multidiscip Optim 2004;27:302–13.
[36] Yoshimura M, Shimoyama K, Misaka T, Obayashi S. Topology optimization of fluid problems using genetic algorithm assisted by the Kriging model. Internat J Numer Methods Engrg 2017;109(4):514–32.
[37] Zhang Y, Li H, Xiao M, Gao L, Chu S, Zhang J. Concurrent topology optimization for cellular structures with nonuniform microstructures based on the kriging metamodel. Struct Multidiscip Optim 2019;59:1273–99.
[38] Zawidzki M. Discrete optimization in architecture: Extremely modular systems. Springer; 2017.
[39] Zawidzki M, Szklarski J. Transformations of Arm-Z modular manipulator with Particle Swarm Optimization. Adv Eng Softw 2018;126:147–60.
[40] Zawidzki M, Nishinari K. Modular Truss-Z system for self-supporting skeletal free-form pedestrian networks. Adv Eng Softw 2012;47(1):147–59.
[41] Wang C, Zhu JH, Zhang WH, Li SY, Kong J. Concurrent topology optimization design of structures and non-uniform parameterized lattice microstructures. Struct Multidiscip Optim 2018;58:35–50.
[42] Čermák M, Sysala S, Valdman J. Efficient and flexible MATLAB implementation of 2D and 3D elastoplastic problems. Appl Math Comput 2019;355:595–614.
[43] Tauzowski P, Blachowski B, Lógó J. Functor-oriented topology optimization of elasto-plastic structures. Adv Eng Softw 2019;135:102690.
[44] Tauzowski P, Błachowski B, Lógó J. Optimal topologies considering fatigue with reliability constraint. Adv Eng Softw 2024;189:103590.
[45] Zawidzki M. Deployable Pipe-Z. Acta Astronaut 2016;127:20–30.
[46] Zawidzka E, Zawidzki M. Discrete (two-state) modular hyper-redundant planar manipulator. Comput Assist Methods Eng Sci 2022;29(4):397–407.
[47] Zawidzki M, Nishinari K. Application of evolutionary algorithms for optimum layout of Truss-Z linkage in an environment with obstacles. Adv Eng Softw 2013;65:43–59.
[48] Zawidzki M, Szklarski J. Effective multi-objective discrete optimization of Truss -Z layouts using a GPU. Appl Soft Comput 2018.
[49] Zawidzki M. Optimization of multi-branch Truss-Z based on evolution strategy. Adv Eng Softw 2016;100:113–25.
[50] Zawidzki M. Retrofitting of pedestrian overpass by Truss-Z modular systems using graph-theory approach. Adv Eng Softw 2015;81:41–9.
[51] Zawidzki M, Jankowski Ł. Optimization of modular Truss-Z by minimum-mass design under equivalent stress constraint. Smart Struct Syst 2018;21(6):715–25.
[52] Zawidzki M, Jankowski Ł. Multiobjective optimization of modular structures: weight versus geometric versatility in a Truss-Z system. Comput-Aided Civ Infrastruct Eng 2019;34(11):1026–40.
[53] Zawidzki M, Chraibi M, Nishinari K. Crowd-Z: The user-friendly framework for crowd simulation on an architectural floor plan. Pattern Recognit Lett 2014;44:88–97.