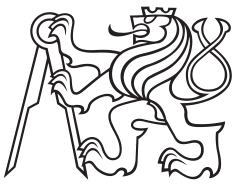
Doctoral Thesis



Czech Technical University in Prague

F1

Faculty of Civil Engineering
Department of Mechanics

Modular-topology optimization of structures and mechanisms

Ing. Marek Tyburec

Supervisor: prof. Ing. Jan Zeman, Ph.D.
Supervisor–specialist: doc. Ing. Matěj Lepš, Ph.D.
Field of study: Physical and Materials Engineering
December 2021

Acknowledgements

First and most importantly, I wish to thank my supervisor, Jan Zeman, for inspiring guidance, scholarly and personal support, and deep understanding. You brought me to the scientific research that I love, and without you, this thesis would not come to life.

Second, I would like to acknowledge my co-supervisor Matěj Lepš and Jan Novák. Only the given opportunities and the trust you had in me let me dive into the amazing world of optimization.

Next, I would like to thank all my friends, colleagues, and co-authors for a wonderful collaboration. Martin Doškář, Martin Kružík, Didier Henrion, Robin Poul, Tomáš Plachý, Stephanie Krueger, Ondřej Rokoš, Jan Havelka, Pavel Trávníček, Vašek Nežerka, Michal Somr, Jiří Němeček, Alexander Brune, Eric Garner, and Weichen Li, thank you for your kindness.

During my studies, I was fortunate to spend few weeks at the University of Birmingham. Hereby, I would like to thank Professor Michal Kočvara for a great experience.

Finally, without the support of my wife and three children, I would be no one. Thank you for always being there for me.

Declaration

PhD candidate: Marek Tyburec

Title: Modular-topology optimization of structures and mechanisms

Herewith, I declare that the enclosed doctoral thesis is my own work, written under the professional guidance of Jan Zeman and the co-supervisor Matěj Lepš. I also certify that all used resources and data are cited and referenced.

The doctoral thesis has been written in connection with the research project: GAČR 19-26143X.

In Prague on .....

Signature .....

Abstract

This thesis establishes modular-topology optimization of structures and mechanisms through a collection of five manuscripts. The modular-topology optimization problem constitutes a bilevel task of optimizing topologies of a set of modules and their spatial arrangement. The first chapter focuses solely on generating module topologies and develops a convex formulation for designing modular truss reinforcement to reduce elastic wall instabilities within a thin-walled composite beam. Subsequently, the author considers a generalization towards bending-resistant structural elements and proposes a global solution method based on the moment-sum-of-squares hierarchy to solve these kinds of problems for the first time. The third chapter introduces a concurrent modular-topology optimization scheme coupling metaheuristics and convex mathematical programming to minimize compliance of structures discretized with reusable truss modules encoded via the Wang tiling formalism to manage the module connectivity. The fourth chapter extends the scheme to compliant mechanisms and modules with continuum description of topologies. In addition, the chapter introduces a sequential approach, which relies on free material optimization and a novel hierarchical clustering algorithm to produce efficient assembly plans heuristically. The approach is illustrated with compliant mechanisms made of reusable modules. The final chapter investigates the problem of generating valid Wang tiling assembly plans alone and it introduces several exact and heuristic methods for its control and solution.

Keywords: modular-topology optimization, semidefinite programming, global optimality, reusability, Wang tiling

Supervisor: prof. Ing. Jan Zeman, Ph.D.

Abstrakt

Tato práce se zabývá modulárně-topologickou optimalizací konstrukcí a poddajných mechanismů. Tato dvouúrovňová úloha hledá současně optimální topologii sady modulů a jejich optimální prostorové uspořádání. V první části je pomocí konvexní optimalizace řešen návrh příhradových modulů pro snížení vlivu elastických nestabilit stěn kompozitního nosníku. Následně je zavedeno zobecnění na konstrukční prvky s ohybovou tuhostí a představen přístup, který pomocí Lasserovy hierarchie umožňuje nalezení globálního optima, což doposud nebylo pro tyto typy úloh možné. Třetí kapitola zavádí koncept modulárně-topologické optimalizace a navrhuje metodu řešení založenou na kombinaci metaheuristiky a konvexního programování pro úlohy minimalizace poddajnosti modulárních příhradových konstrukcí. Popis skladebného plánu a spojitosti jednotlivých modulů je zde řešen pomocí formalismu Wangových dláždění. Ve čtvrté kapitole je představena heuristická sekvenční metoda pro modulárně-topologickou optimalizaci konstrukcí a mechanismů, která není závislá na konvexitě úlohy návrhu topologie modulů. Pro generování skladebných plánů je využita kombinace volnomateriálové optimalizace a shlukování. Efektivitu navržené metody ilustruje návrh poddajných mechanismů složených z opakovaně použitelných modulů. Poslední část se zabývá samotným problémem generování skladebných plánů Wangova dláždění a zahrnuje několik autorem navržených exaktních a heuristických metod řešení.

Klíčová slova: modulárně-topologická optimalizace, semidefinitní programování, globální optimalita, opakovatelnost, Wangovo dláždění

Překlad názvu: Modulárně-topologická optimalizace konstrukcí a mechanismů

Contents

Chapter 1
Introduction

As of 1995, 80 % of the overall product development costs were spent in the first 5 % of the design process [173]. Although the percentage may have decreased since, the early design stage still requires several interconnected multidisciplinary decisions, such as selecting the product shape, material, the principle of functionality, or the manufacturing method. A considerable effort to reduce the human intervention has provoked establishing the Building Information Modeling, Additive Manufacturing, and, ultimately, the Industry 4.0 initiative. In structural design and materials engineering, topology optimization [22] is indeed *the* tool that produces optimized, almost free-form conceptual designs but also enables a nearly instant concept-to-prototype workflow thanks to additive manufacturing [228].

An equally significant contemporary challenge appears in designing sustainable structures reducing the environmental footprint [149]. In materials engineering, this challenge requires focusing on component reusability, structural reconfigurability, repairability, and reduction of waste through recycling [31, 153, 5]. Although conventional topology optimization is unable to address these issues due to customized production, the integration of modularity principles enables this. Modular structures and architectured materials, made up of a few repeating pieces called modules, are used to reduce production costs and provide multi-configurational designs [43, 219, 214]. Modularity can also help balancing structural complexity, which is common in optimal structures [111], while simultaneously enhancing mass production quality [137] and reducing production time by efficient fabrication [63].

In topology optimization, modularity has often improved the solution efficiency by reducing the design space dimensionality through patterned material distribution [13, 6]. However, the *modular-topology* optimization problem is bilevel—it requires not only designing the module topologies but also their connectable spatial arrangement. Since finding an optimal assembly plan is a combinatorial problem, topology optimization is accelerated only if an assembly plan is fixed a priori.

Therefore, the initial studies in the topology optimization of modular structures incorporated a single non-rotatable module type, the Periodic Unit Cell (PUC) [128, 183, 127]. Only recent developments included the module rotations [188, 132] and multiple module types [189, 227, 88], superseding the inefficient structural performance of PUCs. Restricted by the computational aspects, only few studies have exploited modularity to create components reusable within different structures [189, 223] but remained limited to compliance optimization problems.

1.1 Research objectives

In the perspective of the aforementioned, three key questions shall be answered to enable a more general design of reusable modular structures, mechanisms and, in the future, of architectured materials and metamaterials:

1. How should the module topologies look like?
2. How to encode module connectivity?
3. How should the modules be arranged?

This thesis compiles five manuscripts (three published and two in preparation or under review) related to these questions. The author of this thesis was the first author in all of them, conducting the majority of the research, implementation, and manuscript writing.

In Chapter 2, we address Question 1 by means of an industrial problem of reducing elastic wall instabilities in a thin-walled filament-wound carbon fiber composite beam. To this goal, we propose solving a linear semidefinite program to generate a minimum-weight segmented modular internal structure that limits the elastic instabilities by increasing the fundamental free-vibration eigenfrequency. Subsequently, the internal structure is converted to a solid geometry and produced with additive manufacturing, allowing the beam prototype to be wounded, cured and finalized. Finally, verification and validation demonstrate the design effectiveness.

While Chapter 2 assumes discretizations with truss elements, a more general model also incorporates elements with bending stiffnesses. In Chapter 3, we investigate a global solution to topology optimization of frame and shell structures. Aiming to solve these non-convex problems for the first time, we start by recognizing that they are indeed a minimization of a polynomial function over a semi-algebraic feasible set. Consequently, we propose using the moment-sum-of-squares (Lasserre) hierarchy to solve them. First, we provide a suitable formulation for the hierarchy. In each degree of the hierarchy, the emergent semidefinite relaxation produces a lower-bound design that we project onto the feasible set of the original problem to build up a feasible upper bound. Then, each step of the hierarchy provides a certificate of global ε-optimality and, in the limit, these bounds are equal in the case of unique global minimizers. Final numerical experiments reveal that the convergence is finite for all our test cases.

In Chapter 4, we study the modular-topology optimization problem in its full generality for the case of truss module discretizations, thereby covering Questions 1–3. To simplify control over module interfaces and the module number, we propose utilizing stochastic corner Wang tilings as a viable formalism for defining the assembly plans. We treat the design problem in its original, bilevel setting, with the lower-level module design problem modeled via second-order conic programming and the upper-level assembly plan design with meta-heuristics. Due to the convexity of the lower-level problem, the performance of individual assembly plans is unique and allows for their direct comparison. Ultimately, the proposed procedure is applied to generate module designs reusable within a hinge beam and an L-shaped domain.

However, the method in Chapter 4 is impractical in conjunction with continuum topology optimization because the underlying problems are non-convex, thus preventing a reliable comparison of assembly plan performances. To avoid the need of assessing these performances, we propose a sequential procedure that generates efficient assembly plans heuristically in

Chapter 5. The proposed method relies on a solution to free material optimization problems on a module-interface-based mesh. The resulting (locally-)optimal spatially-varying anisotropic material parameters are then partitioned by a novel clustering algorithm into a predefined number of classes, representing the interface labels or edge types in the Wang tiling formalism. Finally, for the Wang tile assembly plan, we solve a conventional topology optimization problem extended towards modularity to produce optimized modules (re)usable in structures and, for the first time in compliant mechanisms.

Generating finite-sized Wang tilings, adopted in this thesis to formalize assembly plans of modular structures and module connectivity, is an $\mathcal{NP}$-complete problem in general. In Appendix A, we study Question 3 alone and ask how to produce and control Wang tilings. To this goal, we develop four integer programming formulations and three heuristic algorithms together with a heuristic scheme.

Chapter 2

Designing modular 3D printed reinforcement of wound composite hollow beams with semidefinite programming

Abstract: *Fueled by their excellent stiffness-to-weight ratio and the availability of mature manufacturing technologies, filament wound carbon fiber reinforced polymers represent ideal materials for thin-walled laminate structures. However, their strong anisotropy reduces structural resistance to wall instabilities under shear and buckling. Increasing laminate thickness degrades weight and structural efficiencies and the application of a dense internal core is often uneconomical and labor-intensive. In this contribution, we introduce a convex linear semidefinite programming formulation for truss topology optimization to design an efficient non-uniform lattice-like internal structure. The internal structure not only reduces the effect of wall instabilities, mirrored in the increase of the fundamental free-vibration eigenfrequency, but also keeps weight low, secures manufacturability using conventional three-dimensional printers, and withstands the loads induced during the production process. We showcase a fully-automatic procedure in detail for the design, prototype manufacturing, and verification of a simply-supported composite machine tool component, including validation with roving hammer tests. The results confirm that the 3D-printed optimized internal structure almost doubles the fundamental free-vibration eigenfrequency, allowing to increase working frequency of the machine tool, even though the ratio between elastic properties of the carbon composite and the ABS polymer used for 3D printing exceeds two orders of magnitude.*

Reproduced from:

[191] M. Tyburec, J. Zeman, J. Novák, M. Lepš, T. Plachý, and R. Poul, Designing modular 3D printed reinforcement of wound composite hollow beams with semidefinite programming, *Materials & Design*, 183:108131, 2019, DOI: 10.1016/j.matdes.2019.108131

2.1 Introduction

Offering excellent stiffness-to-weight ratios, high damping, and a low sensitivity to fatigue and corrosion, carbon fiber reinforced polymers (CFRPs) are employed in high-tech applications, including bodies of racing cars in automotive [66], propulsors and turbines in naval [34, 222], wing boxes and ailerons in aerospace [91], and rocket bodies in the space industry [198]. Considerable attention has been therefore paid to methods for optimizing structural performance of these laminates, particularly the laminate layup [73, 74]. To concurrently maximize bending stiffness and keep weight low, the outer dimensions of these structures tend to be maximized, while the wall thickness is minimized. The "thin-walledness" of the resulting structures, combined with their anisotropy, renders them highly sensitive to shear and wall buckling instabilities manifested in low fundamental free-vibration eigenfrequencies.

Below, we review common approaches to topology optimization that reduce wall instabilities by designing an internal structure, Section 2.1.1. Section 2.1.2 provides a brief introduction to semidefinite programming and highlights several applications in structural optimization. Finally, Section 2.1.3 reveals the merits of designing internal structures using semidefinite programming.

2.1.1 Topology optimization

Topology optimization techniques [22] provide the means for reducing wall instabilities when designing sufficiently stiff yet lightweight structures. In the simplest setting—beam cross section optimization—we search an optimal two-dimensional cross-sectional shape or a stiffening of structures whose outer shape is predefined [102]. Blasques [29], for example, maximized the fundamental free-vibration eigenfrequency while accounting for the mass and shear center position constraints; Nguyen *et al.* [142] optimized cross-sections of prismatic beams to maximize their buckling loads.

The design of optimal core sandwich structures, whose skins are stiffened by a thick core is a related challenge. For honeycomb, solid, truss, and foam rectangular panels under in-plane compression or shear loads, optimal periodic topologies can be found analytically by considering the optimality criterium of all failure modes occurring simultaneously [199]. For complex boundary conditions, parametric shape-optimization studies are usually performed. Wang and McDowell [200] studied the geometry of a metal honeycomb sandwich beam core under torsion and bending and Xu and Qiu [216] optimized the lattice core of a composite sandwich panel to increase the fundamental eigenfrequency while accounting for uncertainties in the model. They concluded that bending eigenfrequencies increase with increasing strut thicknesses, with an increase in the elastic and shear modulus of the composite, and with a decrease in density. Although Daynes *et al.* [47] optimized spatially-graded lattice structures within a single sandwich panel domain, surprisingly, almost no prior research seems to have stepped beyond parametric intuition-based designs [27, 84], the rare exception being the multi-scale topology optimization approach investigated by Coelho and Rodrigues [39].

Questioning whether, where, and how to stiffen already engineered designs in order to further improve their structural performance constitutes the central question of the reinforcement problem [147, 51], superseding the former dimensional reduction and periodicity assumptions. Initial studies in this area have considered maximization of the fundamental eigenfrequency [51] and improving the structural frequency response of plane elastic structures [134] using the

homogenization and optimality criteria methods, respectively.

Using the ground structure approach for topology optimization of truss structures, Bendsøe *et al.* [24] fixed cross-sectional areas of a set of bars and searched for their stiffest truss reinforcement, a (non-smooth) convex quadratic programming formulation. Alternatively, the effect of a fixed boundary structure has been approximated by an appropriate application of nodal forces to the ground structure [14, 151], but this choice influences, however, the optimized design.

In the setting of continuous topology optimization, Luo and Gea [133] developed a systematic optimization approach for the topology and orientation design of composite stiffeners of plates and shells in both static and dynamic settings, and Wang *et al.* [205] optimized the overall structural rigidity of an automobile body through a maximization of the fundamental eigenfrequency. In aerospace applications, Maute and Allen [135] optimized a wing's internal structure, subjected to fluid-surface interactions; Aage *et al.* [1] performed an extremely large-scale optimization of the internal structure of a Boeing 777 wing, while avoiding the traditional rib and spar designs [182]. In military applications, topology optimization was the basis for the design of additively-manufactured lattice-reinforced penetrative warheads [156] and for optimizing the layout weight of stiffeners in composite submarines subjected to nonsymmetric wave slap loads [159].

Other methods relevant to internal structure design have arisen in conjunction with recently introduced coating and infill optimization problems. Clausen *et al.* [36] developed a formulation for the optimization of (uniformly) *coated* structures, wherein a base material, *infill*, was surrounded by another material at the interfaces, finding a porous, complex infill significantly improves both structural buckling resistance and robustness to local perturbations when compared to optimized solid structures of equal weight and similar stiffnesses [37, 38]. In three dimensions, optimized designs further exploit the merits of closed shell surfaces through the sandwich effect [38].

Inspired by natural, bone-like microstructures, Wu *et al.* [210] optimized a spatially non-uniform porous infill, Wang *et al.* [207] developed a sequential approach for generating graded lattice mesostructures, and Zhu *et al.* [229] introduced a novel asymptotic-analysis-based homogenization approach. All these methods automatically design stiff yet porous infills for additive manufacturing products while superseding the traditional pattern-based designs [129]. Finally, Wu *et al.* [211] extended their approach to the ultimate setting of a concurrent optimization of coated structures and porous infills, and Groen *et al.* [76] have developed a homogenization-based method to accelerate solutions.

2.1.2 Semidefinite programming

It has been shown in recent decades that several structural optimization problems can be modeled as semidefinite programs. Linear semidefinite programming (SDP) is a subset of convex optimization of the form

$$\min_{\mathbf{x}} \ \mathbf{c}^{\mathrm{T}}\mathbf{x} \tag{2.1a}$$

$$\text{s.t. } \mathbf{X} = \mathbf{F}_0 + \sum_{i=1}^{m} x_i \mathbf{F}_i, \tag{2.1b}$$

$$\mathbf{X} \succeq 0, \tag{2.1c}$$

and involves minimization of a linear function (2.1a) over a spectrahedron, which is an intersection of an affine space (2.1b) with the cone of symmetric positive semidefinite matrices (2.1c). In (2.1c), the notation "$\succeq 0$" enforces positive semidefiniteness of the left hand side. Due to the linear dependence of $\mathbf{X}$ on $\mathbf{x}$ (2.1b), (2.1c) is commonly referred to as a linear matrix inequality (LMI).

Applications of semidefinite programming to structural design were pioneered by Ben-Tal and Nemirovski [16], de Klerk *et al.* [48], and Vandenberghe and Boyd [197] who developed formulations for minimum-compliance and weight truss topology optimizations. The main added value of SDP lies in its ability to effectively avoid the non-differentiability of multiple eigenvalues for free-vibrations [146, 2] and buckling [19, 106], robust optimization [15], and bounds improvement for optimization problems in a discrete setting [33]. Semidefinite programming has also found applications in optimal materials design, the Free Material Optimization approach [18], or in the limit analyses [28].

2.1.3 Aims and novelty

In this contribution, we consider an industrial problem of designing the least-weight internal structure of a thin-walled filament-wound composite machine tool component prone to shear and buckling wall instabilities. The beam laminate was designed for bearing dynamic loads, allowing us to describe the wall instabilities naturally in terms of free-vibrations eigenfrequencies.

In current production process, the wall instabilities are reduced by inserting a uniform foam core structure into the beam interior, an uneconomical and labor-intensive process. Conversely, we have aimed to automatically design a structurally-efficient internal structure which can easily be manufactured using conventional low-cost 3D printers.

To this goal, we extended the convex (linear) semidefinite programming formulation introduced by Ohsaki *et al.* [146] and Ben-Tal and Nemirovski [15] to design globally-optimal least-weight lattice-like internal structures and apply it to increasing the fundamental eigenfrequency and decreasing the compression-molding compliance of a thin-walled composite beam prototype. Note that Achtziger and Kočvara [2] avoided prescribed structural elements but allowed for a non-structural mass and Ohsaki *et al.* [146] did not consider prescribed mass or stiffness.

After introducing the case study of a simply-supported CFRP beam design in Section 2.2, we develop its finite element representation in Section 2.3.1. For this representation, a semidefinite programming formulation for truss topology optimization of internal structures is developed in Section 2.3.2. Having designed the optimal internal structure, we post-process the optimization outputs and export, in a fully-automated way, the internal structure for additive manufacturing

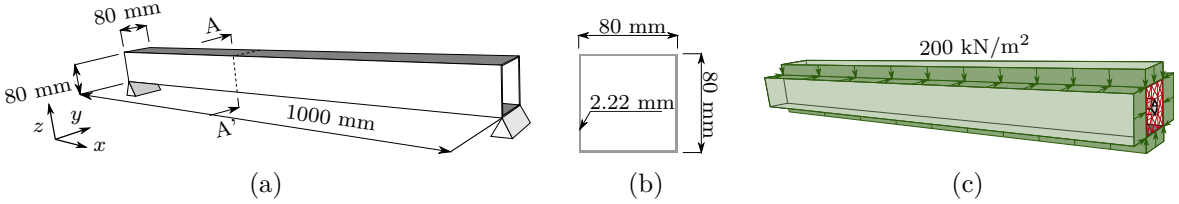


Figure 2.1: Case study setup. (a) Outer dimensions and simply supported boundary conditions, (b) prismatic cross-section, and (c) compression molding load case.

Layer	E_1 [GPa]	E_2 [GPa]	G_{12} [GPa]	ν_{12} [-]	ν_{23} [-]	θ [deg]	ρ [kg/m^3]	t [mm]
1	128.2	5.0	3.4	0.34	0.35	89.3	1,428	0.25
2	421.9	3.7	3.2	0.37	0.35	0.0	1,680	1.25
3	130.9	5.0	3.4	0.34	0.35	26.9	1,458	0.18
4	130.9	5.0	3.4	0.34	0.35	−26.9	1,458	0.36
5	130.9	5.0	3.4	0.34	0.35	26.9	1,458	0.18
6 (casing)	2.0	2.0	0.7	0.37	0.37	0.0	1,040	0.80

Table 2.1: Material properties of the wound composite beam laminae. E_1 and E_2 stand for the Young moduli in the fiber and transverse directions, respectively; G_{12} denotes the shear modulus, ν_{12} and ν_{23} are Poisson's ratios. θ constitutes the angle between the 1-direction and x, rotating around the beam surface normals. Finally, ρ and t denote the density and thickness of the plies.

in Section 2.3.3. During manufacturing, the internal structure serves as the support for carbon fibers in the filament-winding production phase, and a prototype is created. Section 2.4 describes verification and experimental validation of the prototype and concludes that its response agreed well with the model prediction.

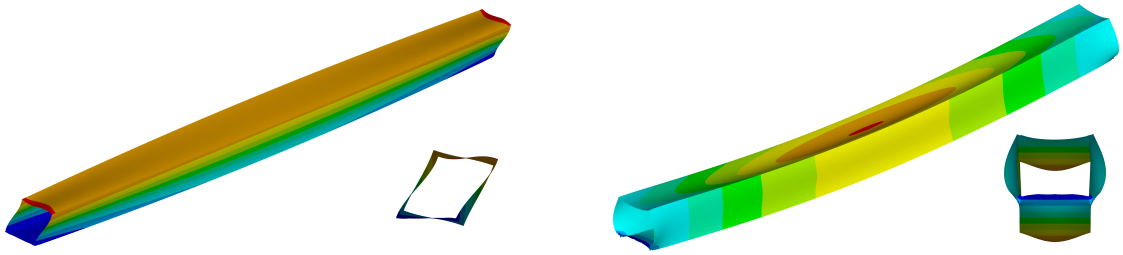
2.2 Case study

As the basic structure, we consider a prismatic, laminated composite beam 1,000 mm long, with a 80 × 80 mm thin-walled cross-section 2.2 mm thick, Fig. 2.1b. According to current manufacturing technology, beam production consists of several steps, in which a supporting structure made of manually processed high-density foam is wound biaxially with a combination of ultra high modulus (UHM) and high modulus (HM) carbon fibers saturated with epoxy resin. The supporting structure prevents cross-section distortions induced by compression-molding loads as shown in Fig. 2.1c. Subsequently, the beam is cured, the supporting structure is pulled out, and the beam outer surface is finalized.

The final product is exposed primarily to loads that induce bending. For this purpose, most of the carbon fibers are aligned with the beam's longitudinal axis (layer 2 in Table 2.1), denoted by x in Fig. 2.1a, whereas the remaining layers reduce the susceptibility to delamination. See Table 2.1, where all layers are listed by their orientations relative to the beam's longitudinal axis, θ. This layered composition reliably transmits the design forces to the supports, and is thus fully sufficient in this sense.

Attributed to transversely isotropic material properties, the beam's walls are, however, prone to elastic wall instabilities under shear and buckling, which also manifests in free-vibration modes and frequencies of the non-reinforced beam. Figure 2.2 confirms that the first fundamental eigenmode with a frequency of 128.5 Hz corresponds to shear wall instabilities, whereas the second eigenmode combines bending with buckling; all higher eigenmodes (not shown) exhibit similar wall instabilities. Because the fundamental eigenfrequency limits the maximum working frequency of the machine part, its increase is of considerable interest.

Although the effect of these instabilities can be reduced by additional laminate layers or by also keeping the uniform foam structure for operational loads, the added weight, decrease in the bending eigenfrequencies, and labor-intensive production process render these approaches both time-inefficient and uneconomical.



(a) First eigenmode with a frequency of 128.5 Hz.

(b) Second eigenmode with a frequency of 403.1 Hz.

Figure 2.2: Axonometric and front view on the (a) first and (b) second eigenmodes of the composite beam predicted by the finite element model.

2.3 Optimal design of internal structure

The aim of this section is to cast the optimal internal structure design problem in the form of a linear semidefinite program (2.1). The internal structure has to withstand compression molding loads with a maximum deflection bound, while the internal structure is temporarily supported by a steel mandrel passing through the beam interior, Fig. 2.1c. Most importantly, the internal structure is supposed to increase the beam fundamental eigenfrequency via reduction of wall instabilities.

In this section, we first describe the finite element model of the composite beam. This finite element model serves then as the basis for establishing the optimization problem formulation, yielding an optimal internal structure design. The section is concluded by discussing post-processing steps necessary to maintain manufacturability of the design.

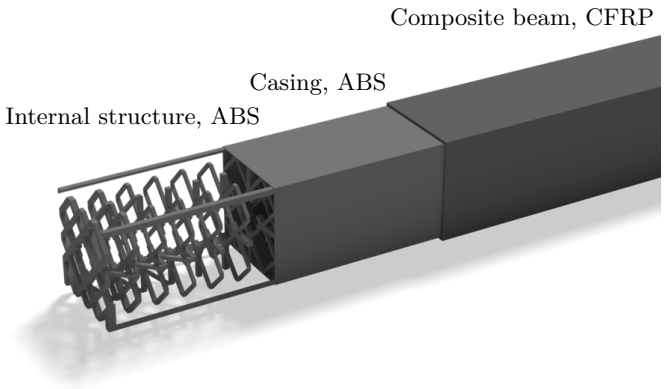


Figure 2.3: The entire structure of considered composite beam design: internal structure (used for the reduction of wall instabilities and for increase of the lowest free-vibration frequency); casing of the internal beam structure (to allow for wounding the final composite layer); composite layers, which transmits working load applied to the beam.

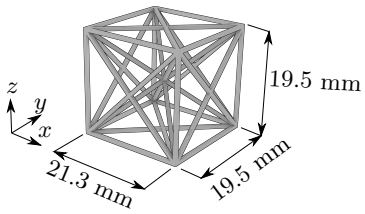


Figure 2.4: Ground structure building block, which fill in the entire internal volume of the composite beam to represent a (to be optimized) internal beam structure. Cross-sectional areas of individual trusses are design variables of the optimization problem (2.2).

2.3.1 Finite element model

The outer composite beam surface is discretized with shell elements which are supplied with the material properties from Table 2.1. The beam internal structure is modeled by bar (truss) elements, with the isotropic Acrylonitrile Butadiene Styrene (ABS) material properties [32]: elastic modulus $E_{\mathrm{ABS}} = 2$ GPa, Poisson ratio $\nu_{\mathrm{ABS}} = 0.37$, and density $\rho_{\mathrm{ABS}} = 1,040$ kg/m^3.

Special care needs to be paid to establishing a rigid connection between the internal structure and the carbon composite. The so-called *casing*, see Fig. 2.3, which is a 0.8 mm thin layer of printed beam walls, further prevents leaking of the epoxy resin into the beam's interior. Casing is modeled as the bottom layer of the laminate composition, recall Table 2.1.

The finite element model for the optimization part was developed in MATLAB. In this model, the outer laminate was modeled with four-node MITC4 elements [60]. The composite beam interior was discretized into the *ground structure* [53], a set of admissible truss[1] elements, whose cross-sections we search in the optimization part. The ground structure was constructed from $47 \times 4 \times 4$ modular building blocks shown in Fig. 2.4, to guarantee manufacturability of the entire internal structure with 3D printing. Note that the bars placed within the location of the steel mandrel were removed from the ground structure and that the shell element nodes coincided with the ground structure nodes, resulting in a rather coarse discretization of the outer layer.

2.3.2 Formulation of the optimization problem

2.3.2.1 Non-convex formulation

Adopting the previously described discretization, our goal is to find the cross-sectional areas $\mathbf{a}$ of n_{b} bars in the minimum-weight (or volume) ground structure, such that the fundamental eigenfrequency exceeds the user-defined lower threshold $\overline{f}$, taken as 300 Hz in what follows, while exhibiting limit displacements $\overline{u}$ of the reinforced structure during the compression

[1] Based on comparative simulations (not shown), modeling internal structure with trusses or beams leads to an insignificant difference in the structural response which enabled us to employ truss topology optimization approaches in Section 2.3.2.

molding load case. This leads to the following optimization problem

$$\min_{\mathbf{a},\mathbf{u}_{\mathrm{cm}},\mathbf{u}} \ \boldsymbol{\ell}^{\mathrm{T}}\mathbf{a} \tag{2.2a}$$

$$\text{s.t.} \quad \inf_{(\mathbf{M}_{\mathrm{fv}}^{\mathrm{IS}}(\mathbf{a})+\mathbf{M}_{\mathrm{fv}}^{\mathrm{C}})\mathbf{u}\neq\mathbf{0}} \frac{\mathbf{u}^{\mathrm{T}}\left(\mathbf{K}_{\mathrm{fv}}^{\mathrm{IS}}(\mathbf{a})+\mathbf{K}_{\mathrm{fv}}^{\mathrm{C}}\right)\mathbf{u}}{\mathbf{u}^{\mathrm{T}}\left(\mathbf{M}_{\mathrm{fv}}^{\mathrm{IS}}(\mathbf{a})+\mathbf{M}_{\mathrm{fv}}^{\mathrm{C}}\right)\mathbf{u}} \geq \ \overline{\lambda}, \tag{2.2b}$$

$$\left(\mathbf{K}_{\mathrm{cm}}^{\mathrm{IS}}(\mathbf{a})+\mathbf{K}_{\mathrm{cm}}^{\mathrm{C}}\right)\mathbf{u}_{\mathrm{cm}} = \mathbf{f}_{\mathrm{cm}}, \tag{2.2c}$$

$$-\overline{u}\mathbf{1} \leq \mathbf{u}_{\mathrm{cm,disp}} \leq \ \overline{u}\mathbf{1}, \tag{2.2d}$$

$$\mathbf{0} \leq \mathbf{a} \leq \ \overline{a}\mathbf{1}, \tag{2.2e}$$

with

$$\overline{\lambda} = 4\pi^2\overline{f}^2. \tag{2.3}$$

In this formulation, the vector $\boldsymbol{\ell}$ appearing in the objective function (2.2a) collects the bar lengths in the truss ground structure. The Rayleigh quotient in (2.2b) involves stiffness, $\mathbf{K}_{\mathrm{fv}}^{\mathrm{C}}$, and mass, $\mathbf{M}_{\mathrm{fv}}^{\mathrm{C}}$, matrices of the outer shell structure for free-vibration analysis together with stiffness, $\mathbf{K}_{\mathrm{fv}}^{\mathrm{IS}}(\mathbf{a})$, and mass, $\mathbf{M}_{\mathrm{fv}}^{\mathrm{IS}}(\mathbf{a})$, matrices of the internal structure. The design-dependent contributions of the internal structure are obtained as

$$\mathbf{K}_{\mathrm{fv}}^{\mathrm{IS}}(\mathbf{a}) = \sum_{e=1}^{n_{\mathrm{b}}} \hat{\mathbf{K}}_{\mathrm{fv},e}^{\mathrm{IS}} a_e, \qquad \mathbf{M}_{\mathrm{fv}}^{\mathrm{IS}}(\mathbf{a}) = \sum_{e=1}^{n_{\mathrm{b}}} \hat{\mathbf{M}}_{\mathrm{fv},e}^{\mathrm{IS}} a_e, \tag{2.4}$$

where $\hat{\mathbf{K}}_{\mathrm{fv},e}^{\mathrm{IS}}$ and $\hat{\mathbf{M}}_{\mathrm{fv},e}^{\mathrm{IS}}$ stand for the stiffness and mass matrix of individual bars in the free-vibration (fv) setting, respectively; a_e is the e-th component of $\mathbf{a}$ and $\overline{\lambda}$ the limit fundamental free-vibrations eigenvalue.

The constraints (2.2c) and (2.2d) address the compression-molding (cm) load case, recall Fig. 2.1c. Specifically, (2.2c) introduces the generalized nodal displacements $\mathbf{u}_{\mathrm{cm}}$ in response to the generalized load vector $\mathbf{f}_{\mathrm{cm}}$ corresponding to the compressive load, and $\mathbf{u}_{\mathrm{cm,disp}}$ denotes the displacement components of $\mathbf{u}_{\mathrm{cm}}$. The stiffness matrix corresponding to this load case consists again of the design-independent, $\mathbf{K}_{\mathrm{cm}}^{\mathrm{C}}$, and design-dependent, $\mathbf{K}_{\mathrm{cm}}^{\mathrm{IS}}(\mathbf{a})$, parts; the latter is obtained as in (2.4). The symbol $\mathbf{1}$ denotes a column vector of all ones. Notice that the stiffness matrices in (2.2b) and (2.2c) differ because of different boundary conditions in the operational, Fig. 2.1a, and manufacturing, Fig. 2.1c, load cases. The constraint (2.2d) requires the displacement components of $\mathbf{u}_{\mathrm{cm}}$ to remain smaller than the user-defined limit value $\overline{u}$, considered to be 0.5 mm in this study. Finally, (2.2e) requires the cross-sectional areas of the bars to be non-negative and smaller than $\overline{a} = 200$ mm^2, a value set by the additive manufacturing constraints.

A closer comparison of the optimization problem of Eq. (2.2) and that of Eq. (2.1) reveals that the problem of Eq. (2.2) lacks the structure of a semidefinite program. Namely, the objective function (2.2a) and the matrices in the constraints depend affinely on the design variables, $\mathbf{a}$. However, the constraints (2.2b) and (2.2c) are non-convex as the stiffness and mass matrices may become singular when the zero lower-bound for cross-sectional areas is attained in (2.2e). Moreover, (2.2b) might become non-differentiable when an eigenvalue with multiplicity higher than one is encountered. Altogether, this renders the problem (2.2) extremely difficult to solve in its original form. In the following section, we show how to re-cast the problem of Eq. (2.2) as a linear semidefinite programming problem.

2.3.2.2 Convex semidefinite program

Similar eigenvalue constraints such as (2.2b) have already been studied in detail by Ohsaki *et al.* [146] and Achtziger and Kočvara [2]. Their results allow us to rewrite (2.2b) equivalently as a convex LMI

$$\mathbf{K}_{\mathrm{fv}}^{\mathrm{IS}}(\mathbf{a}) + \mathbf{K}_{\mathrm{fv}}^{\mathrm{C}} - 4\pi^2\overline{f}^2\left(\mathbf{M}_{\mathrm{fv}}^{\mathrm{IS}}(\mathbf{a}) + \mathbf{M}_{\mathrm{fv}}^{\mathrm{C}}\right) \succeq 0, \tag{2.5}$$

where the left hand side expression is a linear function of $\mathbf{a}$. This constraint also avoids the non-differentiability of multiple eigenvalues, see, e.g., [2], and effectively eliminates the kinematic variables $\mathbf{u}$ from the problem formulation.

To attain convexity of the final formulation, the compression molding constraints (2.2c)–(2.2d) must be enforced only approximately in the form of the LMI [48, 197, 15]:

$$\begin{pmatrix} c_{\mathrm{cm}} & -\mathbf{f}_{\mathrm{cm}}^{\mathrm{T}} \\ -\mathbf{f}_{\mathrm{cm}} & \mathbf{K}_{\mathrm{cm}}^{\mathrm{IS}}(\mathbf{a}) + \mathbf{K}_{\mathrm{cm}}^{\mathrm{C}} \end{pmatrix} \succeq 0, \tag{2.6}$$

in which c_{cm} denotes a prescribed upper bound on compliance (work done by external forces) of the compression molding load case. As found from parametric studies (not shown), an appropriate value of the bound is provided as

$$c_{\mathrm{cm}} = c_{\mathrm{cm},0}\frac{\overline{u}}{\max\{|\mathbf{u}_{\mathrm{cm,disp}}|\}}, \tag{2.7}$$

where $c_{\mathrm{cm},0}$ stands for the compliance of the non-reinforced structure:

$$c_{\mathrm{cm},0} = \tilde{\mathbf{f}}_{\mathrm{cm}}^{\mathrm{T}}\left(\tilde{\mathbf{K}}_{\mathrm{cm}}^{\mathrm{C}}\right)^{-1}\tilde{\mathbf{f}}_{\mathrm{cm}}. \tag{2.8}$$

Here, $\tilde{\mathbf{K}}_{\mathrm{cm}}^{\mathrm{C}}$ and $\tilde{\mathbf{f}}_{\mathrm{cm}}$ are constructed from $\mathbf{K}_{\mathrm{cm}}^{\mathrm{C}}$ and $\mathbf{f}_{\mathrm{cm}}$, respectively, by application of appropriate boundary conditions. For this particular problem, this compliance bound resulted in a maximum deflection of 0.4 mm.

The final linear semidefinite programming formulation eventually reads as

$$\min_{\mathbf{a}} \boldsymbol{\ell}^{\mathrm{T}}\mathbf{a} \tag{2.9a}$$

$$\text{s.t. } \mathbf{K}_{\mathrm{fv}}^{\mathrm{IS}}(\mathbf{a}) + \mathbf{K}_{\mathrm{fv}}^{\mathrm{C}} - 4\pi^2\overline{f}^2\left(\mathbf{M}_{\mathrm{fv}}^{\mathrm{IS}}(\mathbf{a}) + \mathbf{M}_{\mathrm{fv}}^{\mathrm{C}}\right) \succeq 0, \tag{2.9b}$$

$$\begin{pmatrix} c_{\mathrm{cm}} & -\mathbf{f}_{\mathrm{cm}}^{\mathrm{T}} \\ -\mathbf{f}_{\mathrm{cm}} & \mathbf{K}_{\mathrm{cm}}^{\mathrm{IS}}(\mathbf{a}) + \mathbf{K}_{\mathrm{cm}}^{\mathrm{C}} \end{pmatrix} \succeq 0, \tag{2.9c}$$

$$\mathbf{1}\overline{a} \geq \mathbf{a} \geq \mathbf{0}. \tag{2.9d}$$

This formulation now possesses the structure of the linear semidefinite program introduced in Section 2.1.2, and thus can be solved efficiently via modern interior-point methods.

For numerical solution, we adopted the state-of-the-art industrial optimizer Mosek [138]. After discretization, the problem in Eq. (2.9) has 10,216 admissible bars in total, with the corresponding sizes of the linear matrix inequalities $5,154 \times 5,154$ (free-vibration, Eq. (2.9b)) and $4,608 \times 4,608$ (compliance, Eq. (2.9c)). After tweaking the optimization problem with the steps outlined in the following subsection, the optimization process itself required 13 GB of memory, and terminated after 5.75 core hours running on Intel® Xeon® Gold 6130 processors at the MetaCentrum[2] virtual organization cluster. The resulting distribution of the optimal

[2] `https://metavo.metacentrum.cz/`

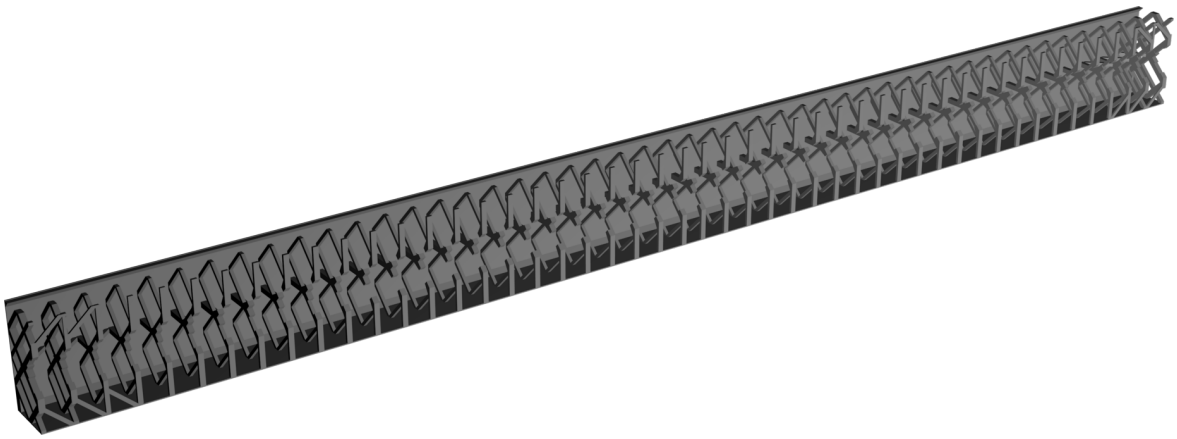


Figure 2.5: Symmetric half of the beam as cut off by the xz plane. The top shell surface is hidden to reveal the internal structure.

internal structure is shown in Fig. 2.5. Note that the internal structure increased the original weight of the beam, 1,094 g, by an additional 488 g (280 g of casing and 208 g of reinforcing bars).

Improving solver performance. To reduce the number of iterations and time per iteration to solve problem (2.9), we rescale the cross-sectional areas to obtain the optimal values of the order of 1.0 mm. Second, to improve both the numerical stability and convergence of the algorithm considerably, we rescale Eqs. (2.9b) and (2.9c) with the square root of the Frobenius norm estimates of $\mathbf{K}_{\mathrm{fv}}^{\mathrm{C}}$ (Eq. (2.9b)), and $\mathbf{K}_{\mathrm{cm}}^{\mathrm{C}}$ (Eq. (2.9c)). Finally, using the static condensation, Appendix A, and Schur complement, Appendix B, decomposition techniques, the sizes of LMIs reduce to $3,426 \times 3,426$ (free-vibration, Eq. (2.9b)) and $2,880 \times 2,880$ (compliance, Eq. (2.9c)). Consequently, memory usage was decreased from 21 GB to 13 GB, and the solution process was accelerated by 71% (from 19.5 to 5.75 core hours).

2.3.3 Post-processing

Manufacturing of the optimal design is preceded by three preprocessing steps addressing individual bars, segmentation into modules, and conversion to a solid model. Note that we checked that none of the steps led to the constraint violation and have a rather negligible impact on the objective function, i.e., after all post-processing steps, the internal structure volume increased from 168.3 cm^3 to 175 cm^3.

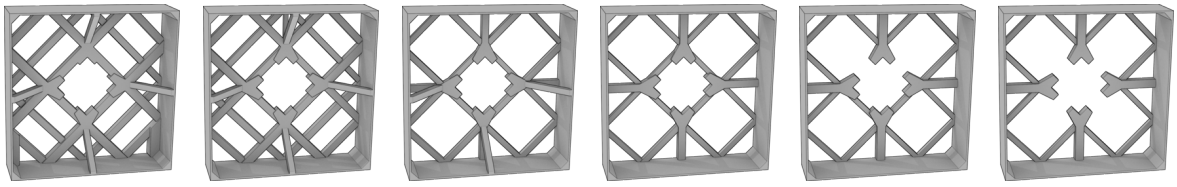


Figure 2.6: Solid models of typical topologies of segments.

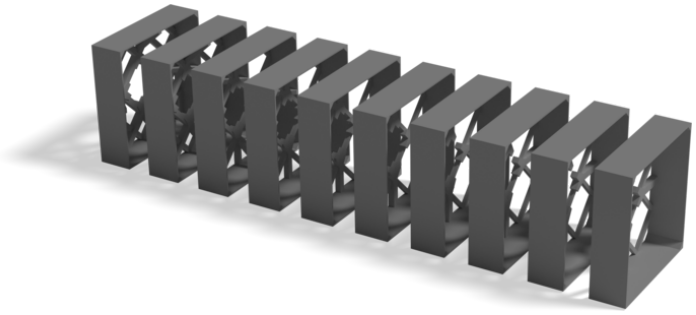


Figure 2.7: Segmentation of the beam internal structure.

Bars post-processing. In the initial step of module post-processing, we assign square cross-sections to each bar with the square side length according to the optimal area, $d_e = \sqrt{a_e}$. Next, we check potential intersection of bars and place a node at each intersection, which subdivides them into two and defines the new element lengths. Third, for each bar, we set the cross-sectional size d_e to at least $l_e/40$, because more slender bars are difficult to manufacture with the Prusa 3D printers used in this study. In addition to the optimized bars, the internal structure is extended with short L-shaped beams that ensure mechanical interaction between the internal structure and the steel mandrel, thus defining an empty 20.05×20.05 mm space along the beam longitudinal axis x for its insertion, see Figs. 2.6 and 2.9a.

Segmentation. To enable parallel manufacturing with conventional 3D printers, we split the optimized internal structure into 48 segments of approximately 20 mm in length, see Fig. 2.7 where ten selected segments are shown, to be assembled later on the steel mandrel. Such segmentation requires re-alignment of bars within each beam cross-section and along the beam longitudinal axis to ensure the correct external beam dimension and a clearly defined interface among adjacent modules, see Fig. 2.8 for an illustration. Note that segment production does not require any supporting material when printed along the beam longitudinal axis x, which would be impossible when printing the internal structure as a single-piece product.

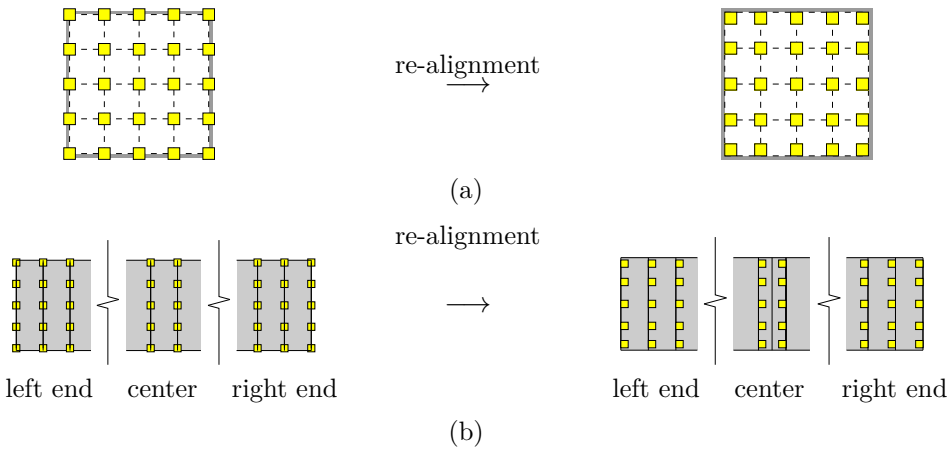


Figure 2.8: Illustration of bar cross-sections re-alignment along the beam (a) yz section, and (b) longitudinal axis x.

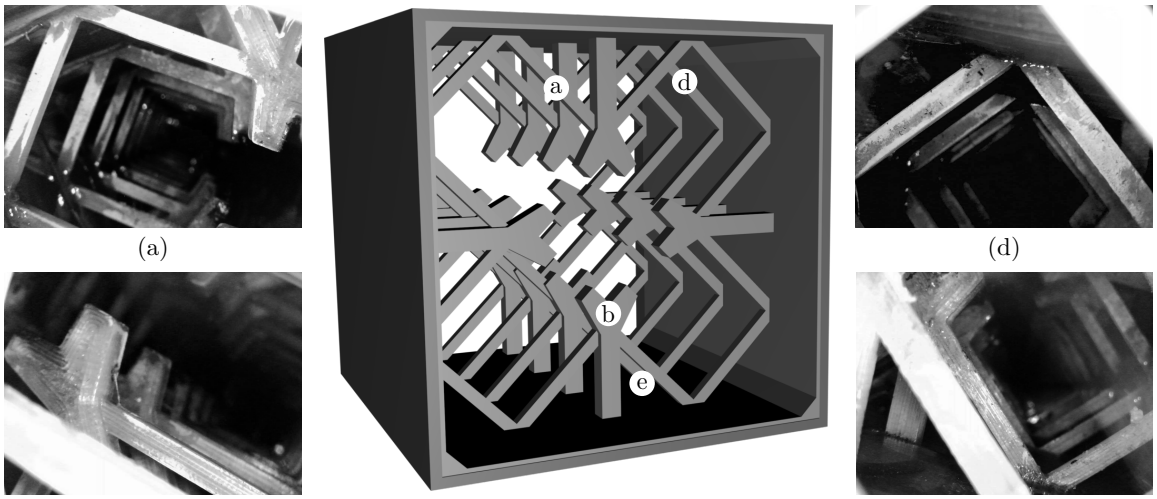


Figure 2.9: Endoscope camera photographs (a), (b), (d) and (e) as captured in the manufactured beam interior (c) showing that the 3D-printed internal structure successfully withstood the compression-molding loads.

Solid conversion. Axial model conversion is performed independently and in parallel for each node of the ground structure. We determine first all bars attached to the considered node, elongate them by one half of their cross-sectional side lengths at both of their ends, and cut the more distant half of each of these bars off. These half-bars are then modeled by a mesh-based representation. Geometries of individual nodes then result from the mesh-boolean operations performed with the CORK[3] library. Finally, the overall segment geometry consists of the union of all nodal geometries, see Fig. 2.6 for typical topologies of post-processed segments, and can be readily exported to patch-based STL file format, for example.

2.4 Results

2.4.1 Manufacturing

After the automated export of the optimized internal structure into STL format, the part was additively manufactured using the Fused Deposition Modeling method with Prusa i3 MK3 printers. Printed segments were inserted on a $20 \times 20 \times 1,200$ mm steel mandrel of 1.5 mm wall thickness, with its surface lubricated with Vaseline to simplify the pull-out process, and connected with acetone etching and a thin layer of epoxy glue.

The prototype beam was produced by CompoTech Plus company using the filament winding technology with axial fiber placement. This technology relies on the positioning of the tows of carbon fibers impregnated by the epoxy resin on the casing, placed in specified directions and specified quantity to reach expected dimensions and mechanical properties of the final product. The casing defines the internal shape of the beam and acts as an internal mold. After the fiber placement operation, the product (with still liquid resin) is placed into the press, the outer shape is formed, and the composite is consolidated. In the press, the product

[3] `https://github.com/gilbo/cork`

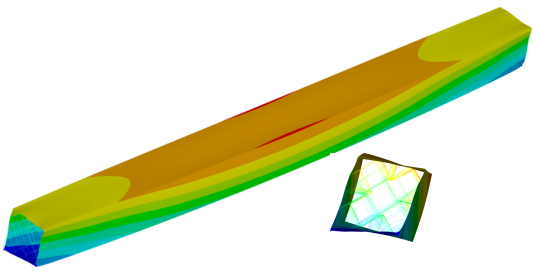
Figure 2.10: Manufactured prototype of a composite beam with optimized stiffening internal structure.

hardens at the room temperature. Finally, the prototype, Fig. 2.10, is postcured at the elevated temperature of 90°C.

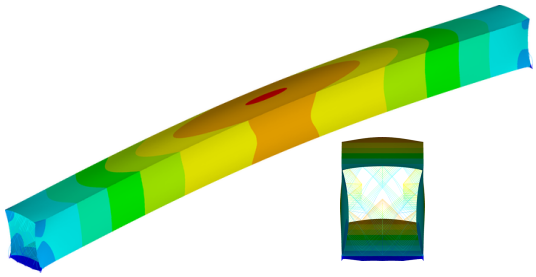
The successful manufacturing process was followed by inspection of the prototype using an endoscope camera. Video and photograph sequences, see Fig. 2.9, revealed that the internal structure successfully withstood the compression molding pressure without any significant visual defects. The minor deviations from the assumed model reside in a small amount of Vaseline residue and slight leakage of the epoxy resin through casing interfaces. Another difference appeared in the increased outer dimensions of the beam, 0.32 mm on average, caused by an insufficiently closed press cover. The total prototype weight of 1.768 kg was therefore by 186 g higher than the model predictions due to the additional epoxy resin.

2.4.2 Verification

Recall that in the optimization we required increasing the fundamental free-vibration eigenfrequency above 300 Hz. To check this value, we employed an independent model in Ansys. Compared to the model used for optimization, this model employs the dimensions measured in-situ, more refined discretization of the outer shells (element type Shell181), and models the internal structure with beam elements (Beam188) instead of trusses. Besides, the composite shells are supplemented with an additional layer of epoxy resin to account for the increased epoxy content. As a result, the model predicts that the beam fundamental eigenfrequency was increased by 92% from 128.5 Hz to 246.7 Hz, compare Figs. 2.2 and 2.11. The effect of



(a) First eigenmode with a frequency of 246.7 Hz.



(b) Second eigenmode with a frequency of frequency 347.0 Hz.

Figure 2.11: Axonometric and front view on the (a) first and (b) second eigenmodes of the reinforced composite beam predicted by the refined finite element model.

Figure 2.12: Free-free-vibration validation setup. Locations of 54 impact points are indicated by gray squares and positions of 3 accelerometers are marked by white circles.

wall instabilities was reduced jointly in all the remaining eigenmodes (not shown).

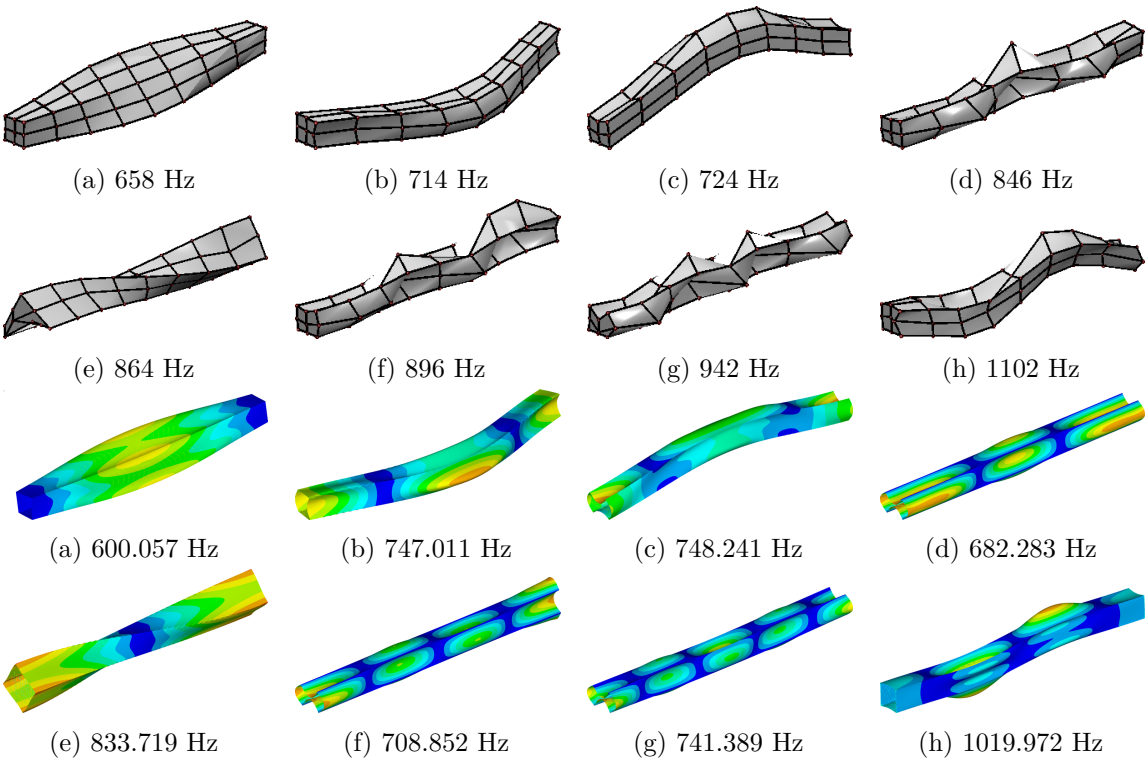
Even though the fundamental eigenfrequency did not exceed the limit value, we find these results satisfactory because of two reasons: First, we attribute this discrepancy mainly to the manufacturing imperfections, which can be attributed to the prototype character of the manufacturing process and can be easily resolved in serial production. Second, the constraint violation is comparable to the difference between numerics and experiments as shown in the next section.

2.4.3 Validation

Dynamic response was validated with the roving hammer test in the free-free-vibration setting because it eliminates the need to reproduce the simply supported kinematic boundary condition in the experiment. To this goal, the beam was suspended at one of its ends, three piezoelectric acceleration transducers Type 4507B005 Brüel&Kjaer were placed on the beam's outer surface, two of which were located in the middle of adjacent sides of the beam's cross-section at one-eighth of the beam's length, and the third one was placed at the corner of the beam, Fig. 2.12. Two adjacent sides of the beam surface were marked with a regularly spaced grid of 54 points, 27 on each side. These points then served as the excitation points for the impact hammer Type 8206 Brüel&Kjaer equipped with a force transducer.

The measurement was realized using data acquisition front-end hardware Type 3560B Brüel&Kjaer. The frequency response functions (FRFs) were evaluated from the recorded response (acceleration) and excitation (force) using the Fast Fourier Transform for all 54 points. The natural frequencies and mode shapes were evaluated from the FRFs with MEscope software developed by the Vibrant Technology company.

Experimentally determined natural modes and the values of natural eigenfrequencies, Fig. 2.13 top, were compared with the results of numerical simulations, Fig 2.13 bottom.



(a) 658 Hz (b) 714 Hz (c) 724 Hz (d) 846 Hz

(e) 864 Hz (f) 896 Hz (g) 942 Hz (h) 1102 Hz

(a) 600.057 Hz (b) 747.011 Hz (c) 748.241 Hz (d) 682.283 Hz

(e) 833.719 Hz (f) 708.852 Hz (g) 741.389 Hz (h) 1019.972 Hz

Figure 2.13: Selected experimentally determined natural frequencies and mode shapes, (a)–(h) top, and finite element model predictions of eigenmodes and eigenfrequencies, (a)–(h) bottom.

Direct comparison in Table 2.2 reveals sufficient agreement of up to 9% for eigenfrequencies of shear, bending, and torsional eigenmodes. In the case of buckling, we failed to measure the first and second buckling natural modes, and for the higher eigenmodes, the model predictions underestimate the natural frequencies by more than 20%. We attribute these deviations to the overall difficulty of measuring the buckling natural modes and to the manufacturing defects discussed in the previous section.

Eigenmode	f_{FEM} [Hz]	f_{EXP} [Hz]	A [Hz]	D [%]
First shear	600.1	658	2	−8.8
First bending y	747.0	714	2	+4.6
First bending z	748.2	724	2	+3.3
Third buckling	682.3	846	4	−19.3
First torsion	833.7	864	4	−3.5
Fourth buckling	708.9	896	4	−20.9
Fifth buckling	741.4	942	6	−21.3
Sixth buckling	790.6	1004	6	−21.3
Second bending z	1020.0	1102	6	−7.4

Table 2.2: Comparison of model prediction of eigenfrequencies f_{FEM} and measured natural frequencies f_{EXP} using the roving hammer test. Accuracy of individual measurements is denoted by A and the deviation of the model from the experiment by D.

2.5 Summary and outlook

This contribution introduces and investigates a unique, fully-automatized procedure from an idea to prototyping, with applications to the manufacturing of thin-walled structural composite hollow beams. In particular, the considered prototype product is stiffened with a low weight internal structure designed by an efficient convex linear semidefinite programming formulation. This formulation increased the fundamental free-vibration eigenfrequency above a specified threshold value while avoiding the traditional issue of non-differentiability of multiple eigenvalues [2], and limited structural compliance of a compression-molding load case. The optimization output of the non-uniformly distributed lattice-like internal structure was further automatically post-processed and converted into a solid model ready for support-less additive manufacturing.

Our methodology was verified by designing and producing the simply-supported CFRP beam prototype. Optimization yielded an internal structure of 488 g which increased the fundamental eigenfrequency by 92% and limited the effect of wall instabilities. Moreover, the deflections within the compression-molding load case were limited to ± 0.5 mm.

After a successful prototype production, the structural response was validated using the roving hammer test, which showed that bending, torsional, and shear eigenmodes exhibited good agreement with model predictions. For the wall buckling eigenmodes, however, the finite element model underestimated the natural frequencies by almost 22%. We attribute this to difficulties in measuring these natural modes and to manufacturing defects associated with compression-molding deformations of the casing.

Improving the structural response with a material more than two orders of magnitude more compliant when compared to CFRP suggests concentrating on substituting ABS with high-stiffness continuous carbon fiber in future studies. Another essential future enhancement resides in accelerating the optimization algorithm by exploiting the range-space sparsity [101] associated with the segment-based internal-structure decomposition.

2.A Static condensation of static LMI

Consider the equilibrium equation

$$\mathbf{K}(\mathbf{a})\mathbf{u} = \mathbf{f} \tag{2.10}$$

split into two sets of equations

$$\begin{pmatrix} \mathbf{K}_\mathrm{a}(\mathbf{a}) & \mathbf{K}_\mathrm{b} \\ \mathbf{K}_\mathrm{b}^\mathrm{T} & \mathbf{K}_\mathrm{c} \end{pmatrix} \begin{pmatrix} \mathbf{u}_\mathrm{a} \\ \mathbf{u}_\mathrm{b} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_\mathrm{a} \\ \mathbf{f}_\mathrm{b} \end{pmatrix}, \tag{2.11}$$

such that only the principal submatrix $\mathbf{K}_\mathrm{a}(\mathbf{a})$ depends affinely on $\mathbf{a}$[4]. Assuming that the system (2.10) is solvable uniquely for some $\mathbf{a}$, i.e., it holds that $\exists \mathbf{a} \geq \mathbf{0} : \mathbf{K}(\mathbf{a}) \succ 0$, where "$\succ 0$" denotes positive definiteness of the left hand side. Note that $\mathbf{a} = \mathbf{1}$ is sufficient for verification that no rigid movement within the structure can occur. Because $\mathbf{K}_\mathrm{c}$ is therefore invertible, the degrees of freedom $\mathbf{u}_\mathrm{b}$ can be expressed from the second row in terms of $\mathbf{u}_\mathrm{a}$

$$\mathbf{u}_\mathrm{b} = (\mathbf{K}_\mathrm{c})^{-1}\,\mathbf{f}_\mathrm{b} - (\mathbf{K}_\mathrm{c})^{-1}\,\mathbf{K}_\mathrm{b}^\mathrm{T}\mathbf{u}_\mathrm{a} \tag{2.12}$$

[4] In the context of this article, the matrix $\mathbf{K}_\mathrm{a}(\mathbf{a})$ comprises the degrees of freedom of the truss ground structure, $\mathbf{K}_\mathrm{c}$ contains the remaining (rotational) degrees of freedom, and $\mathbf{K}_\mathrm{b}$ is the coupling term.

and inserted back into the first row,

$$\left[\mathbf{K}_\mathrm{a}(\mathbf{a}) - \mathbf{K}_\mathrm{b}\left(\mathbf{K}_\mathrm{c}\right)^{-1}\mathbf{K}_\mathrm{b}^\mathrm{T}\right]\mathbf{u}_\mathrm{a} = \mathbf{f}_\mathrm{a} - \mathbf{K}_\mathrm{b}\left(\mathbf{K}_\mathrm{c}\right)^{-1}\mathbf{f}_\mathrm{b}. \tag{2.13}$$

Structural compliance (work done by external forces) is expressed as

$$c = \mathbf{u}_\mathrm{a}^\mathrm{T}\mathbf{f}_\mathrm{a} + \mathbf{u}_\mathrm{b}^\mathrm{T}\mathbf{f}_\mathrm{b}. \tag{2.14}$$

After inserting (2.12) and acknowledging that $\mathbf{K}_\mathrm{c}^{-1}$ is Hermitian, we obtain

$$c = \mathbf{u}_\mathrm{a}^\mathrm{T}\left[\mathbf{f}_\mathrm{a} - \mathbf{K}_\mathrm{b}\left(\mathbf{K}_\mathrm{c}\right)^{-1}\mathbf{f}_\mathrm{b}\right] + \mathbf{f}_\mathrm{b}^\mathrm{T}\left(\mathbf{K}_\mathrm{c}\right)^{-1}\mathbf{f}_\mathrm{b}, \tag{2.15}$$

i.e., compliance of the condensed problem (2.13) and a constant term. Because the compliance of the condensed problem is positive by definition, the constant term represents a non-negative lower bound on compliances achievable by the internal structure design.

Finally, the LMI

$$\begin{pmatrix} c & -\mathbf{f}^\mathrm{T} \\ -\mathbf{f} & \mathbf{K}(\mathbf{a}) \end{pmatrix} \succeq 0 \tag{2.16}$$

is equivalent to a smaller LMI

$$\begin{pmatrix} c - \mathbf{f}_\mathrm{b}^\mathrm{T}\left(\mathbf{K}_\mathrm{c}\right)^{-1}\mathbf{f}_\mathrm{b} & -\mathbf{f}_\mathrm{a}^\mathrm{T} + \mathbf{f}_\mathrm{b}^\mathrm{T}\left(\mathbf{K}_\mathrm{c}\right)^{-1}\mathbf{K}_\mathrm{b}^\mathrm{T} \\ -\mathbf{f}_\mathrm{a} + \mathbf{K}_\mathrm{b}\left(\mathbf{K}_\mathrm{c}\right)^{-1}\mathbf{f}_\mathrm{b} & \mathbf{K}_\mathrm{a}(\mathbf{a}) - \mathbf{K}_\mathrm{b}\left(\mathbf{K}_\mathrm{c}\right)^{-1}\mathbf{K}_\mathrm{b}^\mathrm{T} \end{pmatrix} \succeq 0. \tag{2.17}$$

Further, if $c > \mathbf{f}_\mathrm{b}^\mathrm{T}\left(\mathbf{K}_\mathrm{c}\right)^{-1}\mathbf{f}_\mathrm{b}$ is a prescribed constant (i.e., not a variable), then (2.17) is further reducible, using the Schur complement lemma, e.g., [67, Proposition 16.1], to a yet smaller LMI

$$\begin{aligned} &\mathbf{K}_\mathrm{a}(\mathbf{a}) - \mathbf{K}_\mathrm{b}\left(\mathbf{K}_\mathrm{c}\right)^{-1}\mathbf{K}_\mathrm{b}^\mathrm{T} - \left(-\mathbf{f}_\mathrm{a}^\mathrm{T} + \mathbf{f}_\mathrm{b}^\mathrm{T}\left(\mathbf{K}_\mathrm{c}\right)^{-1}\mathbf{K}_\mathrm{b}^\mathrm{T}\right) \\ &\left(c - \mathbf{f}_\mathrm{b}^\mathrm{T}\left(\mathbf{K}_\mathrm{c}\right)^{-1}\mathbf{f}_\mathrm{b}\right)^{-1}\left(-\mathbf{f}_\mathrm{a} + \mathbf{K}_\mathrm{b}\left(\mathbf{K}_\mathrm{c}\right)^{-1}\mathbf{f}_\mathrm{b}\right) \succeq 0. \end{aligned} \tag{2.18}$$

2.B Reducing size of free-vibration LMI

In the case of the free-vibration constraint, we need to directly apply the (generalized) Schur complement lemma. Beginning with reordering of rows and columns, we split the symmetric LMI (2.9b) such that only the $\mathbf{K}_\mathrm{a}(\mathbf{a})$ and $\mathbf{M}_\mathrm{a}(\mathbf{a})$ matrices are functions of $\mathbf{a}$, and the other blocks are constant,

$$\begin{pmatrix} \mathbf{K}_\mathrm{a}(\mathbf{a}) - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{a}(\mathbf{a}) & \mathbf{K}_\mathrm{b} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{b} \\ \mathbf{K}_\mathrm{b}^\mathrm{T} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{b}^\mathrm{T} & \mathbf{K}_\mathrm{c} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{c} \end{pmatrix} \succeq 0. \tag{2.19}$$

For the (standard) Schur complement trick we require $\mathbf{K}_\mathrm{c} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{c} \succ 0$ [67, Proposition 16.1]. Since $\mathbf{K}_\mathrm{c} \succ 0$ (boundary conditions exclude rigid motions), and $\mathbf{M}_\mathrm{c} \succ 0$ by definition, we only need to secure that the fundamental eigenfrequency f_0 of the generalized eigenvalue problem

$$\mathbf{K}_\mathrm{c}\mathbf{u}_\mathrm{b} - \lambda\mathbf{M}_\mathrm{c}\mathbf{u}_\mathrm{b} = 0, \tag{2.20}$$

with $\lambda = 4\pi^2 f^2$, is strictly greater than $\overline{f}$.

Let us therefore first assume that $0 \leq \overline{f} < f_0$. Then, the inverse of $\mathbf{K}_\mathrm{c} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{c}$ exists and (2.19) can be rewritten equivalently using the Schur complement lemma into a smaller-sized LMI

$$\mathbf{K}_\mathrm{a}(\mathbf{a}) - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{a}(\mathbf{a}) - \left(\mathbf{K}_\mathrm{b} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{b}\right)\left(\mathbf{K}_\mathrm{c} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{c}\right)^{-1}\left(\mathbf{K}_\mathrm{b}^\mathrm{T} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{b}^\mathrm{T}\right) \succeq 0. \tag{2.21}$$

Second, consider that $f_0 < \overline{f}$. Because the matrix $\mathbf{K}_\mathrm{c} - 4\pi^2(f_0+\varepsilon)^2\mathbf{M}_\mathrm{c}$ is indefinite for any $\varepsilon > 0$, which renders the original LMI (2.19) infeasible, the eigenfrequency f_0 constitutes an upper bound for achievable fundamental eigenfrequencies of the reinforced structure. From the mechanical point of view, the eigenmodes $\mathbf{u}_\mathrm{b}$ associated with f_0 excite degrees of freedom not reinforced by the internal structure, and therefore the associated eigenfrequencies can not be increased by any admissible internal structure design (given the specific discretization).

In the case $\overline{f} = f_0$, reduction of (2.19) relies on the generalized Schur complement lemma [67, Theorem 16.1], so that (2.19) is equivalent to

$$\mathbf{K}_\mathrm{a}(\mathbf{a}) - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{a}(\mathbf{a}) - \left(\mathbf{K}_\mathrm{b} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{b}\right)\left(\mathbf{K}_\mathrm{c} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{c}\right)^{\dagger}\left(\mathbf{K}_\mathrm{b}^\mathrm{T} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{b}^\mathrm{T}\right) \succeq 0, \tag{2.22a}$$

$$\left[\mathbf{I} - \left(\mathbf{K}_\mathrm{c} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{c}\right)\left(\mathbf{K}_\mathrm{c} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{c}\right)^{\dagger}\right]\left(\mathbf{K}_\mathrm{b}^\mathrm{T} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{b}^\mathrm{T}\right) = \mathbf{0}, \tag{2.22b}$$

where $(\bullet)^\dagger$ denotes the Moore-Penrose pseudo-inverse of $\bullet$, and $\mathbf{I}$ is the identity matrix. The second condition (2.22b) holds iff the columns of $\mathbf{K}_\mathrm{b}^\mathrm{T} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{b}^\mathrm{T}$ are in the image of $\mathbf{K}_\mathrm{c} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{c}$. Indeed, (2.22b) can then be rewritten to

$$\left[\left(\mathbf{K}_\mathrm{c} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{c}\right) - \left(\mathbf{K}_\mathrm{c} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{c}\right)\left(\mathbf{K}_\mathrm{c} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{c}\right)^{\dagger}\left(\mathbf{K}_\mathrm{c} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{c}\right)\right]\mathbf{C} = \mathbf{0}, \tag{2.23}$$

with the columns of $\mathbf{C}$ being the coefficients of linear combinations of the columns of $\mathbf{K}_\mathrm{c} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{c}$, making the term in the square brackets vanish [67, Lemma 14.1].

Because $\mathrm{Im}(\mathbf{K}_\mathrm{c} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{c}) = \mathrm{Ker}(\mathbf{K}_\mathrm{c} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{c})^\perp$ by [67, Lemma 13.1], it is spanned by

$$\mathrm{span}\left\{\mathbf{u}_\mathrm{b} : \left(\mathbf{K}_\mathrm{c} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{c}\right)\mathbf{u}_\mathrm{b} = \mathbf{0}\right\}^\perp. \tag{2.24}$$

Clearly, $\overline{f} = f_0$ might be achieved iff the columns of the coupling term $\mathbf{K}_\mathrm{b}^\mathrm{T} - 4\pi^2\overline{f}^2\mathbf{M}_\mathrm{b}^\mathrm{T}$ are orthogonal to the eigenmodes occurring in (2.20) at f_0. From the mechanical point of view, induction of these eigenmodes would result in a decrease of the associated eigenfrequencies. Note that in practice, equation (2.22b) can be verified numerically, but it does not guarantee a feasible solution to (2.22a), because other (higher) eigenfrequencies associated with eigenmodes of (2.20) may decrease below f_0 due to the coupling term.

Chapter 3

Global optimality in minimum compliance topology optimization of frames and shells by moment-sum-of-squares hierarchy

Abstract: *The design of minimum-compliance bending-resistant structures with continuous cross-section parameters is a challenging task because of its inherent non-convexity. Our contribution develops a strategy that facilitates computing all guaranteed globally optimal solutions for frame and shell structures under multiple load cases and self-weight. To this purpose, we exploit the fact that the stiffness matrix is usually a polynomial function of design variables, allowing us to build an equivalent non-linear semidefinite programming formulation over a semi-algebraic feasible set. This formulation is subsequently solved using the Lasserre moment-sum-of-squares hierarchy, generating a sequence of outer convex approximations that monotonically converges from below to the optimum of the original problem. Globally optimal solutions can subsequently be extracted using the Curto-Fialkow flat extension theorem. Furthermore, we show that a simple correction to the solutions of the relaxed problems establishes a feasible upper bound, thereby deriving a simple sufficient condition of global ε-optimality. When the original problem possesses a unique minimum, we show that this solution is found with a zero optimality gap in the limit. These theoretical findings are illustrated on several examples of topology optimization of frames and shells, for which we observe that the hierarchy converges in a finite (rather small) number of steps.*

Reproduced from:

[196] M. Tyburec, J. Zeman, M. Kružík, and D. Henrion, Global optimality in minimum compliance topology optimization of frames and shells by moment-sum-of-squares hierarchy, *Structural and Multidisciplinary Optimization*, 64(4):1963–1981, 2021, DOI: 10.1007/s00158-021-02957-5

3.1 Introduction

Structural optimization is a research field developing concepts for the design of efficient structures. It was pioneered by Michell [136], who showed that minimum-weight truss structures under a single load case are fully-stressed, and the optimal trajectories of their bars align with the principal stress directions. Hence, optimal designs can contain an infinite number of bars in general. This drawback, which hinders their manufacturability, was overcome in the work of Dorn [53] by introducing the ground structure approach, effectively discretizing the continuum into a finite set of potential nodes and their interconnections of finite elements. Because the dimensionality of these potential elements is lower than that of the continuum and the presence and sizing of each element are investigated, this setting is referred to as discrete topology optimization.

A tremendous progress has been made for the case of trusses. While Dorn [53] developed a linear programming formulation for the single-load-case plastic design, Bendsøe *et al.* [23] and Achtziger *et al.* [4] introduced a convex displacement-based elastic-design quadratic program that additionally allowed for multiple load cases. Its dual, which incorporates the cross-section variables explicitly, was shown by Lobo *et al.* [130] and Ben-Tal and Nemirovski [17] to be a second-order conic program. This latter formulation handles multiple load cases with stress constraints efficiently [192]. Convexity prevails even for fundamental free-vibration constraints, in which case semidefinite programming can be used [3, 191].

A completely different situation holds for bending-resistant structures with continuous design variables. To the best of our knowledge, no convex formulation has been established so far, and, therefore, mostly local optimization techniques have been used. Among these, Saka [167] developed a sequential linear programming approach to design minimum-weight framed structures, and Wang and Arora [206] improved over its solution efficiency by using sequential quadratic programming instead. Another, relaxation-based sequential semidefinite programming method was proposed by Yamada and Kanno [217] to deal with vibration problems. Nonlinear programming [65], Optimality Criteria (OC) [100, 35], the Method of Moving Asymptotes (MMA) [184, 64], and meta-heuristics [8] are other commonly used alternatives.

The only related, (dual) global optimization approach was adopted in Section 5.3 of Murota *et al.* [140], who used a hierarchy of specific semidefinite programming relaxations [112] for a frame structure topology optimization problem with a lower bound on the fundamental free-vibrations eigenfrequency. Compared to our methodology, only convergence of the objective function values is guaranteed theoretically [113] and approximate solutions may remain unknown in the case of multiple global optima. On the other hand, when applied to the compliance minimization problems considered in this manuscript, upper bounds, serving as a tool to measure the approximate solution quality, could have been provided by developments similar to ours.

Except for our earlier conference paper [194], which is a very preliminary version of this manuscript, the only published global approach that secures extraction of the global optima considers a discrete setting of the problem, in which case the cross-sections are selected from a predefined catalog, allowing to use the branch-and-bound method [96].

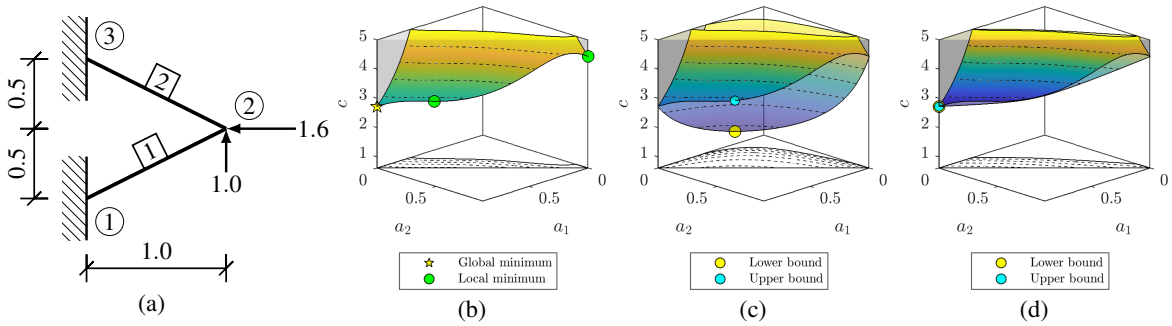


Figure 3.1: (a) Boundary conditions for the motivating problem, (b) the sublevel set $c \leq 5$ of its feasible design space, and its (c) first and (d) second convex outer approximations constructed by the moment-sum-of-squares approach. Variables a_1 and a_2 stand for the cross-section areas of the two elements and c denotes the corresponding compliance (assuming moments of inertia $I_i = a_i^2$, $i \in \{1, 2\}$).

3.1.1 Motivation

This contribution investigates a conceptual design of bending-resistant structures with continuous design variables. As follows from the previous survey, only local optimization approaches have been adopted so far to tackle the frame/shell structure compliance minimization problem. It is, therefore, not surprising that they fail to converge to globally optimal solutions even for toy problems, such as the one shown in Fig. 3.1a: a two-element frame structure with fully clamped nodes ① and ③, subjected to a single load case induced by the forces applied at the node ②.

Consider a linear-elastic material with the dimensionless Young modulus $E = 1$, available material volume $\overline{V} = 1$, element cross-sections parameterized by their area a_i and the moment of inertia $I_i(a_i) = a_i^2$, which corresponds to rectangular cross-sections with the height-to-width ratio of 12, for example. Our goal is to find the cross-section areas a_i associated with the elements $i = \{1, 2\}$ that induce minimum compliance c within all non-negative a_i satisfying the volume bound $\overline{V}$. Notice here that the compliance follows from a non-linear equilibrium equation, in which the stiffness matrix is a second-order polynomial function of $\mathbf{a}$.

Standard local optimization techniques such as OC, MMA, and MATLAB inbuilt optimizer `fmincon` all converge[1] to the optimized compliance $c = 2.895$ and the corresponding areas $a_1 = 0.652$ and $a_2 = 0.242$. For the second local optimum, we have $a_1 = 0$, $a_2 = 0.4\sqrt{5}$ and $c = 4.429$. While a similar design also arises in truss structure optimal design (i.e., when neglecting rotational degrees of freedom), it is not optimal here: the globally optimal design possesses compliance $c^* = 2.719$ and requires $a_1^* = 0.4\sqrt{5}$ and $a_2^* = 0$, see Fig. 3.1b.

Because only few local optima exist in this case, the global optimum may be reached by examining a few different starting points in the optimizers. Such a procedure, however, cannot assure global optimality and cannot neither assess quality of the optimized designs with respect to the global optimum. Even with a moderate increase in the number of structural elements, addition of design-dependent loads, or using higher-order polynomials for the moments of inertia, finding certified globally-optimal minimum-compliance designs becomes extremely challenging.

[1]For OC and MMA, we adopted the commonly-used starting point of uniform mass distribution, i.e., $a_1 = a_2 = 0.2\sqrt{5}$. For `fmincon`, the default starting point was used.

3.1.2 Aims and novelty

We address this problem by exploiting the simple fact that the constraints can be formulated as polynomial functions, hence forming a (basic) semi-algebraic feasible set. Using a polynomial objective function in addition, the moment-sum-of-squares (Lasserre) hierarchy of convex outer approximations (relaxations) can be used to solve and extract all globally-optimal solutions. These relaxations provide a non-decreasing sequence of lower bounds, eventually attaining the optimal compliance, Figs 3.1c-3.1d. In addition, we show how to correct such obtained lower-bound designs, and hence generate feasible upper bounds. A comparison of these bounds then assesses the design quality, and their equality establishes a simple sufficient condition of global optimality. We further show that when a unique global optimum exists, we can expect the occurrence of a bound equality. Fortunately, this situation occurs quite often when the design domain lacks structural and boundary conditions symmetries.

This paper is organized as follows. In Section 3.2, we introduce polynomial optimization and the moment-sum-of-squares hierarchy. Section 3.3.1 develops a non-linear semidefinite programming formulation for topology optimization of frame structures, which we modify subsequently for the moment-sum-of-squares hierarchy in Section 3.3.2. Section 3.3.3 reveals how to correct the lower-bound designs generated by the hierarchy to obtain feasible upper bounds, and Section 3.3.4 introduces the sufficient condition of global ε-optimality as well as a zero optimality gap for the case of a unique global optimum. Section 3.3.5 outlines the required changes in notation to allow for thickness optimization of shell structures. These results are then illustrated on five selected optimization problems in Section 3.4. We finally summarize our contributions in Section 3.5.

3.2 Moment-sum-of-squares hierarchy

In this section, we briefly outline the moment-sum-of-squares hierarchy when applied to solution of problems with polynomial matrix inequalities. For more information, we refer the reader to an expository text [85] and to the excellent books [11, 122].

Suppose we aim to solve an optimization problem of the form

$$f^* = \inf_{\mathbf{x}} f(\mathbf{x}) \tag{3.1a}$$

$$\text{s.t. } \mathbf{G}(\mathbf{x}) \succeq 0, , \tag{3.1b}$$

where $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ is a real polynomial function and $\mathbf{G}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{S}^m$ is a real polynomial mapping, so that $\forall i, j : G_{i,j}(\mathbf{x}) = G_{j,i}(\mathbf{x})$ are real polynomial functions of $\mathbf{x}$. The degree of these polynomials is less than or equal to $k \in \mathbb{N}$. The symbol $\mathbb{S}^m$ denotes the space of real symmetric square matrices of size m, and "$\succeq$" establishes an ordering of fundamental eigenvalues, i.e., $\mathbf{G}(\mathbf{x})$ in (3.1b) is positive semidefinite. Hence, we call (3.1b) a polynomial matrix inequality (PMI) in what follows and denote its feasible set by $\mathcal{K}(\mathbf{G})$.

Clearly, the nonlinear semidefinite program (3.1) covers a variety of convex optimization problems as special cases, including linear and quadratic programming or linear semidefinite programming, see, e.g., [17, Section 4.2]. Although these instances can be solved in a polynomial time, hence efficiently, (3.1) exhibits $\mathcal{NP}$-hardness in general. This can be seen, for example, by a reduction from binary programming, in which case the main diagonal of $\mathbf{G}(\mathbf{x})$ contains both $x_i^2 - x_i$ and $x_i - x_i^2$ terms for all $i \in \{1, \ldots, n\}$.

Despite the fact that the admissible set of $\mathbf{x}$ is generally non-convex, (3.1) admits an equivalent reformulation to a convex optimization problem over a finite-dimensional cone of polynomials $C_k(\mathcal{K}(\mathbf{G}))$ of degree at most k which are non-negative on $\mathcal{K}(\mathbf{G})$, i.e.,

$$f^* = \sup_{\lambda}\{\lambda : f(\mathbf{x}) - \lambda \geq 0, \forall \mathbf{x} \in \mathcal{K}(\mathbf{G})\} \tag{3.2a}$$

$$= \sup_{\lambda}\{\lambda : (f - \lambda) \in C_k(\mathcal{K}(\mathbf{G}))\}. \tag{3.2b}$$

Here, the feasible set of (3.2a) is the intersection of an infinite number of linear inequalities, while the supremum of (3.2b) is performed such that the shifted polynomial $f - \lambda$ is non-negative on $\mathcal{K}(\mathbf{G})$. Unfortunately, it is not known how to handle $C_k(\mathcal{K}(\mathbf{G}))$ simply and in a tractable way.

To introduce an approach that allows to solve (3.2), we first adopt the following notation. Let $\mathbf{x} \mapsto \mathbf{b}_k(\mathbf{x})$ be the polynomial space basis of polynomials of degree at most k,

$$\mathbf{b}_k(\mathbf{x}) = \begin{pmatrix} 1 & x_1 & x_2 & \dots & x_n & x_1^2 & x_1x_2 & \dots & x_1x_n & x_2^2 & x_2x_3 & \dots & x_n^2 & \dots & x_2^3 & \dots & x_n^k \end{pmatrix}, \tag{3.3}$$

Then, any polynomial $p(\mathbf{x})$ of degree at most k can be written as

$$p(\mathbf{x}) = \mathbf{q}^{\mathrm{T}}\mathbf{b}_k(\mathbf{x}), \tag{3.4}$$

in which $\mathbf{q}$ denotes a vector of coefficients associated with the basis $\mathbf{b}_k(\mathbf{x})$.

Definition 3.1. The polynomial matrix $\mathbf{\Sigma}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{S}^m$ is a (matrix) sum-of-squares (SOS) if there exists a polynomial matrix $\mathbf{H}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^{m\times o}$, $o \in \mathbb{N}$, such that

$$\mathbf{\Sigma}(\mathbf{x}) = \mathbf{H}(\mathbf{x})\left[\mathbf{H}(\mathbf{x})\right]^{\mathrm{T}}, \ \forall \mathbf{x} \in \mathbb{R}^n. \tag{3.5}$$

Let $\langle \cdot, \cdot \rangle$ denote the standard inner product on matrices[2], $\boldsymbol{\alpha} \in \mathbb{N}^{|\mathbf{b}_k(\mathbf{x})|}$ with $\mathbf{1}^{\mathrm{T}}\boldsymbol{\alpha} \leq k$ be the multi-index associated with the basis $\mathbf{b}_k(\mathbf{x})$, and let $\mathbf{y} \in \mathbb{R}^{|\mathbf{b}_k(\mathbf{x})|}$ be the moments (of probability measures supported on $\mathcal{K}(\mathbf{G}(\mathbf{x}))$) indexed in $\mathbf{b}_k(\mathbf{x})$; see [122, Section 2.7] for more details. In what follows, we adopt the following notation for the elements of $\mathbf{y}$:

$$y_{\boldsymbol{\alpha}} = y_{\prod_{i=1}^n x_i^{\alpha_i}} \text{ is associated with } \prod_{i=1}^n x_i^{\alpha_i}. \tag{3.6}$$

For example, when $\boldsymbol{\alpha} = (0 \quad 0 \quad 1 \quad 2)^{\mathrm{T}}$, $y_{0012} = y_{x_3^1x_4^2}$ corresponds to the polynomial $x_3^1x_4^2 \in \mathbf{b}_k(\mathbf{x})$, where $k \geq 3$.

Assumption 3.2. [85] Assume that there exist SOS polynomials $\mathbf{x} \mapsto p_0(\mathbf{x})$ and $\mathbf{x} \mapsto \mathbf{R}(\mathbf{x})$ such that the superlevel set $\{\mathbf{x} \in \mathbb{R}^n : p_0(\mathbf{x}) + \langle \mathbf{R}(\mathbf{x}), \mathbf{G}(\mathbf{x})\rangle \geq 0\}$ is compact.

Note that Assumption 3.2 is an algebraic certificate of compactness of the feasible set in problem (3.1). When Assumption 3.2 holds, then, the dual of (3.2) can be written equivalently as an infinite-dimensional generalized problem of moments, which is equipped with a finite-dimensional truncation:

$$f^{(r)} = \min_{\mathbf{y}} \mathbf{q}_0^{\mathrm{T}}\mathbf{y} \tag{3.7a}$$

$$\text{s.t.} \quad y_0 = 1, \tag{3.7b}$$

$$\mathbf{M}_k(\mathbf{y}) \succeq 0, \tag{3.7c}$$

$$\mathbf{M}_{k-d}(\mathbf{G}(\mathbf{x})\mathbf{y}) \succeq 0, \tag{3.7d}$$

[2]Let $\mathbf{X}, \mathbf{Y}$ be real matrices of the same dimensions. Then, $\langle \mathbf{X}, \mathbf{Y}\rangle := \mathrm{Tr}(\mathbf{X}\mathbf{Y}^{\mathrm{T}})$, where Tr is the trace operator.

in which $r = k/2$ denotes the relaxation degree, and d stands for the maximum degree of polynomials in $\mathbf{G}(\mathbf{x})$. In addition, $\mathbf{q}_0$ are the coefficients of the polynomial $f(\mathbf{x})$ with respect to the basis $\mathbf{b}_k(\mathbf{x})$, and $\mathbf{M}_k(\mathbf{y})$ with $\mathbf{M}_{k-d}(\mathbf{G}(\mathbf{x})\mathbf{y})$ are the truncated moment and localizing matrices associated with $\mathbf{y}$ and $\mathbf{G}(\mathbf{x})$. For a precise definition of $\mathbf{M}_k(\mathbf{y})$ and $\mathbf{M}_{k-d}(\mathbf{G}(\mathbf{x})\mathbf{y})$, we refer the reader to [85]. These moment matrices are linear in $\mathbf{y}$, hence (3.7) is a linear semidefinite program. Because (3.7) is a finite-dimensional convex relaxation of (3.1), we have $f^{(r)} \leq f^*$, $\forall r \in \mathbb{N}$. Moreover, these relaxations are tighter with increasing r, making the sequence $\left(f^{(r)}\right)_r^\infty$ monotonically increasing and converging towards f^* (expressed by the $\nearrow$ symbol).

Theorem 3.3. *[85, Theorem 2.2] Let Assumption 3.2 be satisfied. Then, $f^{(r)} \nearrow f^*$ as $r \to \infty$ in* (3.7).

Moreover, all globally optimal solutions of (3.1) can be extracted from (3.7) based on the flat extension theorem of Curto and Fialkow [45]. Indeed, finite convergence occurs when

$$s = \operatorname{Rank}(\mathbf{M}_k(\mathbf{y}^*)) = \operatorname{Rank}(\mathbf{M}_{k-d}(\mathbf{y}^*)), \tag{3.8}$$

where $\mathbf{y}^*$ denotes the vector of optimal moments, and s stands for the minimum number of distinct global minimizers [85, Theorem 2.4].

Example. We illustrate the process of building the (Lasserre) moment-sum-of-squares hierarchy on an elementary example. Consider an optimization problem of the form

$$\min_{a,c} \quad c \tag{3.9a}$$

$$\text{s.t.} \begin{pmatrix} c & \overline{f} \\ \overline{f} & a^2 \end{pmatrix} \succeq 0, \tag{3.9b}$$

$$\overline{V} - a \geq 0, \tag{3.9c}$$

$$a \geq 0. \tag{3.9d}$$

In the first relaxation, $\mathbf{y} = \begin{pmatrix} y_{00} & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \end{pmatrix}^{\mathrm{T}}$ is indexed in the polynomial space basis $\mathbf{b}_1(a,c) = \begin{pmatrix} 1 & c & a & c^2 & ca & a^2 \end{pmatrix}^{\mathrm{T}}$. Then, the associated relaxation reads

$$\min_{\mathbf{y}} \; y_{10} \tag{3.10a}$$

$$\text{s.t.} \quad \begin{pmatrix} y_{10} & \overline{f} \\ \overline{f} & y_{02} \end{pmatrix} \succeq 0, \tag{3.10b}$$

$$\overline{V} - y_{01} \geq 0, \tag{3.10c}$$

$$y_{01} \geq 0, \tag{3.10d}$$

$$y_{00} = 1, \tag{3.10e}$$

$$\begin{pmatrix} y_{00} & y_{10} & y_{01} \\ y_{10} & y_{20} & y_{11} \\ y_{01} & y_{11} & y_{02} \end{pmatrix} \succeq 0. \tag{3.10f}$$

For $r = 2$, we have $\mathbf{y} = (y_{00}\ y_{10}\ y_{01}\ y_{20}\ y_{11}\ y_{02}\ y_{30}\ y_{21}\ y_{12}\ y_{03}\ y_{40}\ y_{31}\ y_{22}\ y_{13}\ y_{04})^{\mathrm{T}}$ indexed in $\mathbf{b}_2(a,c) = (1c\ \ a\ \ c^2\ \ ca\ \ a^2\ \ c^3\ \ c^2a\ \ ca^2\ \ a^3\ \ c^4\ \ c^3a\ \ c^2a^2\ \ ca^3\ \ a^4)^{\mathrm{T}}$. The

corresponding relaxation is written as

$$\min_{\mathbf{y}} \; y_{10} \tag{3.11a}$$

$$\text{s.t.} \begin{pmatrix} y_{10} & \overline{f} & y_{20} & \overline{f}y_{10} & y_{11} & \overline{f}y_{01} \\ \overline{f} & y_{02} & \overline{f}y_{10} & y_{12} & \overline{f}y_{01} & y_{03} \\ y_{20} & \overline{f}y_{10} & y_{30} & \overline{f}y_{20} & y_{21} & \overline{f}y_{11} \\ \overline{f}y_{10} & y_{12} & \overline{f}y_{20} & y_{22} & \overline{f}y_{11} & y_{13} \\ y_{11} & \overline{f}y_{01} & y_{21} & \overline{f}y_{11} & y_{12} & \overline{f}y_{02} \\ \overline{f}y_{01} & y_{03} & \overline{f}y_{11} & y_{13} & \overline{f}y_{02} & y_{04} \end{pmatrix} \succeq 0, \tag{3.11b}$$

$$\begin{pmatrix} \overline{V} - y_{01} & \overline{V}y_{10} - y_{11} & \overline{V}y_{01} - y_{02} \\ \overline{V}y_{10} - y_{11} & \overline{V}y_{20} - y_{21} & \overline{V}y_{11} - y_{12} \\ \overline{V}y_{01} - y_{02} & \overline{V}y_{11} - y_{12} & \overline{V}y_{02} - y_{03} \end{pmatrix} \succeq 0, \tag{3.11c}$$

$$\begin{pmatrix} y_{01} & y_{11} & y_{02} \\ y_{11} & y_{21} & y_{12} \\ y_{02} & y_{12} & y_{03} \end{pmatrix} \succeq 0, \tag{3.11d}$$

$$y_{00} = 1, \tag{3.11e}$$

$$\begin{pmatrix} y_{00} & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{pmatrix} \succeq 0. \tag{3.11f}$$

When solved, this relaxation allows for extracting the global solution of $a^* = \overline{V}$ and $c^* = \overline{f}^2/\overline{V}^2$.

3.3 Methodology

Topology optimization of discrete structures provides a natural application for the ground structure approach [53], a discretized design domain composed of a fixed set of $n_\mathrm{n} \in \mathbb{N}$ nodes and their subsets of admissible $n_\mathrm{e} \in \mathbb{N}$ finite elements. Here, we employ the simplest two-node Euler-Bernoulli frame elements that adopt linear shape functions to interpolate the longitudinal displacements and cubic shape functions to interpolate the lateral displacements and rotations. Another elements can be adopted though, see Section 3.4.4 for applications to the Timoshenko beam element and the MITC4 shell element.

Each of these finite elements (indexed with i) must be supplied with the non-negative cross-section area $a_i \in \mathbb{R}_{\geq 0}$ and the area moment of inertia $I_i \in \mathbb{R}_{\geq 0}$. These are to be found in the optimization process. In this contribution, we assume, for convenience, that the moment of inertia is a second- or third-order polynomial function of the cross-sections,

$$I_i(a_i) = c_\mathrm{II} a_i^2 + c_\mathrm{III} a_i^3, \tag{3.12}$$

with $c_\mathrm{II}, c_\mathrm{III} \in \mathbb{R}_{\geq 0}$ being fixed constants. When $I_i(a_i) = 0$ and $a_i = 0$, the finite element vanishes and does not contribute to the load transfer.

Different topology optimization formulations exist, accommodating specific needs of particular applications. Here, we consider the problem of searching the minimum-compliant design under multiple load cases (3.13a) while satisfying the linear-elastic equilibrium equation (3.13b) and limiting the material volume from above by $\overline{V} \in \mathbb{R}_{>0}$ (3.13c). Physical admissibility of the resulting designs is ensured by the non-negative cross-section areas (3.13d). Combination of these ingredients establishes the basic elastic-design formulation

$$\min_{\mathbf{a},\mathbf{u}_1,\dots,\mathbf{u}_{n_{\mathrm{lc}}}} \sum_{j=1}^{n_{\mathrm{lc}}} \omega_j \mathbf{f}_j(\mathbf{a})^{\mathrm{T}} \mathbf{u}_j \tag{3.13a}$$

$$\text{s.t.} \quad \mathbf{K}_j(\mathbf{a})\mathbf{u}_j - \mathbf{f}_j(\mathbf{a}) = \mathbf{0},\ \forall j \in \{1,\dots,n_{\mathrm{lc}}\}, \tag{3.13b}$$

$$\overline{V} - \boldsymbol{\ell}^{\mathrm{T}}\mathbf{a} \geq 0, \tag{3.13c}$$

$$\mathbf{a} \geq \mathbf{0}, \tag{3.13d}$$

in which $\boldsymbol{\omega} \in \mathbb{R}^{n_{\mathrm{lc}}}_{>0}$ are positive weights associated with n_{lc} load cases, and $\boldsymbol{\ell} \in \mathbb{R}^{n_{\mathrm{e}}}_{\geq 0}$ stands for the element lengths column vector. Further, $\mathbf{f}_j(\mathbf{a}) \in \mathbb{R}^{n_{\mathrm{dof},j}}$ and $\mathbf{u}_j \in \mathbb{R}^{n_{\mathrm{dof},j}}$ denote the force and displacement column vectors of the j-th load case, $n_{\mathrm{dof},j} \in \mathbb{N}$ stands for the associated number of degrees of freedom, and $\mathbf{K}_j(\mathbf{a}) \in \mathbb{R}^{n_{\mathrm{dof},j} \times n_{\mathrm{dof},j}}$ is the corresponding symmetric positive semidefinite stiffness matrix. For these stiffness matrices, we require $\forall \mathbf{a} > \mathbf{0} : \mathbf{K}_j(\mathbf{a}) \succ 0$ to exclude rigid body motions. Using the finite element method, $\mathbf{K}_j(\mathbf{a})$ is assembled as

$$\mathbf{K}_j(\mathbf{a}) = \mathbf{K}_{j,0} + \sum_{i=1}^{n_{\mathrm{e}}} \left[\mathbf{K}^{(1)}_{j,i} a_i + \mathbf{K}^{(2)}_{j,i} a_i^2 + \mathbf{K}^{(3)}_{j,i} a_i^3 \right], \tag{3.14}$$

with $\mathbf{K}_{j,0} \succeq 0$ standing for a design-independent stiffness (such as fixed structural elements), $\mathbf{K}^{(1)}_{j,i} \succeq 0$ being the unit-cross-section-area membrane stiffness of the i-th element in the j-th load case, and $\mathbf{K}^{(2)}_{j,i} \succeq 0$ with $\mathbf{K}^{(3)}_{j,i} \succeq 0$ are the corresponding bending stiffness counterparts associated with the unit cross-section area. The force column vector $\mathbf{f}_j$ is assumed in the form

$$\mathbf{f}_j(\mathbf{a}) = \mathbf{f}_{j,0} + \sum_{i=1}^{n_{\mathrm{e}}} \left[\mathbf{f}^{(1)}_{j,i} a_i \right], \tag{3.15}$$

where $\mathbf{f}_{j,0}$ stands for the design-independent load and $\mathbf{f}^{(1)}_{j,i}$ are the design-dependent loads such as self-weight.

The formulation (3.13) is nonlinear and lacks convexity in general. The non-convexity comes not only from the polynomial entries in the stiffness matrix (3.14), but also from its possible singularity caused by zero cross-section areas (3.13d). But the feasible set of (3.13) is an intersection of a finite number of polynomial inequalities, forming a basic semi-algebraic set. However, using the moment-sum-of-square hierarchy, recall Section 3.2, to solve this formulation directly is inefficient because of three reasons: (i) the number of design variables is fairly large, leading to large moment matrices, and, therefore, long computational times and large memory requirements, (ii) the degree of the polynomials in the objective function and in the equilibrium equation is unnecessarily high, implying the use of higher-degree relaxations, and (iii) to satisfy the Assumption 3.2, one needs (preferably tight) bounds on the displacement variables.

3.3.1 Semidefinite programming formulation for topology optimization of frame structures

To avoid these shortcomings, we develop an approach to simplify (3.13) by eliminating the displacement variables $\mathbf{u}_j$ from the problem formulation. In the nested approach, which is commonly used in topology optimization, the cross-section areas are bounded from below by a strictly positive $\varepsilon \in \mathbb{R}_{>0}$, allowing for a computation of $[\mathbf{K}(\mathbf{a})]^{-1}$. Recall that $\mathbf{K}(\mathbf{a}) \succ 0$ for all $\mathbf{a} > \mathbf{0}$. The optimization procedure then traditionally adopts, e.g., the Method of Moving Asymptotes (MMA) [184], or the Optimality Criteria (OC) method [164]. Notice that $\varepsilon \to 0$ results in a high condition number of $\mathbf{K}(\mathbf{a})$ and that larger values of ε may impair quality of optimized designs, as sizing optimization is solved instead of the original topology optimization. In contrast, here, we eliminate $\mathbf{u}_j$ and allow the cross sections to truly attain zero by deriving a semidefinite programming formulation.

Compared to other, simpler derivations of this formulation, e.g., [3, Proposition 3.1] or [95, Proposition 3.1.5], our proof relies on the Moore-Penrose pseudo-inverse $[\mathbf{K}(\mathbf{a})]^{\dagger}$ which generalizes the possibly non-existent inverse $[\mathbf{K}(\mathbf{a})]^{-1}$. Our motivation for adopting a longer derivation was to reveal interconnections[3] between the formulation structure and the feasible upper bounds in Theorem 3.14.

We start by revealing the role of the Moore-Penrose pseudo-inverse in enforcing the equilibrium conditions.

Lemma 3.4. *Consider the equation* $\mathbf{K}_j(\mathbf{a})\mathbf{u}_j = \mathbf{f}_j(\mathbf{a}) - \mathbf{r}_j$ *with* $\mathbf{u}_j = \mathbf{K}_j(\mathbf{a})^{\dagger}\mathbf{f}_j(\mathbf{a})$ *and a residual vector* $\mathbf{r}_j \in \mathbb{R}^{n_{\mathrm{dof}}}$. *Then,* $\mathbf{r}_j = \mathbf{0}$ *if and only if* $\mathbf{f}_j(\mathbf{a}) \in \mathrm{Im}(\mathbf{K}_j(\mathbf{a}))$.

Proof. Let $\mathbf{f}_j(\mathbf{a}) = \mathbf{v}_j(\mathbf{a}) + \mathbf{w}_j(\mathbf{a})$, in which $\mathbf{v}_j(\mathbf{a}) \in \mathrm{Im}\left(\mathbf{K}_j(\mathbf{a})\right)$, and $\mathbf{w}_j(\mathbf{a}) \in \mathrm{Ker}\left(\mathbf{K}_j(\mathbf{a})\right)$. Then, because $\mathbf{K}_j(\mathbf{a})\mathbf{K}_j(\mathbf{a})^{\dagger}$ is an orthogonal projector onto the range of $\mathbf{K}_j(\mathbf{a})$, we obtain $\mathbf{K}_j(\mathbf{a})\mathbf{K}_j(\mathbf{a})^{\dagger}\mathbf{f}_j(\mathbf{a}) = \mathbf{v}_j(\mathbf{a})$. Clearly, when $\mathbf{f}_j(\mathbf{a}) \in \mathrm{Im}(\mathbf{K}_j(\mathbf{a}))$, we have $\mathbf{v}_j(\mathbf{a}) = \mathbf{f}(\mathbf{a})$ with $\mathbf{w}_j(\mathbf{a}) = \mathbf{0}$, implying that $\mathbf{r}_j = \mathbf{0}$. For the case of $\mathbf{f}_j(\mathbf{a}) \notin \mathrm{Im}(\mathbf{K}_j(\mathbf{a}))$, $\mathbf{w}_j(\mathbf{a}) \neq \mathbf{0}$, showing that $\mathbf{r}_j = -\mathbf{w}_j(\mathbf{a})$. □

Lemma 3.4 allows us to eliminate the displacement variables and write the optimization problem (3.13) only in terms of the cross-section areas $\mathbf{a}$ as

$$\min_{\mathbf{a}} \sum_{j=1}^{n_{\mathrm{lc}}} \omega_j \mathbf{f}_j(\mathbf{a})^{\mathrm{T}} \left[\mathbf{K}_j(\mathbf{a})\right]^{\dagger} \mathbf{f}_j(\mathbf{a}) \tag{3.16a}$$

$$\text{s.t.} \quad \overline{V} - \boldsymbol{\ell}^{\mathrm{T}}\mathbf{a} \geq 0, \tag{3.16b}$$

$$\mathbf{a} \geq \mathbf{0}, \tag{3.16c}$$

$$\mathbf{f}_j(\mathbf{a}) \in \mathrm{Im}(\mathbf{K}_j(\mathbf{a})),\ \forall j \in \{1, \ldots, n_{\mathrm{lc}}\}. \tag{3.16d}$$

Notice that (3.16d) essentially eliminates the nonphysical setup when $\mathbf{K}_j(\mathbf{a}) = [\mathbf{K}_j(\mathbf{a})]^{\dagger} = \mathbf{0}$ produces zero compliance.

Because $[\mathbf{K}_j(\mathbf{a})]^{\dagger} \succeq 0, \forall j \in \{1, \ldots, n_{\mathrm{lc}}\}$, and $\boldsymbol{\omega} > \mathbf{0}$ by definition, (3.16a) is bounded from below by 0. Thus, we introduce slack variables $\mathbf{c} \in \mathbb{R}^{n_{\mathrm{lc}}}_{\geq 0}$ comprising the to-be-minimized upper

[3] In particular, the transition from (3.17b) and (3.17e) to (3.21b) is reverse to how we approach Theorem 3.14 from Proposition 3.12.

bounds on compliances for the load cases and rewrite (3.16) equivalently as

$$\min_{\mathbf{a},\mathbf{c}} \boldsymbol{\omega}^{\mathrm{T}}\mathbf{c} \tag{3.17a}$$

$$\text{s.t. } c_j - \mathbf{f}_j(\mathbf{a})^{\mathrm{T}} \left[\mathbf{K}_j(\mathbf{a})\right]^{\dagger} \mathbf{f}_j(\mathbf{a}) \geq 0,\ \forall j \in \{1,\dots,n_{\mathrm{lc}}\}, \tag{3.17b}$$

$$\overline{V} - \boldsymbol{\ell}^{\mathrm{T}}\mathbf{a} \geq 0, \tag{3.17c}$$

$$\mathbf{a} \geq \mathbf{0}, \tag{3.17d}$$

$$\mathbf{f}_j(\mathbf{a}) \in \mathrm{Im}(\mathbf{K}_j(\mathbf{a})),\ \forall j \in \{1,\dots,n_{\mathrm{lc}}\}. \tag{3.17e}$$

To derive a nonlinear semidefinite programming formulation, let us now recall the generalized Schur complement lemma:

Lemma 3.5. *[67, Theorem 16.1] Let* $\mathbf{A}$ *and* $\mathbf{C}$ *be symmetric square matrices,* $\mathbf{B}$ *have appropriate dimensions, and* $\mathbf{I}$ *denote an identity matrix. Then, the following conditions are equivalent:*

1. $\begin{pmatrix} \mathbf{A} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & \mathbf{C} \end{pmatrix} \succeq 0,$
2. $\mathbf{C} \succeq 0,\ \mathbf{A} - \mathbf{B}^{\mathrm{T}}\mathbf{C}^{\dagger}\mathbf{B} \succeq 0,\ (\mathbf{I} - \mathbf{C}\mathbf{C}^{\dagger})\mathbf{B} = \mathbf{0}.$

Since we already have $\mathbf{K}_j(\mathbf{a}) \succeq 0$ by definition and $c_j - \left[\mathbf{f}_j(\mathbf{a})\right]^{\mathrm{T}} \left[\mathbf{K}_j(\mathbf{a})\right]^{\dagger} \mathbf{f}_j(\mathbf{a}) \geq 0$ in (3.17b), to use Lemma 3.5 it suffices to show that

$$(\mathbf{I} - \mathbf{K}_j(\mathbf{a}) \left[\mathbf{K}_j(\mathbf{a})\right]^{\dagger})\mathbf{f}_j(\mathbf{a}) = \mathbf{0}. \tag{3.18}$$

Proposition 3.6. *The condition* (3.18) *is equivalent to* $\mathbf{f}_j(\mathbf{a}) \in \mathrm{Im}(\mathbf{K}_j(\mathbf{a}))$.

Proof. First, consider $\mathbf{f}_j(\mathbf{a}) \in \mathrm{Im}(\mathbf{K}_j(\mathbf{a}))$. Then, $\mathbf{f}_j(\mathbf{a}) = \mathbf{K}_j(\mathbf{a})\mathbf{u}_j$ for some displacement vector $\mathbf{u}_j$. After inserting it into the left-hand-side of (3.18), we have

$$\left(\mathbf{K}_j(\mathbf{a}) - \mathbf{K}_j(\mathbf{a}) \left[\mathbf{K}_j(\mathbf{a})\right]^{\dagger} \mathbf{K}_j(\mathbf{a})\right) \mathbf{u}_j = \mathbf{0}, \tag{3.19}$$

which holds for all such $\mathbf{u}_j$ as $\mathbf{K}_j(\mathbf{a}) \left[\mathbf{K}_j(\mathbf{a})\right]^{\dagger} \mathbf{K}_j(\mathbf{a}) = \mathbf{K}_j(\mathbf{a})$ by the definition of the Moore-Penrose pseudo-inverse [67, Lemma 14.1].

Otherwise, consider $\mathbf{f}_j(\mathbf{a}) \notin \mathrm{Im}(\mathbf{K}_j(\mathbf{a}))$ and let $\tilde{\mathbf{u}} = \left[\mathbf{K}_j(\mathbf{a})\right]^{\dagger} \mathbf{f}_j(\mathbf{a})$. Then, $\mathbf{K}_j(\mathbf{a})\tilde{\mathbf{u}} = \mathbf{f}_j(\mathbf{a}) - \mathbf{r}_j$ for some $\mathbf{r}_j \in \mathrm{Ker}(\mathbf{K}(\mathbf{a}))$, $\mathbf{r}_j \neq \mathbf{0}$ by Lemma 3.4. Thus, the left-hand-side of (3.18) simplifies to

$$\mathbf{f}_j(\mathbf{a}) - \mathbf{K}_j(\mathbf{a}) \left[\mathbf{K}_j(\mathbf{a})\right]^{\dagger} \mathbf{f}_j(\mathbf{a}) = \mathbf{f}_j(\mathbf{a}) - \mathbf{K}_j(\mathbf{a})\tilde{\mathbf{u}} = \mathbf{r}_j \neq \mathbf{0}, \tag{3.20}$$

which completes the proof. □

Finally, Proposition 3.6 and Lemma 3.5 facilitate an equivalent reformulation of the optimization problem (3.17) as a nonlinear semidefinite program

$$\min_{\mathbf{a},\mathbf{c}} \ \boldsymbol{\omega}^{\mathrm{T}}\mathbf{c} \tag{3.21a}$$

$$\text{s.t. } \begin{pmatrix} c_j & -\mathbf{f}_j(\mathbf{a})^{\mathrm{T}} \\ -\mathbf{f}_j(\mathbf{a}) & \mathbf{K}_j(\mathbf{a}) \end{pmatrix} \succeq 0,\ \forall j \in \{1,\dots,n_{\mathrm{lc}}\}, \tag{3.21b}$$

$$\overline{V} - \boldsymbol{\ell}^{\mathrm{T}}\mathbf{a} \geq 0, \tag{3.21c}$$

$$\mathbf{a} \geq \mathbf{0}, \tag{3.21d}$$

in which only the constraint (3.21b) lacks convexity. Importantly, all constraints are polynomial functions of $\mathbf{a}$, forming therefore a semi-algebraic feasible set.

3.3.2 Efficient polynomial reformulation

The optimization problem (3.21) constitutes a minimization of a linear function over a semi-algebraic set, allowing for a solution using the moment-sum-of-squares hierarchy, as briefly discussed in Section 3.2. However, efficiency of the hierarchy can be improved after modifying (3.21) to provide a tighter feasible set of relaxed problems and to reduce numerical issues by scaling the design variables. These modifications are outlined in the following paragraphs.

3.3.2.1 Compactness of the feasible set

We start by enforcing compactness of the feasible set of the optimization problem (3.21) because of two reasons. First, compactness is required for Theorem 3.3, in the form of Assumption 3.2. Second, compactness also allows tightening the feasible sets of relaxed problems, notably improving numerical performance.

Proposition 3.7. *Assume that* $\mathbf{a}^*$ *and* $\mathbf{c}^*$ *are optimal cross-section areas and compliances associated with the optimization problem* (3.21)*. Then,* $\forall i \in \{1, \dots, n_e\} : 0 \leq a_i^* \leq \overline{a}_i$ *with* $\overline{a}_i = \overline{V}/\ell_i$ *and* $\forall j \in \{1, \dots, n_{\mathrm{lc}}\} : 0 \leq c_j^* \leq \overline{c}/\omega_j$*, where* $\overline{c} = \sum_{j=1}^{n_{\mathrm{lc}}} \left[\omega_j \mathbf{f}(\hat{\mathbf{a}})^{\mathrm{T}} \mathbf{K}(\hat{\mathbf{a}})^{-1} \mathbf{f}(\hat{\mathbf{a}})\right]$ *with* $\hat{\mathbf{a}} = \mathbf{1}\overline{V}/\sum_{i=1}^{n_e} \ell_i$*.*

Proof. The cross-section areas are non-negative by definition (3.21d). Therefore, (3.21c) represents a conic combination and none of the structural elements can occupy a larger volume than the volume bound $\overline{V}$, $\forall i \in \{1, \dots, n_e\} : a_i^* \leq \overline{V}/\ell_i$.

The compliance variables are placed at the main diagonal of the polynomial matrix inequality (PMI) (3.21b) and are hence non-negative, $c_j^* \geq 0$. Then, because $\boldsymbol{\omega} > \mathbf{0}$, the conic combination $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}^*$ is an upper bound for its summands, $\omega_j c_j^* \leq \boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}^*$. Moreover, since $\hat{\mathbf{a}}$ determines uniquely the compliances $\hat{\mathbf{c}}$, $\hat{c}_j = \mathbf{f}_j(\hat{\mathbf{a}})^{\mathrm{T}}\mathbf{K}_j(\hat{\mathbf{a}})^{-1}\mathbf{f}_j(\hat{\mathbf{a}})$, the pair $(\hat{\mathbf{a}}, \hat{\mathbf{c}})$ is a feasible solution to (3.21b)–(3.21d), so that we also have $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}^* \leq \overline{c} = \boldsymbol{\omega}^{\mathrm{T}}\hat{\mathbf{c}}$. Consequently, $\forall j \in \{1, \dots, n_{\mathrm{lc}}\} : \omega_j c_j^* \leq \overline{c}$. □

Among the bounds in Proposition 3.7, only the compliance upper bounds are not enforced in the formulation (3.21). Indeed, for any fixed $\mathbf{a} > \mathbf{0}$, $\mathbf{c} \to \boldsymbol{\infty}$ is feasible to (3.21), so that Assumption 3.2 is not satisfied. To make the design space bounded, we add the (redundant) upper-bound compliance constraint from Proposition 3.7, or, eventually, an upper-bound obtained by solving the convex truss topology optimization problem instead, see Appendix 3.A. Subsequently, we arrive at the optimization problem

$$\min_{\mathbf{a},\mathbf{c}} \ \boldsymbol{\omega}^{\mathrm{T}}\mathbf{c} \tag{3.22a}$$

$$\text{s.t.} \begin{pmatrix} c_j & -\mathbf{f}_j(\mathbf{a})^{\mathrm{T}} \\ -\mathbf{f}_j(\mathbf{a}) & \mathbf{K}_j(\mathbf{a}) \end{pmatrix} \succeq 0, \ \forall j \in \{1, \dots, n_{\mathrm{lc}}\}, \tag{3.22b}$$

$$\overline{V} - \boldsymbol{\ell}^{\mathrm{T}}\mathbf{a} \geq 0, \tag{3.22c}$$

$$\overline{c} - \boldsymbol{\omega}^{\mathrm{T}}\mathbf{c} \geq 0, \tag{3.22d}$$

$$\mathbf{a} \geq \mathbf{0}, \tag{3.22e}$$

for which we have the following result:

Proposition 3.8. *The feasible set of* (3.22) *is compact.*

Proof. The feasible set is bounded based on Proposition 3.7. Moreover, $\mathbf{a}$ and $\mathbf{c}$ satisfying conditions (3.22c)–(3.22e) form a closed set. Thus, it suffices to show that (3.22b) is closed. But the elements in (3.22b) are polynomial functions that are continuous. Moreover, the set of semidefinite matrices is closed so $\mathbf{a}$ and $\mathbf{c}$ satisfying (3.22b) live in a closed set. Boundedness and closeness imply compactness because we are in a finite dimensional space. □

3.3.2.2 Scaling and box constraints

After introducing box constraints in formulation (3.22), we can scale all variables domains to $[-1, 1]$. This scaling reduces numerical issues that may arise during the problem solution. To this goal, we have

$$c_j = \frac{1}{2\omega_j}\left(c_{\mathrm{s},j} + 1\right)\bar{c}, \qquad \forall j \in \{1, \dots, n_{\mathrm{lc}}\}, \tag{3.23a}$$

$$a_i = 0.5\left(a_{\mathrm{s},i} + 1\right)\bar{a}_i, \qquad \forall i \in \{1, \dots, n_{\mathrm{e}}\}, \tag{3.23b}$$

where $\mathbf{a}_{\mathrm{s}}$ and $\mathbf{c}_{\mathrm{s}}$ are the scaled cross-section areas and compliance variables.

In addition, we explicitly insert the box constraints into the optimization problem formulation to tighten feasible sets of the relaxed problems. There are multiple options how to write these box constraints $\mathbf{a}_{\mathrm{s}}, \mathbf{c}_{\mathrm{s}} \in [-1, 1]$, e.g.,

$$\begin{aligned} -1 \le a_{\mathrm{s},i} \le 1, &\quad \forall i \in \{1, \dots, n_{\mathrm{b}}\}, \\ -1 \le c_{\mathrm{s},j} \le 1, &\quad \forall j \in \{1, \dots, n_{\mathrm{lc}}\}, \end{aligned} \tag{3.24a}$$

$$\begin{aligned} a_{\mathrm{s},i}^2 \le 1, &\quad \forall i \in \{1, \dots, n_{\mathrm{b}}\}, \\ c_{\mathrm{s},j}^2 \le 1, &\quad \forall j \in \{1, \dots, n_{\mathrm{lc}}\}. \end{aligned} \tag{3.24b}$$

Despite equivalent in what they enforce, their numerical performance in the moment-sum-of-squares hierarchy varies considerably; we refer the reader to the recent note of Anjos *et al.* [10] and the examples in Appendix 3.B. Here, we use the quadratic bounds (3.24b). Then, the optimization problem reads

$$\min_{\mathbf{a}_{\mathrm{s}}, \mathbf{c}_{\mathrm{s}}} \sum_{j=1}^{n_{\mathrm{lc}}} \left[0.5\left(c_{\mathrm{s},j} + 1\right)\bar{c}\right] \tag{3.25a}$$

$$\text{s.t.} \begin{pmatrix} \frac{1}{2\omega_j}\left(c_{\mathrm{s},j} + 1\right)\bar{c} & -\mathbf{f}_j(\mathbf{a}_{\mathrm{s}})^{\mathrm{T}} \\ -\mathbf{f}_j(\mathbf{a}_{\mathrm{s}}) & \mathbf{K}_j(\mathbf{a}_{\mathrm{s}}) \end{pmatrix} \succeq 0, \ \forall j \in \{1, \dots, n_{\mathrm{lc}}\}, \tag{3.25b}$$

$$2 - n_{\mathrm{e}} - \mathbf{1}^{\mathrm{T}}\mathbf{a}_{\mathrm{s}} \ge 0, \tag{3.25c}$$

$$1 - \boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{s}} \ge 0, \tag{3.25d}$$

$$\text{bound constraints (3.24b)}. \tag{3.25e}$$

Because the feasible set of (3.25) is compact by Proposition 3.8, one may tempt to add a redundant polynomial inequality constraint to satisfy Assumption 3.2. However, the assumption is already satisfied in our case.

Proposition 3.9. *The optimization problem* (3.25) *satisfies Assumption 3.2.*

Proof. Let $\mathbf{G}(\mathbf{a}_{\mathrm{s}}, \mathbf{c}_{\mathrm{s}})$ be a block-diagonal matrix with the blocks (3.25b)–(3.25e) and let $\mathbf{H}$ be a sparse matrix of the same dimensions with the structure

$$\mathbf{H} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}, \tag{3.26}$$

in which the identity matrix $\mathbf{I} \in \mathbb{S}^{n_\mathrm{e}+n_\mathrm{lc}}$ matches the positions of (3.25e) in $\mathbf{G}(\mathbf{a}_\mathrm{s}, \mathbf{c}_\mathrm{s})$. Clearly, $\mathbf{H}$ is a SOS because of $\mathbf{H} = \mathbf{H}\mathbf{H}^\mathrm{T}$, recall Definition 3.1. Then, $p = \langle \mathbf{H}\mathbf{H}^\mathrm{T}, \mathbf{G}(\mathbf{a}_\mathrm{s}, \mathbf{c}_\mathrm{s}) \rangle = n_\mathrm{e} + n_\mathrm{lc} - \sum_{i=1}^{n_\mathrm{e}} a_{\mathrm{s},i}^2 - \sum_{j=1}^{n_\mathrm{lc}} c_{\mathrm{s},j}^2$, so that the level set $\{\mathbf{a}_\mathrm{s} \in \mathbb{R}^{n_\mathrm{e}}, \mathbf{c}_\mathrm{s} \in \mathbb{R}^{n_\mathrm{lc}} \mid p_{(3.24\mathrm{b})} \geq 0\}$ is compact. □

Remark 3.10. The constraints (3.24b) are tighter in the moment representation than (3.24a).

To see this, assume that (y_0, y_1, y_2) are the moments associated with the canonical basis of the vector space of polynomials of degree at most four, $(1, x, x^2)$, where one can substitute x by any element of $\mathbf{a}_\mathrm{s}$ or $\mathbf{c}_\mathrm{s}$. Then, in the first relaxation of the moment-sum-of-squares hierarchy, the quadratic constraint $1 - x^2 \geq 0$ becomes

$$y_0 - y_2 \geq 0, \tag{3.27}$$

and the box constraint $-1 \leq x \leq 1$ provides

$$-y_0 \leq y_1 \leq y_0, \tag{3.28}$$

with $y_0 = 1$. Moreover, the localizing matrix of the entire optimization problem contains the principal submatrix

$$\begin{pmatrix} y_0 & y_1 \\ y_1 & y_2 \end{pmatrix} \succeq 0, \tag{3.29}$$

that must be positive semi-definite as the entire localizing matrix is.

For the quadratic constraints (3.24b), $y_2 \leq 1$ from Eq. (3.27) and $y_2 \geq 0$ because of Eq. (3.29). Writing the determinant of (3.29) then provides us with $y_1^2 \leq y_2$. Consequently, we observe that $0 \leq y_1^2 \leq y_2 \leq 1$.

In the case of pure box constraints (3.24a), we only have $0 \leq y_1^2 \leq 1$, Eq. (3.28), and $y_1^2 \leq y_2$, Eq. (3.29). Note that there is no upper bound for y_2, which can attain arbitrarily large values in the first relaxation. From the mechanical point of view, this allows for an arbitrarily-large rotational stiffnesses $\mathbf{K}_{j,i}^{(2)} a_i^2$ of the elements.

These observations then allow us to show feasibility of the first-order moments for (3.25c)–(3.25e):

Proposition 3.11. *Let $\mathbf{y}_{\mathbf{c}^1}^*$ and $\mathbf{y}_{\mathbf{a}^1}^*$ be the first-order moments associated with the variables $\mathbf{c}_\mathrm{s}$ and $\mathbf{a}_\mathrm{s}$ obtained from a solution to any relaxation of* (3.25) *using the moment-sum-of-squares hierarchy. Then, these moments satisfy*

$$2 - n_\mathrm{e} - \mathbf{1}^\mathrm{T} \mathbf{y}_{\mathbf{a}^1}^* \geq 0, \tag{3.30a}$$

$$1 - \boldsymbol{\omega}^\mathrm{T} \mathbf{y}_{\mathbf{c}^1}^* \geq 0, \tag{3.30b}$$

$$1 - \left(y_{a_i^1}^*\right)^2 \geq 0, \tag{3.30c}$$

$$1 - \left(y_{c_j^1}^*\right)^2 \geq 0. \tag{3.30d}$$

Proof. (3.30a) and (3.30b) hold trivially from construction of the hierarchy. (3.30c)–(3.30d) follow from Remark 3.10. □

3.3.3 Recovering feasible upper-bound solutions

In Proposition 3.11, we have shown that the first-order moments obtained by solving any relaxation of the moment-sum-of-squares hierarchy satisfy all the constraints of (3.25) except for (3.25b). This section is therefore devoted to the question how to "correct" these moments to produce feasible upper-bounds to the original problem (3.21) and provide a natural sufficient condition of global optimality.

We start by proving the following essential result:

Proposition 3.12. *Let $\mathbf{y}^*_{\mathbf{c}^1}$ and $\mathbf{y}^*_{\mathbf{a}^1}$ be the first-order moments associated with the variables $\mathbf{c}_\mathrm{s}$ and $\mathbf{a}_\mathrm{s}$ obtained from a solution to any relaxation of (3.25) using the moment-sum-of-squares hierarchy and let $\forall i \in \{1,\dots,n_\mathrm{e}\} : \tilde{a}_i = 0.5(y^*_{a_i^1}+1)\overline{a}$ be the corresponding cross-section areas. Then,*

$$\mathbf{f}_{j,0} + \sum_{i=1}^{n_\mathrm{e}} \mathbf{f}_{j,i}\tilde{a}_i \in \mathrm{Im}\left(\mathbf{K}_{j,0} + \sum_{i=1}^{n_\mathrm{e}}\sum_{k=1}^{3} \mathbf{K}^{(k)}_{j,i}\tilde{a}_i^k\right). \tag{3.31}$$

Proof. In the lowest relaxation of the moment-sum-of-squares hierarchy, the PMI constraint (3.25b) becomes

$$\begin{pmatrix} \frac{1}{2\omega_j}\left(y^*_{c_j^1}+1\right)\overline{c} & -\mathbf{f}_j^\mathrm{T}\left(\mathbf{y}^*_{\mathbf{a}^1}\right)^\mathrm{T} \\ -\mathbf{f}_j\left(\mathbf{y}^*_{\mathbf{a}^1}\right) & \mathbf{K}_j\left(\mathbf{y}^*_{\mathbf{a}^1},\mathbf{y}^*_{\mathbf{a}^2},\mathbf{y}^*_{\mathbf{a}^3}\right) \end{pmatrix} \succeq 0, \tag{3.32}$$

where $\mathbf{y}^*_{\mathbf{a}^2}$ and $\mathbf{y}^*_{\mathbf{a}^3}$ are the second- and third-order moments associated with $\mathbf{a}_\mathrm{s}$ and, with a slight abuse of notation, $\mathbf{K}_j\left(\mathbf{y}^*_{\mathbf{a}^1},\mathbf{y}^*_{\mathbf{a}^2},\mathbf{y}^*_{\mathbf{a}^3}\right)$ and $\mathbf{f}_j\left(\mathbf{y}^*_{\mathbf{a}^1}\right)$ are the stiffness matrix and force column vector constructed from the moments $\mathbf{y}$. Using Lemma 3.5 and Proposition 3.6, we observe that

$$\mathbf{f}(\mathbf{y}^*_{\mathbf{a}^1}) \in \mathrm{Im}\left(\mathbf{K}_j\left(\mathbf{y}^*_{\mathbf{a}^1},\mathbf{y}^*_{\mathbf{a}^2},\mathbf{y}^*_{\mathbf{a}^3}\right)\right). \tag{3.33}$$

Because we have considered solely degree-one moments in (3.31) and $\forall \mathbf{a} > \mathbf{0} : \mathbf{K}_j(\mathbf{a}) \succ 0$ was our initial assumption, we must show that the combination of $a_i = 0$ with $I_i > 0$ cannot occur for any i, because that would result in a lower rank of $\mathbf{K}_j(\tilde{\mathbf{a}})$ when compared with $\mathbf{K}_j\left(\mathbf{y}^*_{\mathbf{a}^1},\mathbf{y}^*_{\mathbf{a}^2},\mathbf{y}^*_{\mathbf{a}^3}\right)$.

To this goal, let $a_i = 0$, which is equivalent to $y^*_{a_i^1} = -1$. Then, the non-negative determinant of (3.29) and the inequalities (3.24b) imply that $y^*_{a_i^2} = 1$. For $c_\mathrm{III} > 0$, we also need higher-order terms. In the lowest, second relaxation, the inequality (3.24b) implies that

$$\begin{pmatrix} 1-y^*_{a_i^2} & y^*_{a_i}-y^*_{a_i^3} \\ y^*_{a_i}-y^*_{a_i^3} & y^*_{a_i^2}-y^*_{a_i^4} \end{pmatrix} \succeq 0 \tag{3.34}$$

and the moment matrix contains the principal submatrix

$$\begin{pmatrix} 1 & y^*_{a_i^1} & y^*_{a_i^2} \\ y^*_{a_i^1} & y^*_{a_i^2} & y^*_{a_i^3} \\ y^*_{a_i^2} & y^*_{a_i^3} & y^*_{a_i^4} \end{pmatrix} \succeq 0 \tag{3.35}$$

When setting $y^*_{a_i} = -1$, it can easily be verified that (3.34) and (3.35) are feasible if and only if $y^*_{a_i^3} = -1$ and $y^*_{a_i^4} = 1$.

Thus, we write the moment of inertia in terms of the scaled cross-section areas (3.23b). After inserting the moments, we obtain

$$I_i = 0.25 c_{\mathrm{II}} \overline{a}^2 \left(y^*_{a_i^2} + 2 y^*_{a_i^1} + 1 \right) + 0.125 c_{\mathrm{III}} \overline{a}^3 \left(y^*_{a_i^3} + 3 y^*_{a_i^2} + 3 y^*_{a_i^1} + 1 \right) = 0. \tag{3.36}$$

□

Remark 3.13. A similar result to Proposition 3.12 can also be derived for (3.24a), but it relies on relaxations of degree two even for quadratic moments of inertia.

Using Proposition 3.12, we can correct $\mathbf{c}$ based on $\mathbf{y}^*_{\mathbf{a}^1}$ to provide a feasible solution to (3.21). Notice that the following result would be weaker[4] in the case of linear constraints (3.24a) due to Remark 3.13.

Theorem 3.14. *Let* $\mathbf{y}^*_{\mathbf{c}^1}$ *and* $\mathbf{y}^*_{\mathbf{a}^1}$ *be the first-order moments associated with the variables* $\mathbf{c}_{\mathrm{s}}$ *and* $\mathbf{a}_{\mathrm{s}}$ *obtained from a solution to any relaxation of* (3.25) *using the moment-sum-of-squares hierarchy. Then,*

$$\tilde{a}_i = 0.5(y^*_{a_i^1} + 1)\overline{a}_i, \qquad \forall i \in \{1, \dots, n_{\mathrm{e}}\}, \tag{3.37a}$$

$$\tilde{c}_j = [\mathbf{f}_j(\tilde{\mathbf{a}})]^{\mathrm{T}} \mathbf{K}_j^{\dagger}(\tilde{\mathbf{a}}) \mathbf{f}_j(\tilde{\mathbf{a}}), \qquad \forall j \in \{1, \dots, n_{\mathrm{lc}}\} \tag{3.37b}$$

is feasible (upper-bound) to (3.21).

Proof. Based on Proposition 3.11, $\tilde{\mathbf{a}}$ satisfies the constraints imposed on the cross-section areas. By correcting the compliance variables according to (3.37b), the equilibrium equation (and so the PMI (3.21b)) is satisfied due to Proposition 3.12. Consequently, all the constraints of (3.21) are feasible for the pair $\tilde{\mathbf{a}}$, $\tilde{\mathbf{c}}$, showing that $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}^* \leq \boldsymbol{\omega}^{\mathrm{T}}\tilde{\mathbf{c}} < \infty$. □

We wish to emphasize that in Theorem 3.14, we have proved feasibility of the upper bounds to (3.21) and such upper bounds may violate the compliance bound constraints (3.25d). Thus, knowledge of $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}^*$ does not assure convergence of the lowest relaxation to the optimal cross-section areas.

3.3.4 Certificate of global ε-optimality

Because the hierarchy generates a sequence of lower bounds and we have just shown in Theorem 3.14 how to compute upper bounds in each relaxation, we naturally arrive at a simple sufficient condition of global ε-optimality.

Lemma 3.15. *Let* $\mathbf{y}^*_{\mathbf{c}^1}$ *and* $\mathbf{y}^*_{\mathbf{a}^1}$ *be the first-order moments associated with the variables* $\mathbf{c}_{\mathrm{s}}$ *and* $\mathbf{a}_{\mathrm{s}}$ *obtained from a solution to any relaxation of* (3.25) *using the moment-sum-of-squares hierarchy. Then,*

$$\boldsymbol{\omega}^{\mathrm{T}}\tilde{\mathbf{c}} - \sum_{j=1}^{n_{\mathrm{lc}}} \left[0.5(y^*_{c_j^1} + 1)\overline{c} \right] \leq \varepsilon \tag{3.38}$$

is a sufficient condition of global ε-optimality.

Lemma 3.15 is very simple to verify computationally, significantly simpler than the traditional rank-based certificate of global optimality (3.8), e.g., [85]. However, (3.38) fails to be

[4] The lowest relaxation might not produce feasible upper bounds. For example, consider $c_{\mathrm{II}} > 0, c_{\mathrm{III}} = 0$, and a one-element cantilever beam with one end fully clamped and the other carrying a moment load. Then, $y_{a_1} = -1$, $y_{a_1^2} = \infty$ with $y_{c_1} = -1$ belongs to the set of optimal solutions to the first relaxation.

a necessary condition. Indeed, the optimality gap ε may remain strictly positive even when the hierarchy converged according to (3.8) in the case of multiple globally optimal solutions. Then, the optimal first-order moments $\mathbf{y}$ are not unique; for instance, they may correspond to any convex combination of the global optima, we refer to Section 3.4.2 for a specific example.

A stronger result holds, however, when the optimization problem possesses a unique global optimum. To show this, we first prove that, with an increasing relaxation degree r, the feasible space of relaxations converges to the convex hull of the initial (non-convex) problem.

Proposition 3.16. *Let $\mathcal{K}^{(r)}$ be the feasible set of the first-order moments in the r-th relaxation of the moment-sum-of-squares hierarchy of* (3.25)*. Then, $\mathcal{K}^{(r)} \nearrow \mathrm{conv}(\mathcal{K})$ as $r \to \infty$, where $\mathcal{K}$ is the intersection of* (3.25b)–(3.25e)*.*

Proof. Let $f(\mathbf{a}_\mathrm{s}, \mathbf{c}_\mathrm{s})$ be an arbitrary affine function. Based on Proposition 3.9, Assumption 3.2 holds for (3.25) independently of the objective function. Hence, optimization of $f(\mathbf{a}_\mathrm{s}, \mathbf{c}_\mathrm{s})$ over $\mathcal{K}$ yields $f(\mathbf{a}_\mathrm{s}, \mathbf{c}_\mathrm{s}) \nearrow f^*(\mathbf{a}_\mathrm{s}, \mathbf{c}_\mathrm{s})$ as $r \to \infty$ due to Theorem 3.3. Because $f(\mathbf{a}_\mathrm{s}, \mathbf{c}_\mathrm{s})$ is arbitrary, $\mathcal{K}^{(r)} \nearrow \mathrm{conv}(\mathcal{K})$ as $r \to \infty$. □

Finally, we can prove that the hierarchy eventually attains a zero optimality gap.

Theorem 3.17. *If there is a unique global solution to* (3.25)*, then*

$$\boldsymbol{\omega}^\mathrm{T}\tilde{\mathbf{c}} - \sum_{j=1}^{n_\mathrm{lc}} \left\lceil 0.5(y_{c_j^1}^{(r)*} + 1)\overline{c} \right\rceil = 0 \tag{3.39}$$

as $r \to \infty$.

Proof. Assuming $r \to \infty$, optimization of (3.25a) over $\mathcal{K}^{(r)}$ is equivalent to optimization of (3.25a) over $\mathrm{conv}(\mathcal{K})$ by Proposition 3.16. Because $\mathcal{K}$ is compact, its convex hull must be also compact. Hence, it can be equivalently expressed as the convex hull of the limit points of $\mathcal{K}$ that are denoted by $\mathbf{d}_1, \mathbf{d}_2, \ldots$, i.e.,

$$\mathrm{conv}(\mathcal{K}) = \mathrm{conv}(\cup_{i=1}^{\infty}\{\mathbf{d}_i\}). \tag{3.40}$$

Because we assume there is the unique global optimum when optimizing over $\mathcal{K}$, there must be a unique limit point $\mathbf{d}^*$ associated with this optimum. □

Remark 3.18. Although Theorem 3.17 relies on $r \to \infty$, a finite (and fairly small) r is required in all our test cases to reach the zero optimality gap, up to the numerical precision of the semidefinite programming solver. Moreover, this bound equality has occurred when the hierarchy converged based on the rank test (3.8). It might be possible, therefore, to strengthen Theorem 3.17 to a finite termination result.

3.3.5 Global topology optimization of shell structures

Until now, solely frame structures have been considered. However, the optimization formulations (3.21) and (3.25) allow for simple modifications to optimize other discrete structures such as shells. Let $\mathbf{t} \in \mathbb{R}^{n_\mathrm{e}}_{\geq 0}$ be the vector of shell element thicknesses. Then, the formulation

(3.21) becomes

$$\min_{\mathbf{t},\mathbf{c}} \ \boldsymbol{\omega}^{\mathrm{T}}\mathbf{c} \tag{3.41a}$$

$$\text{s.t.} \ \begin{pmatrix} c_j & -\mathbf{f}_j(\mathbf{t})^{\mathrm{T}} \\ -\mathbf{f}_j(\mathbf{t}) & \mathbf{K}_j(\mathbf{t}) \end{pmatrix} \succeq 0, \forall j \in \{1,\dots,n_{\mathrm{lc}}\}, \tag{3.41b}$$

$$\overline{V} - \mathbf{s}^{\mathrm{T}}\mathbf{t} \geq 0, \tag{3.41c}$$

$$\mathbf{t} \geq \mathbf{0}, \tag{3.41d}$$

where $\mathbf{s} \in \mathbb{R}^{n_{\mathrm{e}}}_{>0}$ is a vector of the surface areas of individual shell elements, and $\mathbf{K}_j(\mathbf{t})$ is assembled as

$$\mathbf{K}_j(\mathbf{t}) = \mathbf{K}_{j,0} + \sum_{i=1}^{n_{\mathrm{e}}} \left[\mathbf{K}^{(1)}_{j,i} t_i + \mathbf{K}^{(3)}_{j,i} t_i^3 \right]. \tag{3.42}$$

Because the design variables can be bounded very similarly to Proposition 3.7 and scaled, all proven results hold true.

3.4 Sample problems

This section investigates global topology optimization of selected small-scale structural design problems using the proposed strategy solved numerically by the MOSEK optimizer [138]. These examples demonstrate strengths and weaknesses of the presented approach: certificate of global ε-optimality using Lemma 3.15, extraction[5] of all guaranteed globally optimal solutions based on the flat extension theorem [45], but also higher computational demands when compared to selected local optimization techniques: OC and MMA adopting the nested approach, see, e.g., [22], MATLAB's inbuilt optimizer `fmincon` solving (3.13) directly, and non-linear semidefinite programming (NSDP) formulation (3.21) solved by the PENLAB optimizer [62].

For the OC and MMA methods, we adopt the commonly used starting points of a uniform mass distribution; for the `fmincon` and PENLAB optimizers, we rely on the default starting points. We note here that different starting points may lead to very different structural topologies and varying compliances, even to a global optimum. Hence, the comparison of our approach with these local methods shall be seen from the perspective of computational scalability and certified optimality, and the resulting designs as specific local optima, rather than as a systematic comparison of optimized objective functions. Except for the nested approaches, all optimization problems were modeled using the YALMIP toolbox [131]. Our implementation and the corresponding source codes written in MATLAB can be accessed at [195].

The first three examples involve two finite elements only to allow visualization of the feasible sets and provide intuition about the solution approach. In the later part, we investigate the influence of finite element types on optimal design and increase the number of elements to evaluate scalability of the approach. All computations were performed on a personal laptop with 16 GB of RAM and Intel® Core™ i5-8350U CPU. Times of individual optimization approaches are measured to allow a simple comparison of the computational demands.

[5] For rank computation we considered the eigenvalues with the absolute value smaller than 10^{-8} to be singular.

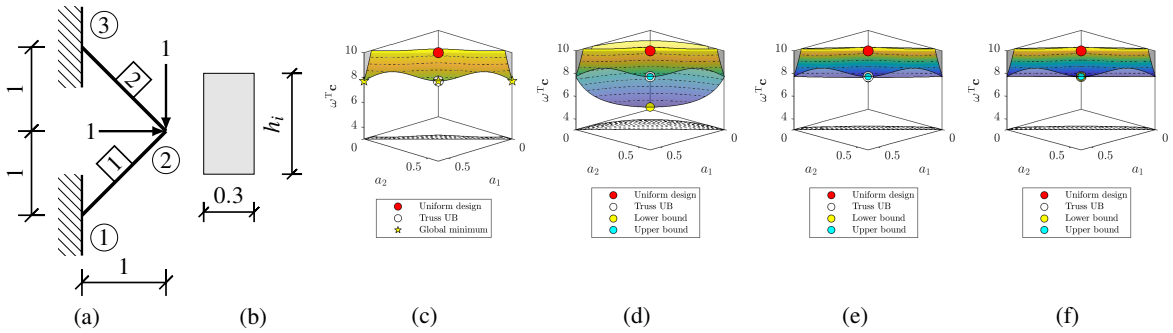


Figure 3.2: Frame structure composed of two elements: (a) boundary conditions, (b) the cross-section parametrization, and the sub-level set $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c} \leq 10$ of the (c) feasible space and of the (d) second, (e) third, and (f) fourth outer approximations with the associated lower- and upper-bounds. Variables a_1 and a_2 stand for the cross-section areas of the two elements and $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}$ denotes the corresponding weighted compliance of the two load cases (assuming the moments of inertia $I_i = 25/27a_i^3$, $i \in \{1,2\}$), and h_i is the cross-section height.

3.4.1 Structure possessing multiple global optima

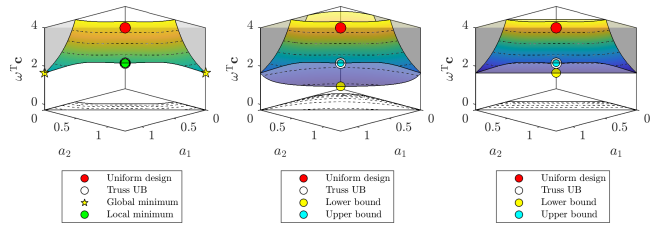
As the first problem, we consider a frame structure composed of two Euler-Bernoulli frame elements, see Fig. 3.2a. Two loads are applied, each of them acting as a separate load case, and weighted equally by $\boldsymbol{\omega} = \mathbf{1}$. Both these frame elements posses the Young modulus $E = 1$, and their overall volume is bounded by $\overline{V} = 0.816597322$ from above[6]. Accordingly with Fig. 3.2b, the elements $i = \{1,2\}$ have rectangular cross-sections with areas $a_i = 0.3h_i$. Then, $I_i = \frac{1}{40}h^3$, which implies that $c_{\mathrm{II}} = 0$ and $c_{\mathrm{III}} = 25/27$ in Eq. (3.12).

The feasible domain of the optimization problem shown in Fig. 3.2c reveals that there are three global optima of the objective function value 7.738, corresponding to the following cases: (i) $a_1^* = \overline{V}/\sqrt{2}$ and $a_2^* = 0$, (ii) $a_1^* = 0$ and $a_2^* = \overline{V}/\sqrt{2}$, and (iii) $a_1^* = a_2^* = \overline{V}\sqrt{2}/4$. All these solutions are extracted in the fourth relaxation of the moment-sum-of-squares hierarchy (Fig. 3.2f), which converged based on the rank condition (3.8) with the rank equal to $s = 3$ and also based on Lemma 3.15, $\varepsilon = 5 \times 10^{-10}$. Notice that in all the relaxations, the upper bounds

[6] Fewer digits may prevent the solver from reaching all three global optima. Although an analytical formula for this specific $\overline{V}$ can be derived, we omit it for the sake of brevity.

method	a_1	a_2	LB	UB	time [s]	$n_{\mathrm{c}} \times m$	n
OC	0.289	0.289	-	7.738	0.009	-	2
MMA	0.289	0.289	-	7.738	0.011	-	2
`fmincon`	0.289	0.289	-	7.738	0.113	-	8
NSDP	0.289	0.289	-	7.738	0.409	2×4	4
$\mathrm{PO}^{(2)}$, Lemma 3.15	0.289	0.289	5.065	7.738	0.018	1×15, 6×5, 2×4	69
$\mathrm{PO}^{(3)}$, Lemma 3.15	0.289	0.289	7.647	7.738	0.070	1×35, 6×15, 2×20	209
$\mathrm{PO}^{(4)}$, Lemma 3.15	0.289	0.289	7.738	7.738	0.545	1×70, 6×35, 2×60	494

Table 3.1: Different optimization methods applied to the first optimization problem. LB denotes lower bound, UB abbreviates feasible upper bounds, and PO stands for polynomial optimization. In addition, n_{c} stands for the number of semidefinite constraints of the size m and n is the number of variables. The entries a_i denote cross-section areas of the i-th element, or the areas constructed from the first-order moments in the case of PO.



(a) (b) (c)

Figure 3.3: Frame structure possessing a non-zero optimality gap. The sub-level set $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c} \leq 4$ of the (a) feasible space and of the (b) second, and (c) third outer approximations with the associated lower- and upper-bounds. Variables a_1 and a_2 stand for the cross-section areas of the two elements and $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}$ denotes the corresponding weighted compliance of the two load cases (assuming the moments of inertia $I_i = 25/27a_i^3$, $i \in \{1, 2\}$).

recovered by Theorem 3.14 are global minima, Figs. 3.2d-3.2f.

Because all local minima are also global, all tested optimization algorithms converge to the optimal objective function value, see Table 3.1. Among these algorithms, OC and MMA exhibited the best performance in terms of computational time.

3.4.2 Irreducible positive optimality gap

Let us now modify the optimization problem described in the preceding section by fixing the volume bound to some $\overline{V} \in (0.816597322, 2.73603242)$. Whilst the boundary points of this open interval match the cases when three global optima occur, the interval interior removes $a_1 = a_2 = \overline{V}\sqrt{2}/4$ from the set of globally optimal solutions. In what follows, we set $\overline{V}$ to the center of the interval.

Solving this modified optimization problem with the moment-sum-of-squares hierarchy produces the sequence of lower- and upper-bounds shown in Fig. 3.3. Although the hierarchy exhibited a finite convergence based on the rank condition (3.8) with $s = 2$, the corresponding optimality gap remains strictly positive ($\varepsilon = 0.521$) and cannot be reduced in the subsequent relaxations. Clearly, all outer convex approximations must contain the convex combination of their limit points. Hence, if the limit points denote the global optima, also their convex combinations attain the globally optimal objective function value. Therefore, they are also optimal for the associated relaxation, but may lack feasibility to the original problem. Depending on the optimization algorithm and its settings, one can either reach a lower bound that is actually feasible for the original problem (as was the case in Section 3.4.1), i.e., a zero optimality gap, or a positive optimality gap that cannot be further reduced, which is the case here.

For this particular problem, all local optimization techniques, using their default starting points and settings, missed the global optima, see Table 3.2. In fact, they approached the feasible upper-bound that was provided by Theorem 3.14.

3.4.3 Frame structure with self-weight

For the previous examples, $\mathbf{f}(\mathbf{a}) = \mathbf{f}$ was constant, so that the optimum designs utilized the entire available volume $\overline{V}$. In these cases, the volume inequality constraint could have been

method	a_1	a_2	LB	UB	time [s]	$n_c \times m$	n
OC	0.628	0.628	-	2.161	0.003	-	2
MMA	0.628	0.628	-	2.161	0.004	-	2
`fmincon`	0.628	0.628	-	2.161	0.049	-	8
NSDP	0.628	0.628	-	2.161	0.200	2×4	4
PO$^{(2)}$, Lemma 3.15	0.628	0.628	0.936	2.161	0.026	1×15, 6×5, 2×4	69
PO$^{(3)}$, Lemma 3.15	0.628	0.628	1.640	2.161	0.062	2×35, 6×15, 2×20	209
PO$^{(3)}$, Eq. (3.8)	1.256	0.000	1.640	1.640	0.062	1×35, 6×15, 2×20	209
	0.000	1.256	1.640	1.640	0.062	1×35, 6×15, 2×20	209

Table 3.2: Different optimization methods applied to the second optimization problem. LB denotes lower bound, UB abbreviates feasible upper bounds, and PO stands for polynomial optimization. In addition, n_c stands for the number of semidefinite constraints of the size m and n is the number of variables. The entries a_i denote cross-sectional areas of the i-th element, or the areas constructed from the first-order moments in the case of PO.

changed to equality, and, therefore, one design variable eliminated. However, such a procedure cannot be applied when design-dependent loads are present.

To visualize this, let our third illustration be the single-load-case frame structure in Fig. 3.4a, composed of two frame elements with $E = 1$ with I-shaped cross-sections, Fig. 3.4b, parameterized by the thickness $t_{\mathrm{p},i}$. The overall volume is bounded from above by $\overline{V} = 1$. The self-weight applies in the vertical direction and is parameterized by the material density $\rho = 10$. For the considered cross-sections, we have $a_i = 18t_{\mathrm{p},i}^2$ and $I_i = 246t_{\mathrm{p},i}^4$. Hence, $c_{\mathrm{II}} = 41/54$ and $c_{\mathrm{III}} = 0$ in Eq. (3.12).

The feasible domain of this optimization problem, Fig. 3.4c, reveals three local optima, and one of them is the global solution. Computation of the optimum by the moment-sum-of-squares hierarchy required three relaxations, Figs. 3.4d–3.4f, which converged based on both the rank condition (3.8) with $s = 1$ and on Lemma 3.15 with $\varepsilon = -7 \times 10^{-8}$; the slightly negative value of ε is due to the numerical accuracy of the optimizer. Also notice that the upper-bounds based on Theorem 3.14 are of very high qualities, see Table 3.3.

Using local optimization techniques, only OC and MMA were able to arrive at the global

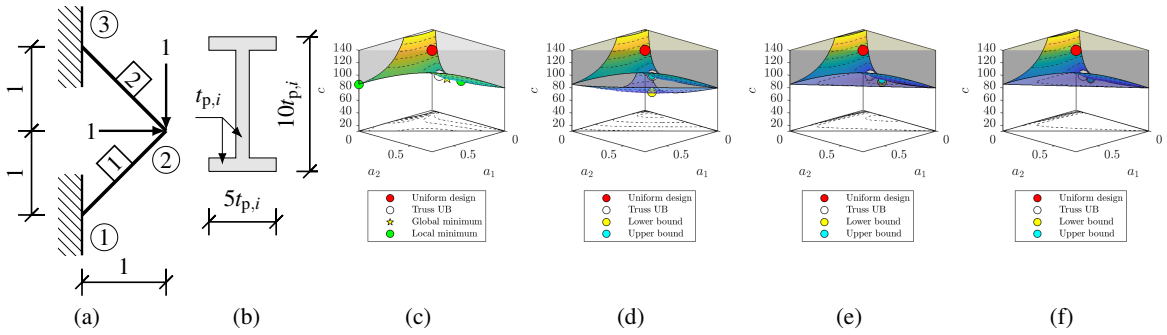


Figure 3.4: Frame structure with self-weight: (a) boundary conditions, (b) the cross-section parametrization, and the sub-level set $c \leq 140$ of the (c) feasible space and of the (d) first, (e) second, and (f) third outer approximations with the associated lower- and upper-bounds. Variables a_1 and a_2 stand for the cross-section areas of the two elements and c denotes the corresponding compliance (assuming the moments of inertia $I_i = 41/54a_i^2$, $i \in \{1, 2\}$), and $t_{\mathrm{p},i}$ stands for the flanges and web thickness.

method	a_1	a_2	LB	UB	time [s]	$n_c \times m$	n
OC	0.022	0.166	-	70.442	1.129	-	2
MMA	0.022	0.166	-	70.442	0.935	-	2
NSDP	0.707	0.000	-	85.846	1.448	1×4	3
PO$^{(1)}$, Lemma 3.15	0.050	0.119	48.246	74.171	0.006	2×4	9
PO$^{(2)}$, Lemma 3.15	0.034	0.220	68.328	71.594	0.015	1×10, 4×4, 16	34
PO$^{(3)}$, Lemma 3.15	0.022	0.166	70.442	70.442	0.058	1×20, 4×10, 40	83

Table 3.3: Different optimization methods applied to the optimization problem with self-weight. LB denotes lower bound, UB abbreviates feasible upper bounds, and PO stands for polynomial optimization. In addition, n_c stands for the number of semidefinite constraints of the size m and n is the number of variables. The entries a_i denote cross-sectional areas of the i-th element, or the areas constructed from the first-order moments in the case of PO.

optimum, see Table 3.3. Among other formulations, NSDP approached the worst local optimum and `fmincon` failed even to find a feasible solution.

3.4.4 Different element types on a cantilever beam

A certain generality of the developed approach is illustrated on a cantilever beam/plate design problem, Fig. 3.5. The dimensions of the cantilever are 5 in length and 1 in width, and the thicknesses of 5 finite elements are to be found in the optimization. The beam is made of a linear-elastic material with the Young modulus $E = 1$ and Poisson ratio $\nu = 0.25$. This structure is subjected to a tip distributed load of magnitude 1 induced under 30° angle with respect to the midline/midsurface. We optimize the frame/shell thicknesses t_i (of rectangular cross-sections) while satisfying $\overline{V} = 10$. The shear correction factor is set to 5/6 where appropriate.

In what follows, we compare the optimization results of the cantilever problem for three finite element types: Euler-Bernoulli and Timoshenko frame elements, and the quadrilateral Mixed Interpolation Tensorial Component (MITC4) shell element [60]. For both of the frame element types, we have $c_{\mathrm{II}} = 0$ and $c_{\mathrm{III}} = I_i(a)/a_i^3 = 1/12$ in Eq (3.12), whereas $c_{\mathrm{II}} = 0$ and $c_{\mathrm{III}} = 1$ for the MITC4 element.

The moment-sum-of-squares hierarchy required three steps (degree-four relaxation) to converge in all three cases, Fig. 3.6, and approached very similar optimal thicknesses, Table 3.4. As expected, the lowest compliance is provided by the Euler-Bernoulli frame elements, which neglect the shear effects. We account for these effects in the Timoshenko frame elements,

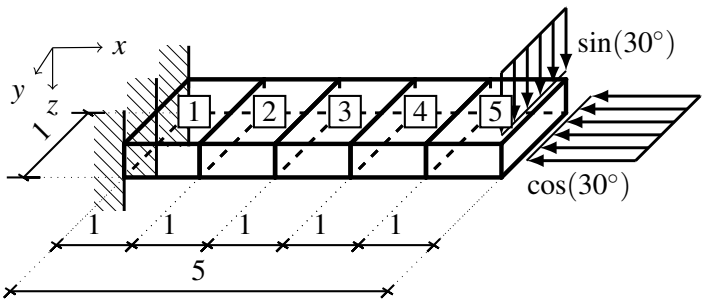


Figure 3.5: Boundary conditions of the cantilever beam design problem.

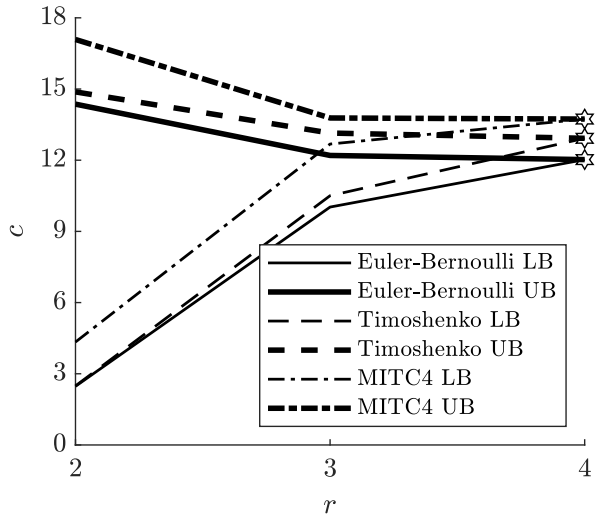


Figure 3.6: Convergence of the moment-sum-of-squares hierarchy for the cantilever problem with three finite element types. Variable c denotes compliance, r stands for the relation degree, and LB with UB abbreviate lower- and upper-bound.

increasing so the value of optimal compliance. Another generalization occurs when using the MITC4 shell elements, which not only consider the effects of shear, but further incorporate effects induced by bending about z axis, recall Fig. 3.5. Therefore, the optimal compliance associated with the MITC4 elements is the highest. Clearly, different finite elements lead to different designs, and their influence on optimal minimum-energy designs can be rigorously studied by the proposed approach.

	Euler-Bernoulli	Timoshenko	MITC4
a_1^*	2.775	2.724	2.754
a_2^*	2.454	2.414	2.462
a_3^*	2.086	2.060	2.091
a_4^*	1.639	1.643	1.651
a_5^*	1.047	1.159	1.041
c^*	12.025	12.923	13.734
time [s]	55.945	53.907	806.538
Lemma 3.15, ε	-3×10^{-9}	-6×10^{-9}	-6×10^{-9}
Eq. (3.8), s	1	1	1
n	3002	3002	3002
$n_c \times m$	1×210, 7×84, 1×448	1×210, 7×84, 1×448	1×210, 7×84, 1×1428

Table 3.4: Globally optimal thicknesses $a_1^*, \ldots, a_5^*$ and compliances c^* for the cantilever problem for three element types: Euler-Bernoulli and Timoshenko frame elements, and the MITC4 shell element. Variables ε and s denote the optimality gap in Lemma 3.15 and the rank of the moment matrices according to (3.8), respectively. In addition, n_c stands for the number of semidefinite constraints of the size m and n is the number of variables.

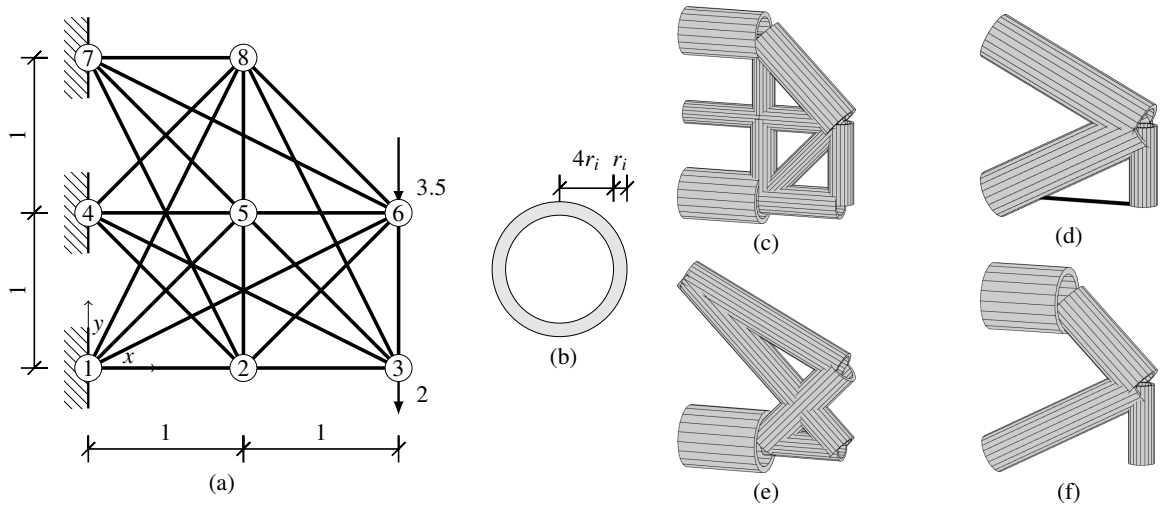


Figure 3.7: (a) Ground structure of the 22-elements frame optimization problem, (b) cross-section parameterized by r_i, and optimized designs of compliances: (c) $c = 3276.3$ obtained by $\mathrm{PO}^{(1)}$, (d) $c^* = 1668.6$ resulting from $\mathrm{PO}^{(2)}$ and OC, (e) $c = 1697.7$ reached by MMA and `fmincon`, and (f) $c = 1741.1$ optimized by NSDP.

3.4.5 22-elements frame structure

Our final example investigates topology optimization of a 22-element frame structure shown in Fig. 3.7a. Two loads are applied at nodes 3 and 6 in a single load case. In addition, we set $E = 1$ and $\overline{V} = 0.5$. All the structural elements possess a thin-walled circular cross-section with the radius $5r_i$ and the wall thickness r_i, Fig. 3.7b. Hence, $a_i = 9\pi r_i^2$ and $I_i = 46.125\pi r_i^4$, so that $c_{\mathrm{II}} = 46.125/(81\pi)$ and $c_{\mathrm{III}} = 0$.

The moment-sum-of-squares hierarchy requires two relaxations to achieve a guaranteed global optimum, both based on Lemma 3.15 with $\varepsilon = 2 \times 10^{-5}$ and on Eq. (3.8) with $s = 1$. However, even the second relaxation is fairly computationally expensive (see Table 3.5), prohibiting solution of higher relaxations of similarly-sized problems on standard hardware.

Evaluation of the local optimization algorithms revealed that only OC converged to the global optimum ($c^* = 1668.6$ shown in Fig. 3.7d). The remaining optimization approaches reached local optima of comparable performance but considerably different topologies: MMA and `fmincon` converged to the design shown in Fig. 3.7e with $c = 1697.7$, and NSDP reached

method	LB	UB	time [s]	$n_c \times m$	n
OC	-	1668.585	2.454	-	22
MMA	-	1697.749	13.816	-	22
`fmincon`	-	1697.665	0.650	-	37
NSDP	-	1741.062	4.406	1×16	23
$\mathrm{PO}^{(1)}$, Lemma 3.15, Eq. (3.8)	1062.105	3276.294	0.103	1×24, 1×16	299
$\mathrm{PO}^{(2)}$, Lemma 3.15, Eq. (3.8)	1668.584	1668.584	1492.842	1×300, 24×24, 1×384	17549

Table 3.5: Different optimization methods applied to the 22-frame structure design problem. LB denotes lower bounds, UB abbreviates feasible upper bounds, and PO stands for polynomial optimization. In addition, n_c stands for the number of semidefinite constraints of the size m and n is the number of variables.

the design in Fig. 3.7f with $c = 1741.1$.

3.5 Conclusions

Our contribution has addressed the fundamental question in the structural design: how to find globally-optimal minimum-compliant bending-resistant structures in discrete topology optimization with continuous design variables. For the cases of frame and shell structures, multiple loading conditions and design-dependent loads, we have formulated this optimization problem as a (non-linear) semidefinite program constrained by a polynomial matrix inequality. The feasible set of this optimization problem forms a semialgebraic set; hence, powerful results on polynomial optimization—the moment-sum-of-squares hierarchy—facilitate computation of the global solutions.

This hierarchy generates a sequence of tightening outer convex approximations in the space of moments of the probability measures, so that the first-order moments converge monotonically to the convex hull of the original problem. Therefore, a non-decreasing sequence of lower bounds is established. Using the first-order moments only, we have shown that a sequence of feasible upper bounds can be obtained by a simple correction. Consequently, because lower and upper bounds are available in each relaxation, the upper-bound design quality can be assessed, establishing a sufficient condition of global ε-optimality. This condition is very simple to check and complements the traditional rank-based certificate of global optimality, e.g., [85].

Our condition fails to be necessary because the first-order moments are not unique when considered optimization problem possesses multiple global optima, potentially leaving a strictly positive optimality gap. For the case of the unique global optimum, we have shown that the hierarchy eventually attains a zero optimality gap as the relaxation number approaches infinity. We note here that the possibility of the global minima multiplicity can almost be avoided in practice when the symmetry of structure and boundary conditions are exploited.

These theoretical results have been illustrated on five problems, which indicate the merits and weaknesses of the presented strategy. First, all of our test problems exhibited a rapid convergence of the hierarchy, allowing for extraction of all global solutions based on the flat extension theorem of Curto and Fialkow [45]. However, the computational complexity is currently fairly high, also when compared to investigated local optimization techniques, leaving the ability to compute proven global optima for small-scale optimization problems only. Yet, even for middle-scale problems, the hierarchy still provides a sequence of upper bounds of reasonable qualities, and, especially, the certificate of their ε-optimality.

In the future, we plan to extend these results in multiple directions. First, we believe that Theorem 3.17 can be strengthened to certify a zero optimality gap when convergence of the hierarchy in a finite number of steps occurs. Second, the computational demands could be decreased by exploiting the structural sparsity via clique-based chordal decomposition in the spirit of [101, 107]. Another research directions may explore eigenvalue constraints and optimization [3, 191], the minimum-weight setting, or, eventually, investigate performance of the hierarchy for different topology optimization formulations.

3.A Relation to truss topology optimization

The problem formulation (3.21) has already been known in the context of truss topology optimization [197], for which the constraints (3.21b) reduce to linear matrix inequalities (LMI). Consequently, the feasible set is convex, allowing for an efficient solution of (3.21) by interior point methods, for example.

A natural question then arises: What happens when the rotational degrees of freedom are neglected, solving truss topology optimization problem instead of the frame one? To this goal, however, we must first satisfy the rather restrictive assumption that the truss ground structure is capable of carrying the loads $\mathbf{f}_j(\mathbf{a})$, i.e.,

$$\mathbf{f}_j(\mathbf{a}) \in \operatorname{Im}\left(\mathbf{K}_{\mathrm{t},j}(\mathbf{a})\right), \forall j \in \{1,\ldots,n_{\mathrm{lc}}\}, \tag{3.43}$$

where $\mathbf{K}_{\mathrm{t},j}(\mathbf{a}) = \mathbf{K}_{j,0} + \sum_{i=1}^{n_{\mathrm{e}}} \mathbf{K}_{j,i}^{(1)} a_i$, and that with all empty rows and columns removed $\mathbf{K}_{\mathrm{t},j}(\mathbf{a})$ is positive definite for all positive $\mathbf{a}$. From the mechanical point of view, we require that no moment loads are imposed, a straight bar does not have to carry transverse loads, and the ground structure is well supported.

Suppose now that $\mathbf{a}_{\mathrm{t}}^*$ are optimal cross-sections obtained from a solution to (3.21) with the terms $\mathbf{K}_{j,i}^{(2)}$ and $\mathbf{K}_{j,i}^{(3)}$ neglected, and $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{t}}^*$ is the associated optimal objective function value, which can be computed from $\mathbf{a}_{\mathrm{t}}^*$ as

$$\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{t}}^* = \sum_{j=1}^{n_{\mathrm{lc}}} \left(\omega_j \left[\mathbf{f}_j(\mathbf{a}_{\mathrm{t}}^*)\right]^{\mathrm{T}} \left[\mathbf{K}_{\mathrm{t},j}(\mathbf{a}_{\mathrm{t}}^*)\right]^{\dagger} \mathbf{f}_j(\mathbf{a}_{\mathrm{t}}^*)\right). \tag{3.44}$$

When the optimal cross-sections of a truss structure, $\mathbf{a}_{\mathrm{t}}^*$, are reused in a frame structure, the resulting objective function value changes to

$$\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{f}} = \sum_{j=1}^{n_{\mathrm{lc}}} \left(\omega_j \left[\mathbf{f}_j(\mathbf{a}_{\mathrm{t}}^*)\right]^{\mathrm{T}} \left[\mathbf{K}_{\mathrm{t},j}(\mathbf{a}_{\mathrm{t}}^*) + \mathbf{K}_{\mathrm{b},j}(\mathbf{a}_{\mathrm{t}}^*)\right]^{\dagger} \mathbf{f}_j(\mathbf{a}_{\mathrm{t}}^*)\right) \tag{3.45}$$

with $\mathbf{K}_{\mathrm{b},j}(\mathbf{a}) = \sum_{i=1}^{n_{\mathrm{e}}} \left(c_{\mathrm{II}}\mathbf{K}_{j,i}^{(2)} a_{\mathrm{t},i}^2 + c_{\mathrm{III}}\mathbf{K}_{j,i}^{(3)} a_{\mathrm{t},i}^3\right)$.

To state a relation between $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{t}}^*$ and $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{f}}$ we recall a useful lemma:

Lemma 3.19. *[115] Let $\mathbf{A} \in \mathbb{S}^n$ and $\mathbf{B} \in \mathbb{R}^{n\times q}$. Then,*

$$\left(\mathbf{A} + \mathbf{B}\mathbf{B}^{\mathrm{T}}\right)^{\dagger} = \mathbf{A}^{\dagger} - \mathbf{A}^{\dagger}\mathbf{B}\left(\mathbf{I} + \mathbf{B}^{\mathrm{T}}\mathbf{A}^{\dagger}\mathbf{B}\right)^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{A}^{\dagger} + \left(\mathbf{B}_{\perp}^{\dagger}\right)^{\mathrm{T}}\mathbf{B}_{\perp}^{\dagger} \tag{3.46}$$

with $\mathbf{B}_{\perp} = \left(\mathbf{I} - \mathbf{A}\mathbf{A}^{\dagger}\right)\mathbf{B}$.

Using this lemma, we prove that $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{t}}$ provides an upper bound for $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{f}}$.

Lemma 3.20. *Suppose that $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}^*$ is the optimal objective function value of the frame structure design problem* (3.21) *and* (3.43) *holds. Then, $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}^* \leq \boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{f}} \leq \boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{t}}^*$.*

Proof. Because of $\mathbf{f}_j(\mathbf{a}_{\mathrm{t}}^*) \in \operatorname{Im}\left(\mathbf{K}_{\mathrm{t},j}(\mathbf{a}_{\mathrm{t}}^*)\right)$, we clearly have $\mathbf{f}_j(\mathbf{a}_{\mathrm{t}}^*) \in \operatorname{Im}\left(\mathbf{K}_{\mathrm{t},j}(\mathbf{a}_{\mathrm{t}}^*) + \mathbf{K}_{\mathrm{b},j}(\mathbf{a}_{\mathrm{t}}^*)\right)$. Therefore, $\mathbf{a}_{\mathrm{t}}^*$ is a feasible solution to the frame structure design problem (3.21) and the associated objective function is bounded from below by the global optimum $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}^*$. Hence, $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}^* \leq \boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{f}}$.

For the other inequality, we express (3.45) using Lemma 3.19. To this goal, let $\mathbf{K}_{\mathrm{b},j}(\mathbf{a}_{\mathrm{t}}^*) = \mathbf{B}_j\mathbf{B}_j^{\mathrm{T}}$, where $\mathbf{B}_j$ is a real matrix because $\mathbf{K}_{\mathrm{b},j}(\mathbf{a}_{\mathrm{t}}^*) \succeq 0$ by definition. Then, (3.45) can be written as

$$\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{f}} = \boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{t}}^* - \boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{a}} + \boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{b}}, \tag{3.47}$$

where

$$c_{\mathrm{a},j} = [\mathbf{f}_j(\mathbf{a}_{\mathrm{t}}^*)]^{\mathrm{T}}\,\mathbf{A}_j^{\dagger}\mathbf{B}_j\left(\mathbf{I} + \mathbf{B}_j^{\mathrm{T}}\mathbf{A}_j^{\dagger}\mathbf{B}_j\right)^{-1}\mathbf{B}_j^{\mathrm{T}}\mathbf{A}_j^{\dagger}\mathbf{f}_j(\mathbf{a}_{\mathrm{t}}^*), \tag{3.48a}$$

$$c_{\mathrm{b},j} = [\mathbf{f}_j(\mathbf{a}_{\mathrm{t}}^*)]^{\mathrm{T}}\left(\mathbf{B}_{\perp,j}^{\dagger}\right)^{\mathrm{T}}\mathbf{B}_{\perp,j}^{\dagger}\mathbf{f}_j(\mathbf{a}_{\mathrm{t}}^*), \tag{3.48b}$$

with $\mathbf{A}_j = \mathbf{K}_{\mathrm{t},j}(\mathbf{a}_{\mathrm{t}}^*)$. Clearly, Eq. (3.48a) is non-negative. For (3.48b), $\left(\mathbf{B}_{\perp}^{\dagger}\right)^{\mathrm{T}}\mathbf{B}_{\perp}^{\dagger} \in \mathrm{Ker}(\mathbf{A}_j)$, so that $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{b}}$ vanishes. Hence, $\boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{f}} = \boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{t}}^* - \boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{a}} \leq \boldsymbol{\omega}^{\mathrm{T}}\mathbf{c}_{\mathrm{t}}^*$. □

Thus, when (3.43) holds true, the truss topology optimization produces an upper bound to the optimal objective of the frame structure topology optimization problem.

3.B Numerical performance of bound constraints

This section illustrates the effect of using different types of bound constraints, (3.24a) or (3.24b), on performance of the moment-sum-of-squares hierarchy.

We start by re-evaluating the problem in Section 3.4.3. Our numerical experiments in Table 3.6 reveal that for this specific problem the quadratic constraints (3.24b) are substantially tighter in terms of generated lower bounds, they require smaller relaxation degree to converge, and are computationally more efficient. Also notice that fewer constraints are needed when using (3.24b).

For the problem in Section 3.4.4 discretized by Euler-Bernoulli frame elements, we observe that the difference in performance becomes less noticeable, see Table 3.7: the quadratic constraints lead to a better lower bound in the third relaxation only, but are still computationally more efficient because fewer constraints are needed.

formulation	a_1	a_2	LB	time [s]	$n_{\mathrm{c}} \times m$	n
PO$^{(1)}$ (3.24a)	0.008	0.008	0.000	0.008	2×4	9
PO$^{(1)}$ (3.24b)	0.050	0.119	48.246	0.006	2×4	9
PO$^{(2)}$ (3.24a)	0.052	0.170	49.805	0.058	1×10, 7×4, 1×16	34
PO$^{(2)}$ (3.24b)	0.034	0.220	68.328	0.015	1×10, 4×4, 1×16	34
PO$^{(3)}$ (3.24a)	0.026	0.201	69.755	0.173	1×20, 7×10, 1×40	83
PO$^{(3)}$ (3.24b)	0.022	0.166	**70.442**	0.058	1×20, 4×10, 1×40	83
PO$^{(4)}$ (3.24a)	0.022	0.166	**70.442**	0.291	1×35, 7×20,1×80	164

Table 3.6: Performance of the hierarchy on the problem with self-weight using different bound constraints. LB denotes lower bound, and PO stands for polynomial optimization. In addition, n_{c} stands for the number of semidefinite constraints of the size m and n is the number of variables. The entries a_i denote the cross-section areas constructed from the first-order moments.

formulation	a_1	a_2	a_3	a_4	a_5	LB	time [s]	$n_c \times m$	n
$PO^{(2)}$ (3.24a)	2.346	2.129	1.950	1.822	1.754	2.471	0.072	1×28, 13×7, 1×16	209
$PO^{(2)}$ (3.24b)	2.236	2.129	1.950	1.822	1.754	2.471	0.074	1×28, 7×7, 1×16	209
$PO^{(3)}$ (3.24a)	2.947	2.524	2.043	1.520	0.966	10.018	2.091	1×84, 13×28, 1×112	923
$PO^{(3)}$ (3.24b)	2.946	2.523	2.043	1.522	0.967	10.025	1.854	1×84, 7×28, 1×112	923
$PO^{(4)}$ (3.24a)	2.775	2.454	2.086	1.640	1.047	**12.025**	64.615	1×210, 13×84, 1×448	3002
$PO^{(4)}$ (3.24b)	2.775	2.454	2.086	1.640	1.047	**12.025**	54.634	1×210, 7×84, 1×448	3002

Table 3.7: Performance of the hierarchy on the cantilever problem using different bound constraints. LB denotes lower bound, and PO stands for polynomial optimization. In addition, n_c stands for the number of semidefinite constraints of the size m and n is the number of variables. The entries a_i denote the cross-section areas constructed from the first-order moments.

Chapter 4

Modular-topology optimization with Wang tilings: An application to truss structures

Abstract: *Modularity is appealing for solving many problems in optimization. It brings the benefits of manufacturability and reconfigurability to structural optimization, and enables a trade-off between the computational performance of a Periodic Unit Cell (PUC) and the efficacy of non-uniform designs in multi-scale material optimization. Here, we introduce a novel strategy for concurrent minimum-compliance design of truss modules topologies and their macroscopic assembly encoded using Wang tiling, a formalism providing independent control over the number of modules and their interfaces. We tackle the emerging bilevel optimization problem with a combination of meta-heuristics and mathematical programming. At the upper level, we employ a genetic algorithm to optimize module assemblies. For each assembly, we obtain optimal module topologies as a solution to a convex second-order conic program that exploits the underlying modularity, incorporating stress constraints, multiple load cases, and reuse of module(s) for various structures. Merits of the proposed strategy are illustrated with three representative examples, clearly demonstrating that the best designs obtained by our method exhibited decreased compliance: by* 56% *up to* 69% *compared to the PUC designs.*

Reproduced from:

[192] M. Tyburec, J. Zeman, M. Doškář, M. Kružík, and M. Lepš, Modular-topology optimization with Wang tilings: an application to truss structures, *Structural and Multidisciplinary Optimization*, 63(3):1099–1117, 2020, DOI: 10.1007/s00158-020-02744-8

4.1 Introduction

Modular structures, composed of repeated building blocks (modules), offer multiple appealing advantages over non-modular designs. These include more economical mass fabrication, increased production productivity [188], and better quality control [137]. In addition, modules facilitate structural reconfigurability, conversion among designs with considerably different structural responses [141]. Finally, the design of modular structures enables structural efficiency to be balanced with design complexity [189], which often arises in optimal structures [111].

Our approach to designing modular structures follows recent successful applications of Wang tiles in compression and reconstruction of heterogeneous microstructures [143, 56, 12, 58], where modularity suppresses artificial periodicity artifacts inherent to the periodic-unit-cell approach, e.g., [225]. Here, we focus on the reverse direction: designing modular structures or materials composed of a compressed set of modular LEGO®-like building blocks. In this endeavor, the concept of Wang tiles provides us with a convenient mechanism for describing and controlling module types as well as their interface types. Our approach is explained in the simplest setting: the topology optimization of truss structures.

Below, we review recent developments in the modular design of truss structures (Section 4.1.1), and modular microstructures, (Section 4.1.2). Finally, we discuss the benefits of our approach in Section 4.1.3.

4.1.1 Design of modular trusses

The optimal design of modular trusses appears to be a new, to a large extent unexplored, branch of structural optimization. In one of the pioneering works in this area of research, Tugilimana *et al.* [188] developed a method to optimize the topology of a single module as well as the module's spatially-varying rotations within the design domain. With this method, the optimization part relies on a plastic design formulation [53] and thus provides lower-bound designs only as the kinematic compatibility is not automatically satisfied. The need for an elastic design formulation resulted in a follow-up work [187], which proposed a non-convex formulation allowing for multiple load cases, stress constraints, multiple module types, as well as the module reusability among structures. However, this formulation still requires a manual definition of the module spatial distribution within the design domains. Therefore, Tugilimana *et al.* [189] introduced a two-level approach to optimize the topology of multiple modules as well as their spatial placement within a structural design domain. While the lower-level formulation they proposed additionally extends itself to self-weight and local and global stability constraints, the upper-level simulated annealing with an adaptive neighborhood ensures dynamic grouping of modules.

Another approach, that of Zawidzki and Jankowski [223], proposes a bilevel optimization method to optimize the Truss-Z system [224], a modular pedestrian network. In the upper-level of this approach, the NSGA-II algorithm optimizes the module outer shape to provide geometrically versatile structures that can construct connected paths between pairs of access points. Lower-level optimization then employs simulated annealing with an adaptive neighborhood and minimizes the module weight in the sizing optimization of circular thin-walled sections with constraints on the von-Mises stresses and the Euler buckling ratio.

Limiting the number of unique cross-section areas of truss structures to improve con-

structability resulted in the use of dynamic grouping in topology optimization. In contrast to the former methods, where modules comprise multiple truss elements, this approach groups individual cross-sections into sets whose number and cardinalities result from the optimization. For example, Shea *et al.* [174] adopted the shape annealing approach with a grouping criterion based on the optimized non-grouped cross-section areas, and Toğan and Daloğlu [186] employed genetic algorithms with grouping criteria based on the internal forces and on the slenderness ratios. Finally, Lemonge *et al.* [123] extended a genetic algorithm to multiple cardinality constraints to solve frame structure sizing optimization problems.

4.1.2 Design of modular microstructures

Distinguished modular and structural scales in the design of modular trusses evoke the standard multi-scale topology optimization for the design of (meta-)material microstructures. In such settings, theoretically optimal designs occur when microstructures vary pointwise in macro-scale design domains [163], i.e., with each macro-point associated with an independent module type.

In a single-scale setting, a large resolution is usually required to attain a sufficient level of details at the microstructure scale, with a price of considerable computational costs [1]. Limiting the computational demands has thus been the aim of various investigations. One group of such methods extends the inverse homogenization approach of Sigmund [175, 176], developed originally for uniform microstructures, by designing an independent microstructure for each macro point [163, 40]. Unfortunately, resulting designs may lack manufacturability due to discontinuous material distribution over microstructural cell interfaces. This issue has been resolved recently, e.g., in [71]. Alternatively, projection-based methods optimize parametric unit cells while maintaining their connectedness by a diffeomorphic projection of optimally-oriented parametric microstructures at a given length scale, reaching the limits of structural efficiency [152, 75, 7, 72].

Another way of accelerating solutions reduces the design space by limiting the number of unique microstructures. The periodic unit cell (PUC) approach—repeating a single, possibly graded, microstructural cell throughout the entire design domain—was initially employed [128, 183, 127]; however, the enforced periodicity may significantly compromise the quality of optimized designs. In general, structural performance improves when the number of microstructure types increases [116], i.e., when the macro-structure becomes modular. The emerging modularity thus paves the way for balancing computational demands with structural efficiency. In this context, Sivapuram *et al.* [179] extended the multi-scale optimization approach by limiting the number of microstructures, but their method requires a predefined spatial placement of microstructural cells and still lacks material continuity across their boundaries. More general methods, published by Li *et al.* [126] and Zhang *et al.* [227], introduced concurrent approaches to simultaneously optimize spatial placement and the topology of a finite set of modules while ensuring the material continuity with predefined kinematical connectors.

4.1.3 Aims and novelty

As seen from the state-of-the-art review, the optimum design of modular structures is a rapidly-evolving line of research for the structural optimization community. In this contribution, we consider this inherently two-level design problem in its original form; similarly to earlier

studies [188, 187, 189], we design optimized topologies of modules (lower level) and the assembly plan (upper level) concurrently. In contrast to the earlier contributions, however, we develop a convex lower-level formulation for the topology optimization of truss modules, while still allowing for stress constraints, module reusability, and multiple load cases.

Thanks to convexity, we can reach true global optima for coarse discretizations and assure that the algorithm avoids poor local optima for module topologies. We find this convex subproblem very important in this incipient phase of research because it provides a rigorous answer regarding what is achievable. Unfortunately, this convexity does not translate into continuum topology optimization, which explains our choice for discretization with truss elements.

Second, we adopt the formalism of corner Wang tilings to describe the assembly plan of modules. This generalization [143, 56, 54, 57, 58] of the PUC allows us to design compressed yet non-periodic assemblies, a novel class of connectable and reusable (micro-)structures and materials. Contrary to [126, 227], our approach generates mechanically compatible structures fully automatically and avoids prescribing fixed cell interfaces.

The rest of the paper is structured as follows. In Section 4.2, we introduce the formalism of corner Wang tilings and recall a convex elastic-design formulation for topology optimization of trusses. Subsequently, we extend this formulation with modularity, stress constraints, multiple load cases, and module reusability. To handle the discreteness of the modular assembly plan, a genetic algorithm is applied as the solver in the upper-level optimization in Section 4.3. Finally, Section 4.4 illustrates our method with three examples, assessing the scalability of the approach and multiple constraint types, and leads us to the conclusion that the proposed methodology is fairly efficient.

4.2 Background

4.2.1 Wang tilings

One of the goals of this article is to explore the merits of Wang tilings [201] for the optimal design of modular structures. Wang tiles—unit squares with colored edges and fixed orientation—constitute a formalism introduced by Wang [201] to visualize the $\forall\exists\forall$[1] decidability problem of predicate calculus using an equivalent domino problem. Wang considered an infinite number of copies of an arbitrary set of Wang tiles and investigated whether there exists a simply-connected tiling of the infinite plane such that the adjoining edges of neighboring tiles share the same color—the so-called valid tilings. Contrary to Wang's conjecture, Berger [26] built a tileset that covers the infinite plane and that yet does not allow any periodic pattern to emerge, a property proved by a reduction from the Turing machine halting problem [190].

The two properties that made Wang tiles appealing in multi-disciplinary research are notably Turing completeness and the ability to form aperiodic patterns. While computations by self-assembly of DNA [209, 208] and automated theorem proving [201] exploit the Turing completeness, non-periodic patterns apply in computer graphics to construct compressed yet naturally-looking textures [41]. The latter research also provided the motivation to employ

[1] The $\forall\exists\forall$ is a decision problem investigating whether there is a general algorithm that decides emptiness of a satisfiable set for all logical formulas of the form "for all x there is a y such that for all $z\ldots$" which is followed by a logical combination of predicates without quantifiers [203].

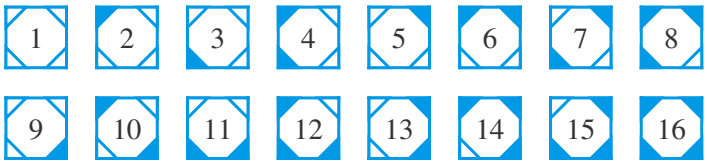


Figure 4.1: The complete set of corner Wang tiles over two colors.

Wang tiles in compression and reconstruction of microstructures, generalizing the concept of the periodic unit cell [143, 56, 54, 57, 58].

Although traditional Wang tiles maintain information continuity across the edges, some discontinuity artifacts might appear in their corners. To solve this so-called corner problem [41] and avoid these periodically repeating artifacts [58], tiles with connectivity information stored in colored corners were proposed [119]. These corner tiles form a subset of Wang tiles, as each combination of vertex color codes denotes a unique edge type. Note that a reverse procedure is not generally applicable [58].

The corner Wang tiles proved to be preferable over the traditional Wang tiles, allowing for simpler generation of valid tilings, reduced memory requirements, and easier generalization to multiple dimensions [119], while preserving the possibility of building aperiodic tilings [121]. These findings inspired us to employ corner tiles in this paper. In particular, we consider here the complete set of planar corner Wang tiles over two colors, containing one corner tile for each possible combination of color codes as depicted in Fig. 4.1.

Using corner tiles, valid assemblies or tilings must satisfy identical colorings of shared vertices over all adjacent tiles, compare Fig. 4.2b and 4.2c. Marking each color code with an integer value [119], any valid assembly determines the color-code connectivity matrix $\mathbf{C}$ uniquely. Note that in the case of only two vertex colors, the connectivity matrix becomes Boolean, Fig. 4.2a. Conversely, for all complete sets of corner tiles over a limited set of colors, any connectivity matrix containing integer values corresponding to the vertex codes of the set automatically defines a valid rectangular tiling. Notice that an extension to non-rectangular tiling with holes is straightforward, using a flattened one-dimensional array. In addition, non-rectangular tiles can be adopted by prescribing various shapes to different edge types, see, e.g., the mapping from regular Wang tilings to aperiodic Penrose tiling containing kites and darts [79, Fig. 11.1.5].

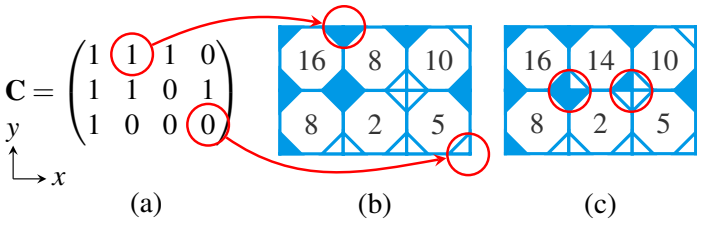


Figure 4.2: Illustration of (a) a connectivity matrix and (b) its correspondence to a valid tiling, (c) an example of an invalid tiling.

4.2.2 Optimal truss design

Trusses are structures consisting of nodes and straight bars which transmit axial forces only. While the optimal topology of the least-compliant trusses under a single load case aligns structural stiffness with the principal strains [136], their trajectories are not straight in general and hence an optimal design can contain an infinite number of bars. To overcome this undesirable property, the continuum design domain is usually discretized into the so-called ground-structure [53], constituting a finite-dimensional design space formed by the sets of $n_\mathrm{n} \in \mathbb{N}$ fixed nodes and $n_\mathrm{b} \in \mathbb{N}$ potential bars. Design variables of truss topology optimization then involve cross-section areas $\mathbf{a} \in \mathbb{R}^{n_\mathrm{b}}_{\geq 0}$ [22], possibly attaining zero values, so that the structurally inefficient truss elements vanish. To this goal, two branches of truss topology optimization commonly apply.

The traditional *plastic design* [53] poses the problem in terms of the member axial forces only, and searches for the minimum-weight topology under the static equilibrium constraint and bounds on allowed stresses. Despite neglecting the kinematic compatibility, convergence to statically determinate[2] optimal designs occurs [185] when a single load case is considered [165] and cross-section areas are not constrained, so that the compatibility conditions hold. Unfortunately, modularity violates the latter assumption by implicitly requiring equality of cross-sections within specified sets of elements. This can be seen, for example, by optimizing any statically indeterminate ground structure and requiring all bars to share the same cross-section area. Consequently, modular designs obtained by the plastic formulation may and usually are statically indeterminate, violate the compatibility conditions, and in turn provide a lower bound for the objective function value only.

Therefore, the compatibility conditions must be considered, impelling us to use an *elastic design* formulation instead. Here, we search for the least-compliant design (4.1a), subjected to the bound on the available volume of material (4.1c) and the equilibrium equation (4.1b), i.e.,

$$\begin{aligned}
\min_{\mathbf{a},\mathbf{u}} \quad & \frac{1}{2}\mathbf{f}^\mathrm{T}\mathbf{u} && \text{(4.1a)}\\
\text{s.t.} \quad & \mathbf{K}(\mathbf{a})\mathbf{u} = \mathbf{f}, && \text{(4.1b)}\\
& \boldsymbol{\ell}^\mathrm{T}\mathbf{a} \leq \overline{V}, && \text{(4.1c)}\\
& \mathbf{a} \geq \mathbf{0}, && \text{(4.1d)}
\end{aligned}$$

where, $\mathbf{f} \in \mathbb{R}^{n_\mathrm{dof}}$ denotes the nodal forces column vector, $n_\mathrm{dof} \in \mathbb{N}$ refers to the number of degrees of freedom, $\mathbf{u} \in \mathbb{R}^{n_\mathrm{dof}}$ stands for the displacements column vector, $\mathbf{K}(\mathbf{a}) \in \mathbb{R}^{n_\mathrm{dof} \times n_\mathrm{dof}}$ is the structural stiffness matrix, an affine function of the cross-section areas, $\boldsymbol{\ell} \in \mathbb{R}^{n_\mathrm{b}}$ stands for the column vector of the bars lengths, and $\overline{V} \in \mathbb{R}_{>0}$ denotes an upper-bound on the total structural volume. Finally, the objective of the optimization $c = \frac{1}{2}\mathbf{f}^\mathrm{T}\mathbf{u}$, $c \in \mathbb{R}_{\geq 0}$, (4.1a) denotes compliance, i.e., half of the work done by the external loads.

While the problem (4.1) is straightforward to formulate, it lacks convexity due to the bilinear equilibrium equation (4.1b) with a possibly singular stiffness matrix, and is hard to solve (to global optimality) even for small-scale problems [108]. However, based on [130] and Section 3.4.3 in [17], we can write (its dual) convex second-order cone programming (SOCP)

[2] Statically determinate designs uniquely determine the axial forces only based on the static equilibrium [165].

reformulation:

$$\min_{\mathbf{a},\mathbf{w},\mathbf{s}} \quad \mathbf{1}^{\mathrm{T}}\mathbf{w} \tag{4.2a}$$

$$\text{s.t.} \quad \boldsymbol{\ell}^{\mathrm{T}}\mathbf{a} \leq \overline{V}, \tag{4.2b}$$

$$\mathbf{A}\mathbf{s} = \mathbf{f}, \tag{4.2c}$$

$$\left\| \begin{pmatrix} w_i - a_i \\ \sqrt{\frac{2\ell_i}{E_i}}\, s_i \end{pmatrix} \right\|_2 \leq w_i + a_i, \quad \forall i \in \{1, \ldots, n_{\mathrm{b}}\}, \tag{4.2d}$$

$$\mathbf{a} \geq \mathbf{0}, \tag{4.2e}$$

which is efficiently solvable to global optimality by interior-point methods [11]. In Eq. (4.2), the symbol $\mathbf{s} \in \mathbb{R}^{n_{\mathrm{b}}}$ stands for the axial forces column vector, $E_i \in \mathbb{R}_{>0}$ denotes the modulus of elasticity, $w_i \in \mathbb{R}_{\geq 0}$ constitutes the complementary strain energy, and ℓ_i is the length of the i-th bar, respectively. Further, $\mathbf{A} \in \mathbb{R}^{n_{\mathrm{dof}} \times n_{\mathrm{b}}}$ stands for the static matrix relating axial, $\mathbf{s}$, and nodal, $\mathbf{f}$, forces. At the optimum, the objective function (4.2a) attains the value of the complementary strain energy, equal to the compliance, as $\mathbf{w}$ provides a to-be-minimized upper bounds on the complementary strain energies of individual bars in the second-order conic constraints (4.2d).

4.3 Methodology

Modularity constitutes a partitioning of a complex structure into several simpler repeated units—modules. Here, we assume that a fixed (rectangular) structural design domain of the size $n_{\mathrm{t},y} \times n_{\mathrm{t},x}$ consists of square truss modules with a fixed orientation, see Fig. 4.3a. Without loss of generality, the number of (employed) module types n_{t} is at most $n_{\mathrm{t},y}n_{\mathrm{t},x}$. With $n_{\mathrm{t}} = n_{\mathrm{t},y}n_{\mathrm{t},x}$, the problem is equivalent to the non-modular design, while $n_{\mathrm{t}} = 1$ implies a single-module periodic design.

When $n_t \ll n_{\mathrm{t},y}n_{\mathrm{t},x}$, it may happen that each module type neighbors with all the remaining module types, implying n_t^2 ways of possible module interconnections. In this setting, it is therefore not surprising that a single solid/high-stiffness interface is obtained during optimization [71] and this interface then propagates periodically through the macro design domain. Aiming at controlling the numbers of modules and their interface types directly, Wang tiles appear to be a natural approach.

Therefore, we further restrict our formalism of describing modular assembly plans to Wang tilings. For the sake of demonstration, we consider here the complete set of corner Wang tiles over two colors, recall Section 4.2.1. Consequently, we have $n_t = 16$ together with four independent horizontal and vertical edge types. Discretized by trusses, these modules are compatible by definition over the matching edges in the sense of generating a statically admissible ground structure.

In this section, we first suitably modify the SOCP formulation for truss topology optimization (4.2) to account for structural modularity and to handle multiple loading conditions, stress constraints, and module reusability. Ultimately, we develop a bilevel optimization approach to optimize the topologies of all the module types and their assemblies simultaneously.

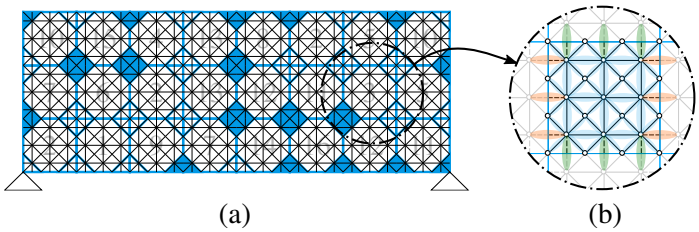


Figure 4.3: (a) Partitioning of a design domain into modules. (b) Bars forming each module split up into two sets—module-associated (drawn as solid lines highlighted in blue), and edge-associated (drawn as dashed lines with green and red backgrounds).

4.3.1 Truss topology optimization extended to structural modularity

Structural modularity is inherently prescribed in the form of equality constraints of certain cross-section areas, consequently reducing the number of *unique* cross-section areas within the ground structure. The developed formulation preserves its convexity and applies when a fixed module assembly plan $\mathbf{C}$ is specified a priori.

In the following text, we consider that all truss modules share the same *module ground structure* inspired by the union-jack lattice [81], see Fig. 4.3. Another ground structure can be adopted though, if needed.

To secure the mechanical compatibility of modules over their edges, the bars within each module split up into two sets. Those located entirely inside a module—the *module-associated* bars—occur only in the specific module type, and are highlighted in blue in Fig. 4.3b. The second set contains bars that intersect module boundaries. To preserve constant cross-section areas of the inter-domain bars along their lengths, they need to be associated with the edge types, occurring in multiple modules. These *edge-associated* bars are highlighted in red and green in Fig. 4.3b. Distinction of horizontal (green) and vertical (red) bars is the consequence of forbidden rotations in Wang tiling formalism.

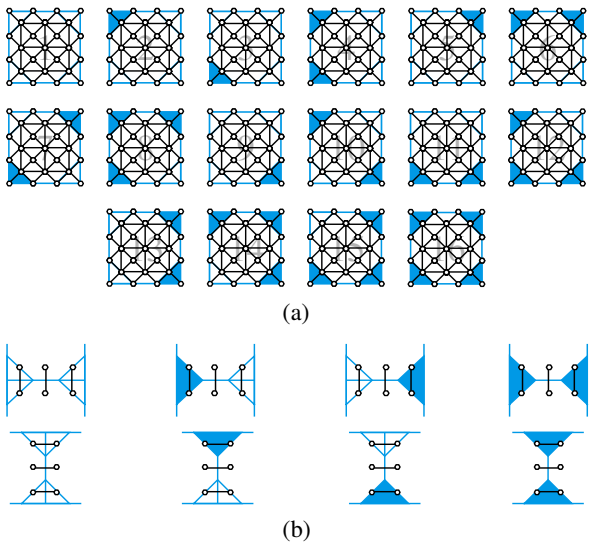


Figure 4.4: (a) Module- and (b) edge-associated bars. Scattered points represent nodes.

To achieve identical topology of the modules and their edge connections in all their occurrences, the bars are divided into groups, such that all bars in the same group share the same cross-section area. Assignment of bars to a particular group is provided by the group vector $\mathbf{g}(\mathbf{C}) \in \mathbb{N}^{n_\mathrm{b}}$, uniquely for each assembly plan $\mathbf{C}$. This group vector assigns a single number in the range of $\{1, \ldots, n_\mathrm{g}\}$ to each bar of the ground structure, where n_g denotes the number of groups, i.e., the number of unique cross-section areas.

For the complete tileset shown in Fig. 4.1 and the module ground structure, Fig. 4.3b, we have 48 bars associated with each module type, i.e., $16 \times 48 = 768$ groups of module-associated bars in total, see Fig. 4.4a. Similarly, each edge, either horizontal or vertical, accommodates 3 edge-associated bars, leading to $8 \times 3 = 24$ groups of edge-associated bars, see Fig. 4.4b. Consequently, we have $n_\mathrm{g} = 792$ for this specific choice of the tileset and the module ground structure.

Following the definition of the group vector, let $\mathbf{G} \in \mathbb{B}^{n_\mathrm{b} \times n_\mathrm{g}}$ denote the group matrix defined as

$$G_{i,j}(\mathbf{C}) = \begin{cases} 0 & \text{if } j \neq g_i(\mathbf{C}), \\ 1 & \text{if } j = g_i(\mathbf{C}), \end{cases} \quad \forall i \in \{1, \ldots, n_\mathrm{b}\}, \forall j \in \{1, \ldots, n_\mathrm{g}\}, \tag{4.3}$$

with $G_{i,j}(\mathbf{C})$ being the element in the i-th row and j-th column of the group matrix $\mathbf{G}(\mathbf{C})$, and g_i standing for the i-th element in $\mathbf{g}$. The group matrix represents a linear transformation mapping the space of the unique cross-section areas $\mathbf{a}_\mathrm{g} \in \mathbb{R}^{n_\mathrm{g}}$ into the space of all cross-section areas $\mathbf{a} \in \mathbb{R}^{n_\mathrm{b}}$,

$$\mathbf{a} = \mathbf{G}(\mathbf{C})\mathbf{a}_\mathrm{g}. \tag{4.4}$$

Because of modularity, the original topology optimization formulation (4.2) must be modified, reducing the number of cross-section areas from n_b to n_g, as they are substituted by unique cross-section areas[3]. Moreover, a similar procedure can be applied to reduce the number of the complementary strain energies of bars $\mathbf{w} \in \mathbb{R}^{n_\mathrm{b}}$ to $\mathbf{w}_\mathrm{g} \in \mathbb{R}^{n_\mathrm{g}}$, where $w_{\mathrm{g},j}$ is the complementary strain energy of all the bars in group j. Consequently, the objective function reads as

$$\mathbf{1}^\mathrm{T}\mathbf{w}_\mathrm{g}, \tag{4.5}$$

and the volume constraint (4.2b) transforms into

$$\boldsymbol{\ell}^\mathrm{T}\mathbf{G}(\mathbf{C})\mathbf{a}_\mathrm{g} \leq \overline{V}. \tag{4.6}$$

For the second-order conic constraints (4.2d), we follow the aggregation in Appendix 4.A to receive

$$w_{\mathrm{g},j} + a_{\mathrm{g},j} \geq \left\| \begin{pmatrix} w_{\mathrm{g},j} - a_{\mathrm{g},j} \\ \mathbf{G}_{:,j}(\mathbf{C}) \odot [2\boldsymbol{\ell} \oslash \mathbf{E}]^{\circ\frac{1}{2}} \odot \mathbf{s} \end{pmatrix} \right\|_2, \ \forall j \in \{1, \ldots, n_\mathrm{g}\}, \tag{4.7}$$

with $\oslash$, $\odot$, and $^\circ$ denoting the Hadamard (element-wise) division, multiplication, and power; and $\mathbf{G}_{:,j}$ being the j-th column of $\mathbf{G}$.

The final formulation of truss topology optimization extended to structural modularity

[3]Note that if $n_\mathrm{g} = n_\mathrm{b}$, the problem simplifies back to non-modular design.

then reads

$$\min_{\mathbf{a}_\mathrm{g},\mathbf{w}_\mathrm{g},\mathbf{s}} \ \mathbf{1}^\mathrm{T}\mathbf{w}_\mathrm{g} \tag{4.8a}$$

$$\text{s.t.} \quad \boldsymbol{\ell}^\mathrm{T}\mathbf{G}(\mathbf{C})\mathbf{a}_\mathrm{g} \leq \overline{V}, \tag{4.8b}$$

$$\mathbf{A}\mathbf{s} = \mathbf{f}, \tag{4.8c}$$

$$w_{\mathrm{g},j} + a_{\mathrm{g},j} \geq \left\| \begin{pmatrix} w_{\mathrm{g},j} - a_{\mathrm{g},j} \\ \mathbf{G}_{:,j}(\mathbf{C}) \odot [2\boldsymbol{\ell} \oslash \mathbf{E}]^{\circ\frac{1}{2}} \odot \mathbf{s} \end{pmatrix} \right\|_2, \ \forall j \in \{1,\dots,n_\mathrm{g}\}, \tag{4.8d}$$

$$\mathbf{a}_\mathrm{g} \geq \mathbf{0}. \tag{4.8e}$$

Because the number of constraints and variables in (4.8) decreases compared to the non-modular design (4.2), finding the optimal topology of a modular structure is faster than obtaining the optimal non-modular design. The acceleration factor depends on the repeatability of individual module types and, of course, on the optimization solver employed.

4.3.2 Handling stress constraints, multiple load cases, and module reusability

In addition to modularity, we extend formulation (4.8) to handle stress constraints, multiple loading scenarios, and module reusability among multiple structures. Most importantly, all these extensions preserve the convexity of the optimization problem, and thus do not compromise the efficiency of the solution.

In the case of stress constraints, let us assume that

$$\sigma_\mathrm{L} \leq \sigma_i \leq \sigma_\mathrm{U}, \quad \forall i \in \{1,\dots,n_\mathrm{b}\}, \tag{4.9}$$

where σ_L and σ_U are the bounds for stress in element i, σ_i. To maintain convexity and avoid additional variables, we can multiply the inequality (4.9) by a_i and constrain the internal forces, s_i, instead of the stress variables, σ_i, which provides us with equivalent convex linear inequalities

$$\sigma_\mathrm{L}\mathbf{G}(\mathbf{C})\mathbf{a}_\mathrm{g} \leq \mathbf{s} \leq \sigma_\mathrm{U}\mathbf{G}(\mathbf{C})\mathbf{a}_\mathrm{g}. \tag{4.10}$$

Based on these inequalities, the stresses vanish with zero cross-sections, because $a_i = \mathbf{G}_{i,:}(\mathbf{C})\mathbf{a}_\mathrm{g} = 0$ implies $s_i = 0$.

For multiple loading conditions and module reusability, we assume that there are $n_\mathrm{s} \in \mathbb{N}$ structures subjected to $n_\mathrm{lc} \in \mathbb{N}$ load cases, and we minimize the weighted average of the complementary strain energies of all the load cases with the weights[4] $\boldsymbol{\omega} \in \mathbb{R}_{>0}^{n_\mathrm{lc}\times n_\mathrm{s}}$. To minimize the number of design variables and constraints, we aggregate the complementary strain energies of bars with the same cross-sections across load cases. Because this aggregation follows the steps already outlined in Appendix 4.A, we omit the details here for the sake of brevity.

The final convex second-order conic formulation for optimization of trusses with prescribed modularity and subjected to stress constraints, multiple loading scenarios, and allowing for module reusability reads as (4.11), where $\mathbf{s}_k$ and $\mathbf{f}_k$ stand for the axial and nodal forces associated with the k-th load case. Moreover, the superscript $\bullet^{(m)}$ associates the variable $\bullet$ with the m-th structure.

[4] These weights are fixed in advance to prioritize individual load cases, for example, based on their time of occurrence.

$$\begin{pmatrix} \mathbf{a}_\mathrm{g}^* \\ \mathbf{w}_\mathrm{g}^* \\ \mathbf{s}^* \end{pmatrix} \in \min_{\mathbf{a}_\mathrm{g},\mathbf{w}_\mathrm{g},\mathbf{s}} \quad \mathbf{1}^\mathrm{T}\mathbf{w}_\mathrm{g} \tag{4.11a}$$

$$\text{s.t.} \quad \sum_{m=1}^{n_\mathrm{s}} \left(\boldsymbol{\ell}^{(m)}\right)^\mathrm{T} \left[\mathbf{G}(\mathbf{C})\right]^{(m)} \mathbf{a}_\mathrm{g} \leq \overline{V}, \tag{4.11b}$$

$$\mathbf{A}\mathbf{s}_k^{(m)} = \mathbf{f}_k^{(m)}, \ \forall k \in \{1,\dots,n_\mathrm{lc}\}, \ \forall m \in \{1,\dots,n_\mathrm{s}\}, \tag{4.11c}$$

$$w_{\mathrm{g},j} + a_{\mathrm{g},j} \geq \left\| \begin{pmatrix} w_{\mathrm{g},j} - a_{\mathrm{g},j} \\ \omega_{1,1} \left[\mathbf{G}_{:,j}(\mathbf{C})\right]^{(1)} \odot \left[2\boldsymbol{\ell}^{(1)} \oslash \mathbf{E}\right]^{\circ\frac{1}{2}} \odot \mathbf{s}_1^{(1)} \\ \vdots \\ \omega_{n_\mathrm{lc},1} \left[\mathbf{G}_{:,j}(\mathbf{C})\right]^{(1)} \odot \left[2\boldsymbol{\ell}^{(1)} \oslash \mathbf{E}\right]^{\circ\frac{1}{2}} \odot \mathbf{s}_{n_\mathrm{lc}}^{(1)} \\ \omega_{1,2} \left[\mathbf{G}_{:,j}(\mathbf{C})\right]^{(2)} \odot \left[2\boldsymbol{\ell}^{(2)} \oslash \mathbf{E}\right]^{\circ\frac{1}{2}} \odot \mathbf{s}_1^{(2)} \\ \vdots \\ \omega_{n_\mathrm{lc},n_\mathrm{s}} \left[\mathbf{G}_{:,j}(\mathbf{C})\right]^{(n_\mathrm{s})} \odot \left[2\boldsymbol{\ell}^{(n_\mathrm{s})} \oslash \mathbf{E}\right]^{\circ\frac{1}{2}} \odot \mathbf{s}_{n_\mathrm{lc}}^{(n_\mathrm{s})} \end{pmatrix} \right\|_2 , \quad \forall j \in \{1,\dots,n_\mathrm{g}\}, \tag{4.11d}$$

$$\sigma_\mathrm{L} \left[\mathbf{G}(\mathbf{C})\right]^{(m)} \mathbf{a}_\mathrm{g} \leq \mathbf{s}_k^{(m)} \leq \sigma_\mathrm{U} \left[\mathbf{G}(\mathbf{C})\right]^{(m)} \mathbf{a}_\mathrm{g}, \quad \forall k \in \{1,\dots,n_\mathrm{lc}\}, \forall m \in \{1,\dots,n_\mathrm{s}\}, \tag{4.11e}$$

$$\mathbf{a}_\mathrm{g} \geq \mathbf{0}. \tag{4.11f}$$

4.3.3 Modular-topology optimization

The objective function of the optimal design obtained by solving (4.8) or (4.11) depends inherently on the specified assembly plan of modules $\mathbf{C}$. However, because the number of potential valid assemblies increases with the number of entries in $\mathbf{C}$ exponentially, exploring all possible combinations may be intractable. Therefore, a method to efficiently find a "good" connectivity matrix must be developed. In this section, we propose an approach to solve this bilevel optimization problem, i.e., optimizing the module topologies as well as their assembly simultaneously. While the lower-level problem (4.11) exhibits convexity, the upper-level is combinatorial and NP-hard in general [49]. Therefore, we propose tackling the problem with a combination of mathematical programming and meta-heuristics.

The bilevel optimization problem then reads

$$\mathbf{C}^* \in \arg\min_{\mathbf{C}} \ \mathbf{1}^\mathrm{T}\mathbf{w}_\mathrm{g}^*(\mathbf{C}), \tag{4.12}$$

with w^* following from (4.11), and the globally optimal design is eventually recovered as

$$\mathbf{a}^* = \mathbf{G}(\mathbf{C}^*)\mathbf{a}_\mathrm{g}^*(\mathbf{C}^*). \tag{4.13}$$

The problem (4.12) is solved with (i) the globally optimal connectivity matrix $\mathbf{C}^*$ and with (ii) the globally optimal vector of unique cross-section areas $\mathbf{a}_\mathrm{g}^*$ at the globally optimal complementary strain energy c^*.

Algorithm 4.1 Genetic Algorithm

```
function GeneticAlgorithm(C, n_pop, n_gen, n_t, p_t, p_c, p_m)
    population ← RandomPopulation(C, n_pop)
    fitness ← PopulationFitness(population)
    for δ ← {1, ..., n_gen} do
        elite ← EliteIndividual(population, fitness)
        matingPool ← Selection(population, fitness, n_t, p_t)
        population ← CrossOver(matingPool, p_c)
        population ← Mutation(population, p_m)
        population ← AppendElite(population, elite)
        population ← Diversify(population)
        fitness ← PopulationFitness(population)
    end for
    return C ← EliteIndividual(population)
end function
```

To approximately solve the upper-level assembly problem, we adopt the genetic algorithm (GA) [87], a stochastic meta-heuristic optimization algorithm that simulates the process of evolution by following Darwin's "survival of the fittest" rule [180]. Genetic algorithms receive high recognition in combinatorial and multi-objective optimizations, as they handle discrete, non-differentiable, and non-convex optimization problems fairly efficiently. Here, we implement the standard GA routine [87], see Algorithm 4.1, consisting of the following steps: (i) First, we generate a random population of n_{pop} individuals (connectivity matrices) that evolve through n_{gen} generations. (ii) Fitness of individuals, inversely proportional to the complementary strain energy, result from a parallel solution to (4.11). Next, (iii) tournament selection is performed, controlled by probability p_{t} and the number of competitors n_{t}. Both the (iv) cross-over and (v) mutation operators are applied, governed by the probabilities p_{c} and p_{m}, and following the standard implementations. We also enforce population diversity, so that duplicate individuals get substituted by random ones, and elitism applies. Specific numerical values of these parameters appear in Appendix 4.B. Finally, we note here that one may sacrifice optimized design performance to shorter computation times by reducing the number of generations and of individuals.

4.4 Examples

The proposed modular-topology optimization approach was implemented in MATLAB, with the source codes available at [193], and applied to three illustrative two-dimensional problems. In Section 4.4.1, we consider a hinge-supported beam subjected to a single load case without stress constraints. In this setting, we first adopt a coarse discretization, as this allows us to obtain a globally optimal design through brute-force enumeration. Subsequently, we assume a finer discretization and discuss the results. Section 4.4.2 investigates an L-shaped domain with two load cases and stress constraints. The third problem in Section 4.4.3 aims at designing modules and assemblies that are reusable for both of the former cases.

All computations were performed on a Linux workstation with two Intel® Xeon® E5-2630 processors. The second-order cone programs for topology optimization of trusses were solved

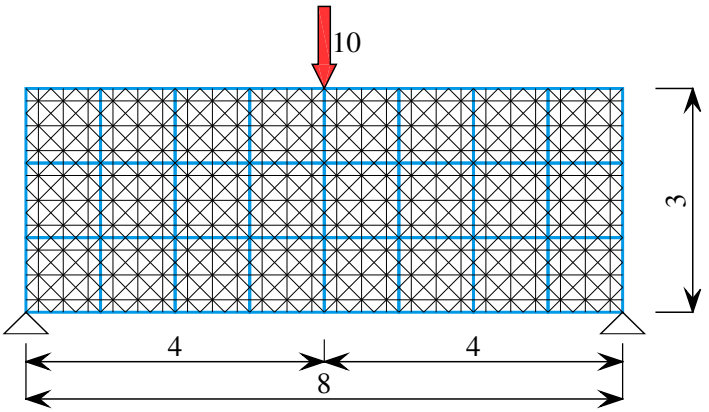


Figure 4.5: Dimensions, discretization into modules, boundary conditions, and ground structure of the coarsely discretized beam.

using the state-of-the-art MOSEK optimizer [138], interfaced with MATLAB via the YALMIP toolbox [131]. All times reported in this section include launching and termination of MATLAB and its parallel pool.

4.4.1 Hinge-supported beam

4.4.1.1 Coarse discretization

As the first illustration, we investigate a simply-supported beam of dimensions 8×3, see Fig. 4.5. Under coarse discretization, the beam splits up into 24 unit-size square modules that follow the corner Wang tiling formalism, recall Fig. 4.1. We assume the module ground structures shown in Fig. 4.3b, with Young's modulus E of each bar equal to 1. The beam is supported with two hinges at the very bottom-left and bottom-right corners, and loaded with an external force of magnitude 10 at the mid-span of the top edge.

Bounds on the global optimum. Structural modularity comes at a price of higher complementary strain energy when compared to the non-modular design [89]. Because all the modules share identical module ground structures, performing topology optimization without modularity constraints, recall Eq. (4.2), provides, in 0.1 s, the lower-bound energy $\underline{c} = 61.9$ with the design shown in Fig. 4.6a. Analogously, the upper-bound complementary strain energy arises in the topology optimization of the design domain assembled from a single module type, indicated by the connectivity matrix $\mathbf{C}$ containing all-zeros or all-ones. In this setting, topology optimization results in the optimal design depicted in Fig. 4.6b, with $\overline{c} = 191.2$ obtained in 0.1 s. Therefore, the strain energy of the optimal modular design must lie in the interval $c^* \in [61.9, 191.2]$.

We wish to emphasize here that the initial choice of the module ground structure and of the modules aspect ratio strongly influences which designs can emerge in the optimization. On one hand, this choice brings the benefit of a simple method to ensure manufacturability by 3D printing [191]. On the other hand, truss discretization provides suboptimal designs in general. For example, the module ground structures used in this work, recall Fig. 4.3b, exclude the analytically optimal von Mises truss of compliance 34.7. This choice was made intentionally: knowledge of analytical optima is seldom available and convergence of the bilevel optimization

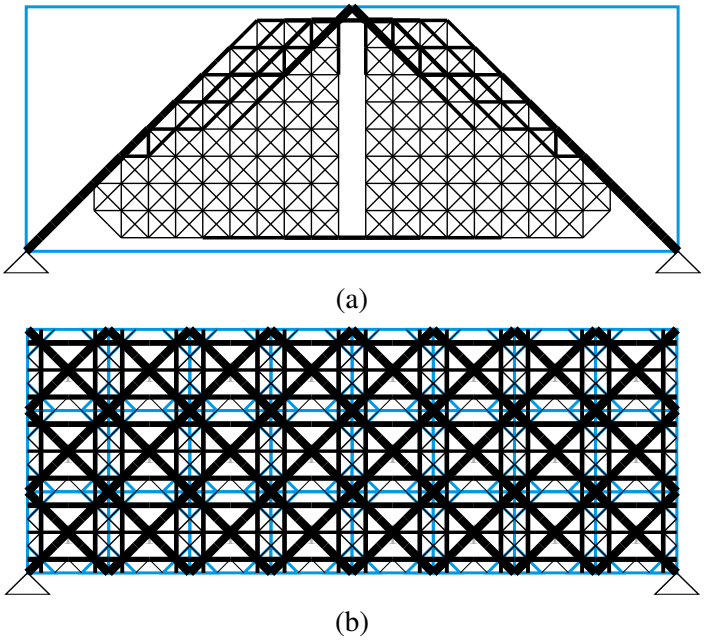


Figure 4.6: (a) Lower-bound non-modular, and (b) worst-case designs of the evaluated coarsely discretized beam with complementary strain energies $\underline{c} = 61.9$ and $\overline{c} = 191.2$, respectively.

is immediate in this case, which can be verified by running the `MTSimpleBeamAR.m` example available in the supplementary implementation [193].

Complete enumeration. Because the discretization is coarse enough, the globally optimal connectivity matrix $\mathbf{C}^*$ and the corresponding complementary strain energy c^* can be determined using the complete enumeration, which allows us to assess the performance of the bilevel optimization rigorously. Notice that even the coarse discretization permits 2^{36} possible connectivity matrices, preventing a complete enumeration in a reasonable time. To reduce the complexity, we have enforced symmetry of the connectivity matrix $\mathbf{C}$ with respect to the vertical axis of the beam, consequently reducing the number of combinations to 2^{20} feasible assemblies. Because all modules comprise identical module ground structures, the number of combinations is further decreased by noticing that the vertex types lack any physical meaning. This makes the problem invariant against the coloring of Wang tiles corners, i.e., $\mathbf{w}_\mathrm{g}(\mathbf{C}) = \mathbf{w}_\mathrm{g}(\overline{\mathbf{C}})$, where the connectivity matrix $\overline{\mathbf{C}}$ follows from $\mathbf{C}$ by inverting 0 to 1 and vice versa. Subsequently, we need to enumerate 2^{19} distinct combinations instead of the original 2^{36}.

Evaluations of the optimization problem (4.11) for all the combinations took 9.5 core hours. During the enumeration, a globally optimal design of the complementary strain energy $c^* = 62.7$ was obtained, see Fig. 4.7a, yielding a 1.3% increase of the objective function compared to the lower-bound design. The module set of the globally optimal design, shown in Fig. 4.7b, contains 13 modules that contribute to the load transfer, allowing for potential elimination of three empty modules from the set. One of these, module 1, is not used because of the edge-matching rules. Overall, the enumerated combinations generate a nearly Gaussian distribution with a mean value of 107.7 and a standard deviation of 14.6, see Fig. 4.8.

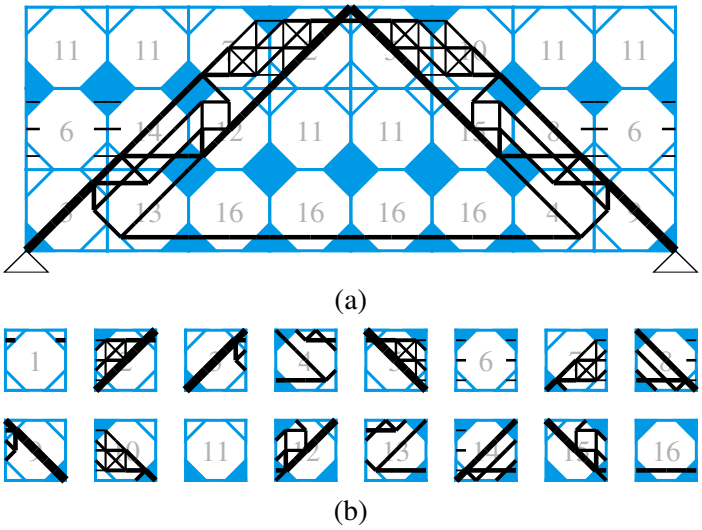


Figure 4.7: Globally optimal design for the (a) coarsely discretized beam of $c^* = 62.7$, and (b) the corresponding tileset.

Bilevel optimization using GA. The bilevel optimization solver was launched 50 times, each time with a random population of 16 individuals, evaluating the statistical properties of the bilevel optimization approach. On average, a single run took 11 core minutes. The distribution of the complementary strain energy of the best individual within the population is shown in Fig. 4.9.

The initial random populations yielded topologies with a mean energy of 107.4, approximately matching the mean value 107.7 of the nearly Gaussian distribution of the complete enumeration. Throughout the prescribed 40 generations of the genetic algorithm, the complementary strain energy decreased to the final mean value of the best individual, 67.4, being on average 8.9% higher than the lower-bound solution and 7.5% higher than the global optimum. Through bilevel optimization, a second-best design, with strain energy 64.6, was obtained. All the achieved objectives are within the lowest 0.2% of all combinations, recall Fig. 4.8.

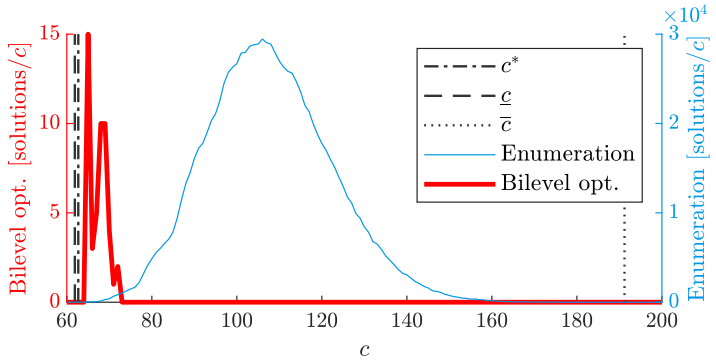


Figure 4.8: Distribution of optimal complementary strain energies c of all enumerated combinations (in blue), and 50 independent runs of the bilevel optimization (in red). While $\overline{c}$ and c^* denote the complementary strain energies of the worst and best modular designs, $\underline{c}$ is the complementary strain energy of the non-modular one.

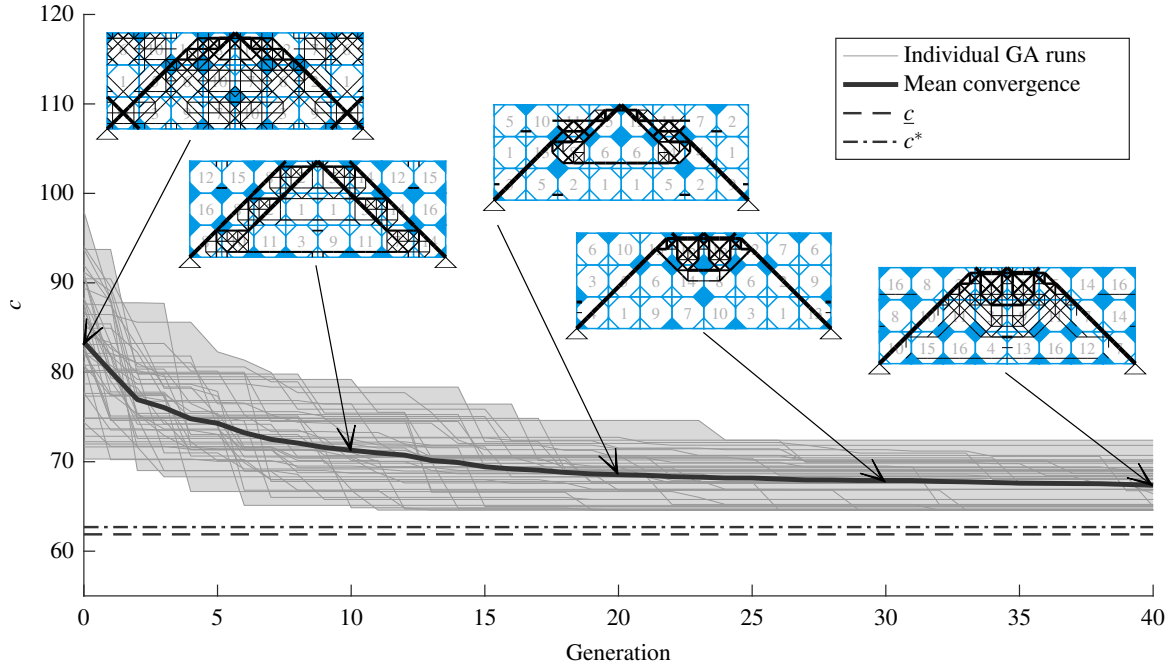
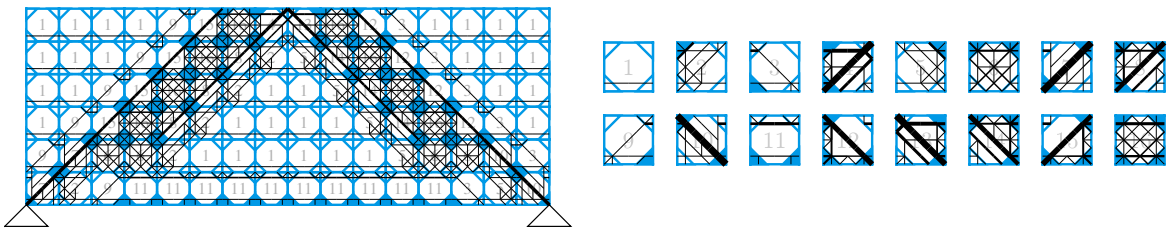


Figure 4.9: Convergence of complementary strain energies c of the best individuals with 50 independent runs of the genetic algorithm; c^* denotes the complementary strain energies for the best modular design and $\underline{c}$ stands for the complementary strain energy of the non-modular one.

4.4.1.2 Fine discretization

Let us now consider a beam with the same dimensions and with identical boundary conditions as in the previous subsection, recall Fig. 4.5, but with a refined discretization 6×16 and modules with dimensions 0.5×0.5. To preserve comparability with the previous case, the connectivity matrix again satisfies symmetry along the midspan of the beam. Consequently, fine discretization permits 2^{62} distinct combinations of assemblies.

This huge number of combinations, pronounced further with an increased number of degrees of freedom, makes it impossible to perform the complete enumeration as in the previous case, leaving us without the knowledge of a guaranteed global optimum. However, similarly to the previous example, we can obtain the bounds on the optimum: $\underline{c} = 61.1$ obtained in 0.7 s and $\overline{c} = 228.7$ reached in 0.4 s, implying that $c^* \in [61.1; 228.7]$.



(a) (b)

Figure 4.10: Best design of the (a) finely discretized beam with $c = 71.6$ as obtained via bilevel optimization, and (b) the corresponding tile set.

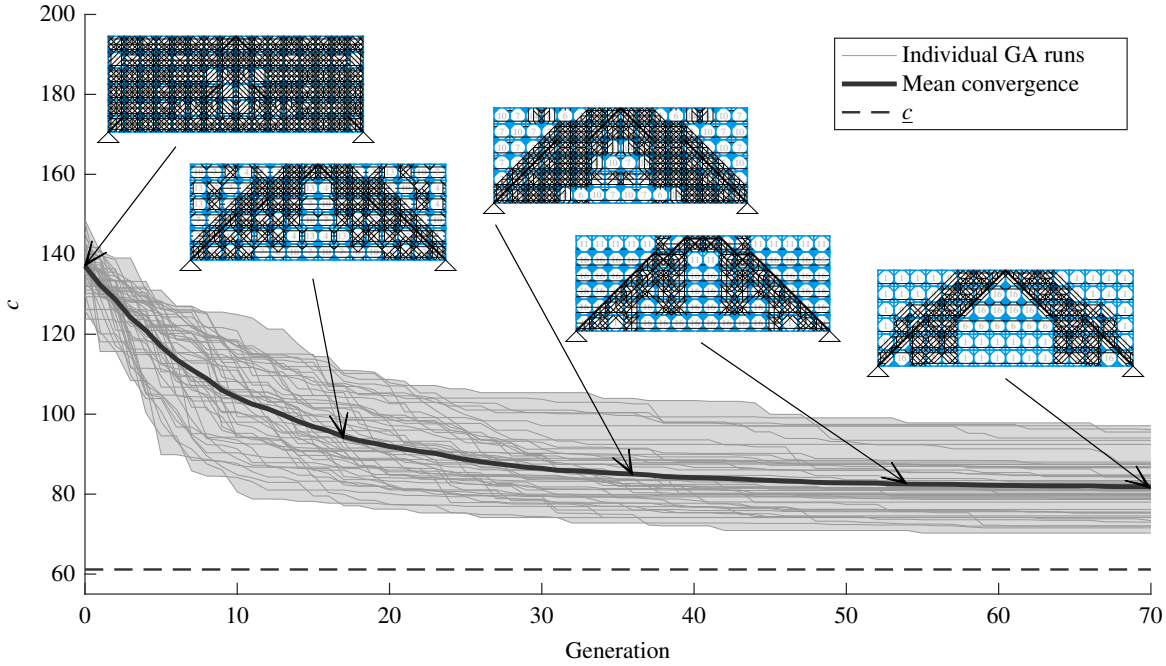


Figure 4.11: Convergence of complementary strain energies c of the best individuals with 50 independent runs of the genetic algorithm for the finely discretized beam; $\underline{c}$ stands for the complementary strain energy of the optimal non-modular design. For details in this figure, we kindly ask the reader to consult the online version of the article.

Compared to the coarse problem, fine discretization produces a richer ground structure, which allows the algorithm to reach a decreased lower-bound complementary strain energy. Conversely, the upper-bound energy noticeably increases because a larger ratio of material volume $\overline{V}$ appears to be placed inefficiently. Similar consequences of modularity also emerged in [6, 89].

Convergence of the bilevel optimization algorithm, independently launched 50 times, is shown in Fig. 4.11. A mean run took 1.6 core hours. The initial random populations of 29 individuals determined designs of the mean objective value 156.5—a significant increase (45.3%) compared to coarse discretization. Throughout 70 generations, bilevel optimization converged to mean objective value of 82.1, being 34.4% more-compliant than the lower-bound design. The best design achieved, with $c = 71.6$, is shown in Fig. 4.10, which amounts to a 17.0% increase over the lower-bound complementary strain energy $\underline{c}$.

4.4.2 L-shaped beam with stress constraints and multiple load cases

As the second illustrative problem, we assume an L-shaped design domain as shown in Fig. 4.12. For this domain, two equally-weighted load cases, indicated by the two arrows in Fig. 4.12, apply. Furthermore, we limit the structural volume by $\overline{V} = 100$, set the Young modulus to $E = 1$, and fix the maximum value of stress to $\sigma_{\mathrm{UB}} = -\sigma_{\mathrm{LB}} = 20$. Although the maximum (absolute value of) stress equals to 4.6 in the lower-bound non-modular setting, which makes the stress constraint inactive, these constraints become active for some modular designs. For example, the worst-case modular design would yield a maximum stress of 39.4 without stress constraints. When imposed, the worst-case modular complementary strain energy approaches

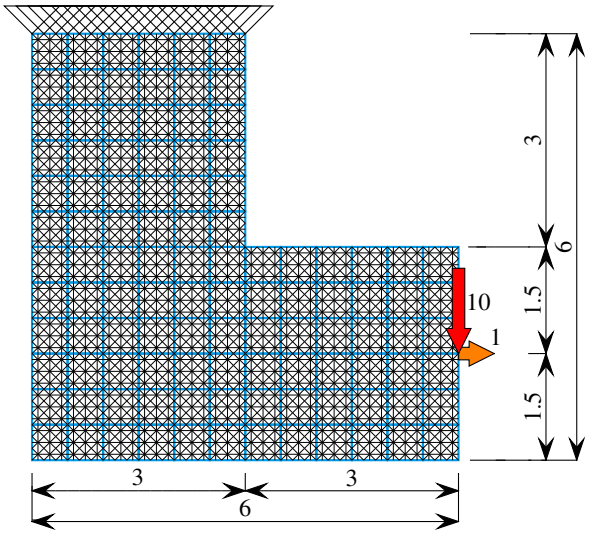


Figure 4.12: Dimensions, discretization into modules, boundary conditions, and ground structure of the L-shaped domain.

$\overline{c} = 1837.9$ in 2.2 s and both the stress constraint bounds are active. Because $\underline{c} = 552.3$ arises from the lower-bound non-modular design in 2.2 s, the optimum lies in $c^* \in [552.3, 1837.9]$.

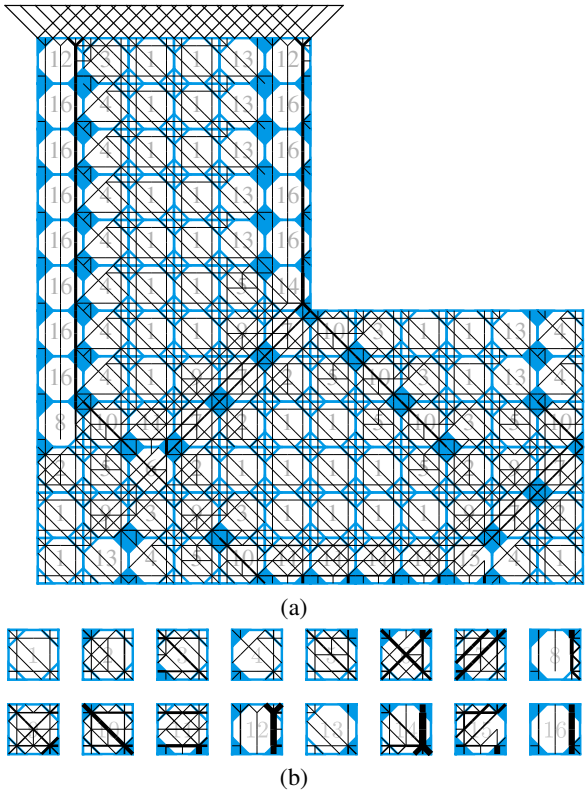


Figure 4.13: Best design of the (a) evaluated L-shaped beam with $c = 803.1$ as obtained via bilevel optimization, and (b) the corresponding tile set.

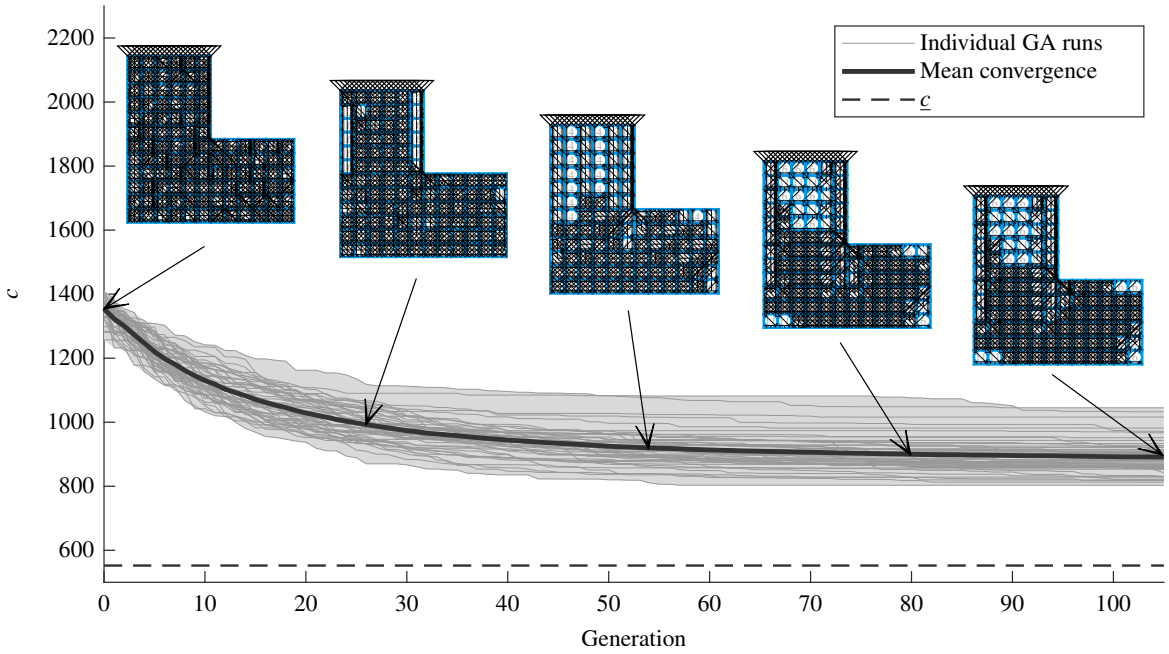


Figure 4.14: Convergence of complementary strain energies c of the best individuals with 50 independent runs of the genetic algorithm for an L-shaped beam; $\underline{c}$ stands for the complementary strain energy of the optimal non-modular design. For details in this figure, we kindly ask the reader to consult the online version of the article.

After launching the bilevel optimization approach, the first generation of random individuals yielded a mean energy of 1468.4, a value not too distant from the worst case. Through 106 generations of 42 individuals, the population evolved to set the mean energy of the best individuals at 891.0, including a best design of 803.1 (45.4% higher than $\underline{c}$), and the worst design had a complementary strain energy of 1045.0. See Fig. 4.14 for the statistics of the best individuals within 50 independent random runs of the algorithm. A mean run of the algorithm terminated in 27.8 core hours.

While the best design, Fig. 4.13, clearly aligns structural stiffness with the principal stress direction of the first load case (e.g., modules 10, 11, 14, 16), the interior of the module types 1, 4, and 13 serves mainly as a structural stabilization against the second load case. Stress constraints bounds are active both in tension and compression.

4.4.3 Module reusability in simply-supported and L-shaped beams

The final example concerns the concurrent design of the finely discretized beam from Section 4.4.1.2 with the L-shaped domain from Section 4.4.2. In this case, the modules become reusable among these two domains, which is a key benefit of modularity. Additionally, we introduce three minor deviations from the settings of the original problems: the stress constraints of the L-shaped domain also apply to the hinge-supported beam; the hinge supported beam does not enforce symmetric colorings; and instead of independent volume constraints, we constrain the overall volume by $\overline{V} = 200$. All these changes are justified by practical considerations: stress constraints should apply over the same material, symmetric coloring of the simply-supported beam may become inefficient as the L-shaped domain lacks the symmetry, and (one of the)

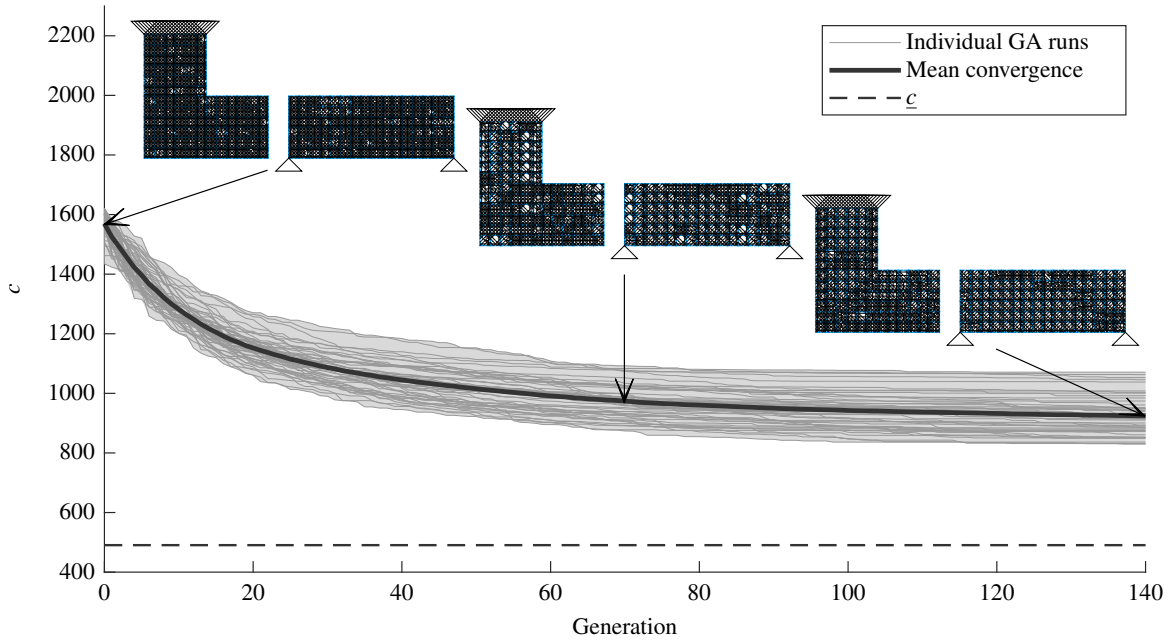


Figure 4.15: Convergence of complementary strain energies c of the best individuals with 50 independent runs of the genetic algorithm for reusable design; $\underline{c}$ stands for the complementary strain energy of the optimal non-modular design. For details in this figure, we kindly ask the reader to consult the online version of the article.

volume constraints may become inactive. For these reasons, the designs in this section are not directly comparable with the previous ones. However, the original constraints can still be imposed, with few adjustments in the code.

In the modular-topology optimization framework, we obtained the lower-bound complementary strain energy of $\underline{c} = 490.5$ in 4.8 s and a worst-case design with $\overline{c} = 2056.8$ in 3.5 s, implying that $c^* \in [490.5, 2056.8]$. In 50 independent runs of the algorithm, lasting 155 core hours on average, the originally random population of 57 individuals evolved in 140 generations from an initial mean complementary strain energy of 1695.1 to the best individual having the complementary strain energy of 924.5, see Fig. 4.15. The best design acquired, Fig. 4.16, exhibited a strain energy of 829.6, which is 69% worse than the lower-bound design. Modules in the design of Fig. 4.16b are clearly distinguished by their major effective stiffness

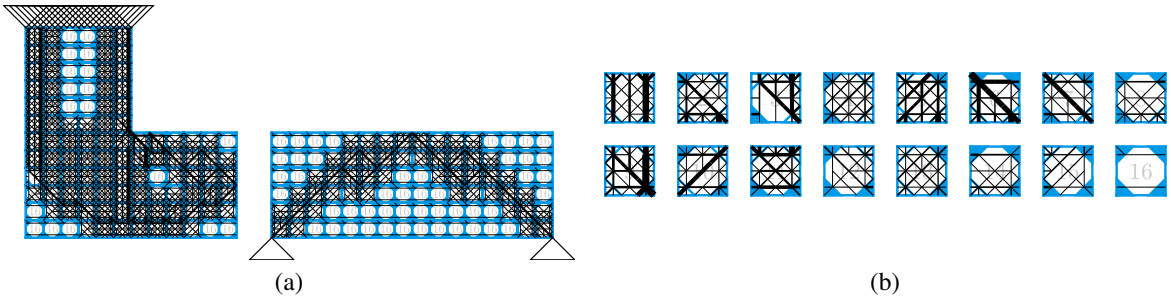


Figure 4.16: (a) Best reusable designs of the two domains with $c = 829.6$ as obtained via bilevel optimization, and (b) the corresponding tile set.

directions: the vertical direction of module 1; the almost horizontal direction of modules 11 and 16; and the rest inclined (2, 3, 5, 6, 7, 9, 10, 12, 15), or stiff in three (8, 14) or all directions (4, 13).

4.5 Conclusions

In this paper, we introduced a novel bilevel modular-topology optimization approach, facilitating simultaneous optimization of the topologies of 16 independent truss modules together with their optimal placement within their respective structural macro-scale domains. This method adopts the concept of corner Wang tiles as a suitable formalism for describing non-periodic assemblies of structural modules to maintain edge compatibility: a new class of compatible and reconfigurable (micro-)structures.

Lower-level optimization constitutes the truss topology least-compliant design problem, extended to structural modularity, stress constraints, multiple load cases, and modules reusability. We formulate this optimization problem as a convex second-order cone program (SOCP), efficiently solvable to global optimality by employing modern mathematical programming solvers. In addition, modularity enables us to aggregate the constraints and variables of the original non-modular problem, resulting in the optimal modular designs being found faster compared to their non-modular counterparts. Since the compliances of modular designs are bounded by the periodic-unit-cell (PUC) design from above and by the non-modular design from below, any (and even random) assembly plan balances the solution efficiency of PUC with the design performance of the non-modular design. The final optimized design quality thus strongly depends upon the supplied assembly plan. To mitigate and take advantage of this dependence, we have developed a bilevel modular-topology optimization framework, using meta-heuristics (namely, a genetic algorithm) to search for an optimal assembly plan.

After implementing this approach in MATLAB, we assess its performance using three sample problems. For the first, we consider a hinge-supported beam. For the case illustrating coarse discretization and symmetric module interfaces, we compute the globally optimal design using brute-force enumeration. It turns out that the optimal modular design almost achieves the quality of a non-modular one. The bilevel optimization approach converges to solutions near the optimum, and is, therefore, suitable for finding an approximate solution to the optimization problem in much shorter times compared to the enumeration. When a finer discretization for the hinge-supported beam is adopted, the quality of the optimized modular design decreases compared to coarse discretization. This issue seems to be a common drawback of modularity, as reported earlier by Huang and Xie [89]. For the second and third problems, which include an L-shape design domain, we demonstrated how stress constraints, multiple load cases, and module reusability can be imposed while maintaining convexity (and thus solution efficiency) for the inner optimization problem. For both of these sample problems, the optimized designs outperform the optimal PUC designs considerably, and exhibit structural efficiency and material distribution nearly equivalent to the non-modular designs.

In the future, several important extensions need to be considered to build upon these pilot results. First, upper-level optimization (GA) could be replaced by a heuristic procedure based on free-material optimization [231, 88] or by machine learning to allow more efficient solution of larger problems. Tuning such procedures can beneficially incorporate knowledge acquired by our extensive exploration of the design space of module assemblies. Machine learning may also allow for us to design the topology of individual modules [80]. Second,

since the chosen Wang tileset plays a crucial role, different tilesets may yield different optimal designs. Adaptive choice of a set cardinality and distribution of color codes within set is worth investigating. Similarly, the module ground structure as well as the shape of the modules influence the final design; for instance, in the first example of the hinge-supported beam, the best modular design always tends to be the two-bar truss when a correct aspect ratio of the modules and a suitable module ground structure is used. Another important issue that deserves investigation is the decrease in stiffness under the refinement of the macroscopic assembly. Here, the influence of tileset cardinality and coloring as well as the module-assembly scale separation shall be investigated in particular.

The formulation and approach here presented should extend seamlessly to 3D. However, the scaling of the computational requirements required in the 3D case will likely require future developments or different approaches before it becomes practically usable. Then, optimized module sets could be applied in the modular design of 3D-printed LEGO®-like products [169] or combinatorial aperiodic metamaterials [43] with complex shapes [12]. Moreover, the inner SOCP formulation also allows for other convex extensions, e.g., the fundamental free-vibrations eigenfrequency lower-bound constraint [146, 191] and bounds on peak power [83]. An extension to modular buckling mechanisms [148] is another challenge. Last, adopting continuum topology optimization might also provide invaluable insights and broaden the potential of the proposed approach.

4.A Complementary strain energy conic constraints for modular designs

Although the original elastic design formulation (4.1) lacks convexity, it allows for reformulation into a convex conic optimization problem. Here, we consider dual complementary-strain-energy reformulation (4.2) that can exploit modularity by aggregation of constraints and design variables. This section of the appendix derives this aggregation and explains the basic mechanical reasoning behind the reformulation.

Let us assume a minimization of the structural complementary strain energy function, Eq. (4.2a), defined to be the sum of the upper bounds for the complementary strain energies w_i of individual bars $i \in \{1, \dots, n_\mathrm{b}\}$, i.e.,

$$w_i \geq \frac{1}{2} \frac{\ell_i}{E_i} \frac{s_i^2}{a_i}, \tag{4.14}$$

where ℓ_i, E_i, and a_i are the length, Young modulus, and the cross-section area of the i-th element, respectively. The axial force in this element is denoted by s_i. Notice that since we minimize the sum of the upper bounds w_i (4.2a), they attain the value of the complementary strain energy at the optimum, which is in turn equal to the compliance in (4.1).

Instead of minimizing the sum of complementary strain energies of individual bars we can, however, minimize the sum of aggregated complementary strain energies

$$w_{\mathrm{g},j} = [\mathbf{G}_{:,j}(\mathbf{C})]^\mathrm{T} \mathbf{w} \geq \frac{1}{2a_{\mathrm{g},j}} [\mathbf{G}_{:,j}(\mathbf{C})]^\mathrm{T} \left(\boldsymbol{\ell} \oslash \mathbf{E} \odot \mathbf{s}^{\circ 2} \right), \tag{4.15}$$

where $w_{\mathrm{g},j}$ is the upper bound for the sum of complementary strain energies of the bars that share the cross-section area $a_{\mathrm{g},j}$, and $\oslash$, $\odot$ with $^\circ$ are the Hadamard "element-wise" division,

multiplication, and power. This step effectively eliminates summands in the objective function and aggregates constraints as well as design variables.

Because (4.15) is not defined for $a_{\mathrm{g},j} = 0$, we perform a multiplication by the non-negative $4a_{\mathrm{g},j}$ to obtain

$$4w_{\mathrm{g},j}a_{\mathrm{g},j} \geq [\mathbf{G}_{:,j}(\mathbf{C})]^{\mathrm{T}} \left([2\boldsymbol{\ell} \oslash \mathbf{E}]^{\circ\frac{1}{2}} \odot \mathbf{s} \right)^{\circ 2}. \tag{4.16}$$

Eq. (4.16) now allows for zero cross-section areas as is required by topology optimization. However, in this case, the corresponding internal forces vanish and $w_{\mathrm{g},j}$ is arbitrary. Notice that $w_{\mathrm{g},j}$ may even attain arbitrarily low negative values, making the complementary strain energy functional non-physical and the objective function (4.2a) unbounded.

Because the aggregated constraints share the same cross-section, adding $w_{\mathrm{g},j}^2 - 2w_{\mathrm{g},j}a_{\mathrm{g},j} + a_{\mathrm{g},j}^2$ to both sides of the inequality provides us with the sum-of-squares inequality

$$(w_{\mathrm{g},j} + a_{\mathrm{g},j})^2 \geq (w_{\mathrm{g},j} - a_{\mathrm{g},j})^2 + [\mathbf{G}_{:,j}(\mathbf{C})]^{\mathrm{T}} \left([2\boldsymbol{\ell} \oslash \mathbf{E}]^{\circ\frac{1}{2}} \odot \mathbf{s} \right)^{\circ 2}, \tag{4.17}$$

which is equivalent to

$$w_{\mathrm{g},j}^{+} + a_{\mathrm{g},j}^{+} \geq \left\| \begin{pmatrix} w_{\mathrm{g},j}^{+} - a_{\mathrm{g},j}^{+} \\ \mathbf{G}_{:,j}(\mathbf{C}) \odot [2\boldsymbol{\ell} \oslash \mathbf{E}]^{\circ\frac{1}{2}} \odot \mathbf{s} \end{pmatrix} \right\|_2, \tag{4.18a}$$

$$-w_{\mathrm{g},j}^{-} - a_{\mathrm{g},j}^{-} \geq \left\| \begin{pmatrix} w_{\mathrm{g},j}^{-} - a_{\mathrm{g},j}^{-} \\ \mathbf{G}_{:,j}(\mathbf{C}) \odot [2\boldsymbol{\ell} \oslash \mathbf{E}]^{\circ\frac{1}{2}} \odot \mathbf{s} \end{pmatrix} \right\|_2, \tag{4.18b}$$

with

$$a_{\mathrm{g},j} = a_{\mathrm{g},j}^{+} - a_{\mathrm{g},j}^{-}, \tag{4.19a}$$

$$w_{\mathrm{g},j} = w_{\mathrm{g},j}^{+} - w_{\mathrm{g},j}^{-}, \tag{4.19b}$$

$$a_{\mathrm{g},j}^{+} \geq 0, a_{\mathrm{g},j}^{-} \geq 0, w_{\mathrm{g},j}^{+} \geq 0, w_{\mathrm{g},j}^{-} \geq 0. \tag{4.19c}$$

Clearly, (4.18b) is redundant, as both the $a_{\mathrm{g},j}$ and $w_{\mathrm{g},j}$ must be non-negative, i.e., $w_{\mathrm{g},j}^{-} = a_{\mathrm{g},j}^{-} = 0$. Moreover, if $\mathbf{a}_{\mathrm{g}} \geq \mathbf{0}$ is enforced explicitly, which is our case—recall Eq. (4.2e)—the non-physical situation of the negative complementary strain energy is automatically eliminated because the Euclidean norm is non-negative by definition. Consequently, we end up with the conic constraint

$$w_{\mathrm{g},j} + a_{\mathrm{g},j} \geq \left\| \begin{pmatrix} w_{\mathrm{g},j} - a_{\mathrm{g},j} \\ \mathbf{G}_{:,j}(\mathbf{C}) \odot [2\boldsymbol{\ell} \oslash \mathbf{E}]^{\circ\frac{1}{2}} \odot \mathbf{s} \end{pmatrix} \right\|_2, \tag{4.20}$$

which is convex and equivalent to (4.15) for all positive cross sections. For zero cross sections, complementary strain energy is implicitly enforced to be non-negative, and actually zero, since $w_{\mathrm{g},j}$ is to be minimized.

4.B Binary genetic algorithm

In the considered bilevel optimization problem, we use the following parameters for the genetic algorithm: The population consists of n_{pop} individuals, heuristically set to

$$n_{\mathrm{pop}} = \left\lfloor 3.6\sqrt{|\mathbf{C}|} + 0.5 \right\rfloor, \tag{4.21}$$

where $\lfloor \bullet \rfloor$ denotes rounding of $\bullet$ towards the nearest integer less than or equal to $\bullet$.

The individuals evolve through n_{gen} generations, where

$$n_{\text{gen}} = 5 \lfloor 0.49 n_{\text{pop}} + 0.5 \rfloor \, .in \tag{4.22}$$

Further, the *selection* of parents giving birth to offspring follows from tournament selection. The size of tournament n_{t} equals

$$n_{\text{t}} = \left\lfloor \frac{4}{3} \sqrt{|\mathbf{C}|} + 0.5 \right\rfloor, \tag{4.23}$$

and the individuals participating in the tournament are chosen randomly. Sorted accordingly to their fitness values, the probability p_i of the i-th individual to win the tournament equals

$$p_i = p_{\text{t}} \cdot (0.7)^i, \tag{4.24}$$

where $p_{\text{t}} = 0.3$.

For combinations of individuals, we used uniform *cross-over* with a probability of $p_{\text{c}} = 0.94$ and a combination of the parents genes based on their fitness, or keeping the better parent otherwise.

The *mutation* operator applies for each gene with a probability of

$$p_{\text{m}} = \frac{1}{|\mathbf{C}|}, \tag{4.25}$$

reversing the binary value of the affected genes. Additionally, the genetic algorithm was set to guarantee population *diversity*, i.e., substituted duplicate individuals with random ones and to keep the best individual through *elitism*.

Chapter 5

Modular-topology optimization of structures and mechanisms with free material design and clustering

Abstract: *In the optimal design of modular structures and mechanisms, two key questions must be answered: (i) what should the topology of individual modules be like and (ii) how should modules be arranged at the product scale? We address these challenges by proposing a bi-level sequential strategy that combines free material design, clustering techniques, and topology optimization. First, using free material optimization enhanced with post-processing for checkerboard suppression, we determine the distribution of elasticity tensors at the product scale. To extract the sought-after modular arrangement, we partition the obtained elasticity tensors with a novel deterministic clustering algorithm and interpret its outputs within Wang tiling formalism. Finally, we design interiors of individual modules by solving a single-scale topology optimization problem with the design space reduced by modular mapping, conveniently starting from an initial guess provided by free material optimization. We illustrate these developments with three benchmarks, covering compliance minimization of modular structures, and, for the first time, the design of compliant modular mechanisms. Furthermore, we design a set of modules reusable in an inverter and in gripper mechanisms, which ultimately pave the way towards the rational design of modular architectured (meta)materials.*

5.1 Introduction

Modular architectured materials, (self-)assembled [93] from a number of patterned components called modules, boost structural sustainability with possibly (re)configurable, multi-functional structures and mechanisms [221, 90, 141, 189, 213] while enabling efficient mass manufacturing through prefabrication, e.g., [63].

Optimal design of these modular materials is, however, still in its infancy. In principle, the design problem involves solving a bilevel *modular-topology* optimization problem incorporating (i) the design of the module topologies, and (ii) the spatial arrangement of the modules, both of which drive the resulting performance and versatility of the final structure. While the lower-level problem alone introduces only a slight modification to standard topology optimization (TO) procedures [192, 189, 212], the upper-level assembly plan design adds another combinatorial dimension to the structure of the problem [192]. For the case of continuum topology optimization, the module design problem lacks convexity and appears to be strongly dependent on initial guesses, which makes a rigorous assessment of assembly plan performance challenging.

Following a brief introduction to microstructural optimization in Section 5.1.1, we focus on modular designs and divide the contemporary methods for solving the modular-topology optimization problem into three groups: multi-material-based and meta-heuristic methods (both covered in Section 5.1.2), and clustering-based procedures (Section 5.1.3). In addition, any of these paradigms enable either a *concurrent* design scheme, in which the two levels are solved simultaneously and converged designs are (locally) optimal with respect to both the assembly plan and the module topologies, or a *sequential* scheme, in which the assembly plan design precedes the design of modules.

In this contribution we develop a novel sequential clustering method for designing modular structures and mechanisms based on Wang tiling formalism, Section 5.1.4. Main challenges of the methods and our contributions are summarized in Section 5.1.5.

5.1.1 Microstructural optimization

Microstructural topology optimization has been a very vivid research area [212], particularly due to the link to numerical homogenization [30]. However, conversely to homogenization, which evaluates effective properties of periodic media, the *inverse* homogenization approach proposed by Sigmund [175] has established a new framework for designing periodic materials with prescribed constitutive properties. The local periodicity requirement for homogenization-based methods relies on an infinite scale separation in order to maintain material connectedness of non-uniform microstructures and theoretical, superior performance. Because non-uniform distributions are optimal in general [163, 192], but since only finite-scale separations can be practically considered, the disconnectedness issue of neighboring microstructural cells has challenged researchers for years [170, 71].

Except for periodic-unit-cell (PUC) designs, which are compatible by definition but usually provide poor structural performance [128, 188, 213], several conceptual ideas have been proposed to resolve this problem for practical finite-scale separations during the last decade: prescribing connectors of neighboring, often parametric microstructures [227], design post-processing [75, 7, 170], and maintaining natural *mechanical* connectivity via single-scale optimization of the whole domain [192, 189] or subdomains [71].

Among these methods, the de-homogenization approach [152, 7, 77, 72] has proven to be a superior tool for single-load-case compliance optimization problems. With it, optimal microstructures are known to be rank-2 or rank-3 laminates whose stiffness is aligned with principal stress directions. Consequently, the optimal structures are found by projecting and smoothing optimal parametric microstructures. In more general cases such as multiple load cases, modular structures, or compliant mechanism designs, the topology of optimal microstructures is not known, and hence the method cannot be applied.

5.1.2 The optimum design of modular materials

Unlike the theoretically-optimal non-uniform distributions of microstructures, modularity limits itself to a finite small number of pattern types. As a consequence, it only approximates the optimal distribution, unless the optimal distribution is periodic. There are two extreme cases of modular designs: PUC microstructures, which repeat a single module in the design domain; and the optimal, infinite number of module types. As a result, modularity drastically reduces the number of design variables compared to the theoretically-optimal solution, but also provides a better structural response than PUC designs [192]. Furthermore, modularity may also improve the susceptibility of structures to local damage or imperfections [213].

As we have already outlined, the main challenge in the design of modular structures and mechanisms remains a coupled design of module topologies and their connectable spatial arrangements. Hence, early methods for modular design relied on a predefined distribution of modules [179, 61]. Although these designs performed better than PUCs in general, they were unable to handle the macro-scale module-assembly design problem, which substantially influenced their performance.

Several of the early methods for modular-topology optimization relied on meta-heuristics coupled with numerical optimization. Among these, Tugilimana *et al.* [188] optimized the spatially-varying rotations of a truss module, and, ultimately, both rotations and placements of multiple module types [189]. In the latter work, a simulated annealing was used for selecting an appropriate assembly plan, and a non-convex mathematical program for designing module topologies. Another approach was developed by Tyburec *et al.* [192], who optimized truss modules and their assemblies in the form of Wang tilings. In this method, a genetic algorithm was combined with convex mathematical programming, which provided a reliable way of assessing the performance of individual assembly plans.

Another class of approaches to modular design relies on concurrent multi-material-based methods. Among these, Zhang *et al.* [227] developed an approach based on ordered SIMP interpolation. Its main idea encompassed a repeated solution to two subproblems: solving a multi-material distribution problem for fixed material properties, and finding the material properties for a given material distribution. Notice that the former subproblem is not considered in the above meta-heuristic approaches. A similar setting to [227] can also be found in [132], where a rotation of a single parameterized unit cell was optimized for compliant mechanism problems.

5.1.3 Clustering-based methods

The third class of methods relies on clustering algorithms. Because these methods are particularly relevant to us, we review relevant developments in a dedicated section. For the

readers' convenience, we categorize the methods according to the quantities that are clustered: densities, the stress or strain tensors, and material stiffness tensors.

Density-based clustering. Density-based clustering determines candidate modular arrangements as clustered pseudo-densities of the variable thickness sheet problem. Since these densities usually interpolate isotropic materials, the isotropic homogenized properties of the modules are implicitly assumed. Due to the continuity of the density field, an infinite number of isotropic microstructures is needed in order to describe a general macro-domain. To effectively limit this number, Li *et al.* [126] and Gao *et al.* [69, 70] partitioned their ranges based on predefined thresholds. Each of these clustered densities was subsequently associated with a unique microstructure found by an inverse homogenization based on the level set method.

A related concept was adopted in Yan *et al.*'s [218] concurrent method, in which the strain energy of thermo-elastic lattice structures was minimized and the discreteness of the macro-scale structure assembly was secured by the `kmeans` algorithm which clustered similar strain energies of the cells without the need to specify fixed thresholds a priori.

Stress- and strain-based clustering. Because the idea of density-based clustering relies on the assumption of isotropic microstructures, it is usually too restrictive, since it reduces all microstructural information into a scalar field. A more general viewpoint incorporates the directional information, i.e., the data provided by the stress and strain tensors. However, even this perspective lacks full generality: it assumes designing each microstructure based on the principal directions only, i.e., rank-3 laminates in 3D. Such settings thus cannot handle cases with different optimum microstructures properly. These cases include multiple loading scenarios in minimum-compliance optimization or compliant mechanism design.

In one of the earliest works leading towards stress-based clustering, Xu and Cheng [215] investigated the concurrent setting of minimum-compliant multiscale topology optimization. In their work, the requirement of multiple unique microstructures was enforced by using so-called misplaced material volume constraints, which penalized the assignments of microstructures out of their assumed range of principal stress directions. Individual microstructures were designed by the inverse homogenization approach, but lacked mutual connectivity. Another recent concurrent method from Qiu *et al.* [157] proposed using dynamic clustering to cluster vectors containing the direction of the principal stresses as well as their ratios. The clustering task was solved by the `kmeans` algorithm. To ensure similar local optima of the clustering algorithm among the iterations of topology optimization, the cluster centers of `kmeans` were initialized based on the previous run.

In contrast to density-based clustering, stress- or strain-based clustering usually operates without microstructural volume fraction information. This drawback has been circumvented by the combined clustering proposed by Kumar and Suresh [116], who clustered the vectors containing densities and the ratios of principal strains.

Material stiffness clustering. The most structurally-efficient designs can only be found when exploiting all available information. For microstructural design, this information follows from the fourth-order material stiffness tensor. Considering isotropic materials only, Yang and Li [220] optimized the Poisson ratios of isotropic materials by introducing free isotropic material optimization. Their work combined a hierarchical clustering algorithm with a second-order conic formulation in the transformed polyhedral P-Q space, optimizing the

structural complementary strain energy. A natural generalization, developed by Hu *et al.* [88], explored the entire design freedom of positive definite material tensors by free material optimization (FMO), e.g. [231]. The optimal spatially-varying material tensors were clustered by a hierarchical clustering and individual micro-structures were obtained using the inverse homogenization method with embedded physics-based connectivity control.

5.1.4 Wang tilings

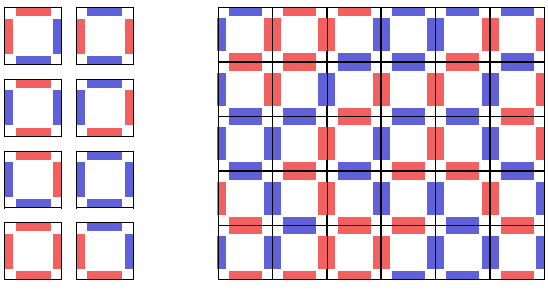
In this work, we adopt the concept of Wang tiles—a natural generalization of the PUC—as a convenient formalism for describing modular assemblies.

Visualized as unit squares with colored edges and a fixed orientation, the concept of Wang tiles was developed by and is named in honor of Hao Wang to simulate the $\forall\exists\forall$ decision problem of predicate calculus in mathematical logic. In an equivalent domino problem, Wang [201] assumed an arbitrary but fixed finite set of unitary tiles or dominoes and asked whether one can determine that their copies can cover an infinite plane while satisfying the domino rule of matching colors at shared edges of adjacent pieces; see Fig. 5.1 for an illustration.

An answer to the domino problem flourished in the work of Wang's student Berger [26], who proved that, in addition to tilesets that can cover the infinite plane by repeating periodic patterns, *aperiodic* tilesets also exist which satisfy the tiling objective yet forbid any periodic arrangement. Hence, the domino problem is not decidable. Since then, numerous aperiodic tilesets have emerged; see [79, Chapter 11] for a thorough introduction.

Berger's original proof relied on a reduction of the Turing machine halting problem into the domino problem. This reduction implies that Wang tiles are Turing-complete, and hence their (self)-assembly can simulate desired Turing computations. This property has motivated applications, e.g., at the DNA level [209, 208], and, recently, by Jílek *et al.* [93] on the centimeter-scale.

Visually appealing, nonperiodic yet compressed arrangements have impelled the adaptation of Wang tiles in computer graphics to generate naturally-looking textures [41, 226] and Poisson disk distributions [41, 86]. Sharing objectives, these applications have inspired the use of Wang tiles in modeling materials microstructures, both for geometrical representation [143, 56, 58] and calculations [144, 54, 57, 59]. Because the cardinality of the Wang tileset balances the computation efficiency of the PUC with precise (micro-)structural description, Wang tilings also appear to be useful in securing automatic connectivity in the topology optimization of modular truss structures [192].



(a) (b)

Figure 5.1: Illustration of Wang tiles: (a) a tileset of 8 tiles over two colors, and (b) a random 6×5 valid tiling satisfying the domino rule.

5.1.5 Aims and novelty

As follows from the literature survey, the design of modular architectured materials has only recently become an active research domain in the topology optimization field. Although several significant results have already appeared, there still remain fundamental questions to be answered.

Because the overall response of modular structures follows from two constituents, with the smooth macro-domain response driven by the effective constitutive parameters that are impaired by the presence of heterogeneities [59] in the form of module topologies, finding an appropriate assembly plan is the major challenge that, when successfully addresses, would allow attaining the desired structural effectiveness.

In this contribution, we rely on the clustering approach, recall Section 5.1.3, operating over material stiffnesses, and a standard single-scale topology optimization [22] extended to structural modularity. Each of our modules is defined as a quadruple of edge labels, thus defining a Wang tile, recall Fig. 5.1. In this dual point of view, we ask *how do we place at most m different interface types to make the resulting modular structure behave efficiently?* The answer is indeed in the form of a valid Wang tiling over at most m colors.

To make our setting fully general, our clustering dataset originates from a solution to an FMO problem, which we extend to handle both minimum-compliance and compliant mechanism design objectives, Section 5.2.1.1. To the best of our knowledge, the procedures for designing modular assembly plans in linear elasticity problems have mostly been limited to minimum-compliance optimization. The only exception we are aware of is [132], optimizing rotations of a single unit cell for compliant mechanisms via multi-material topology optimization. However, a more general setting has already been requested [213].

Surprisingly, as we will show in Section 5.2.1.2, the optimal elastic properties obtained from the FMO problem often contain checkerboard patterns. Nevertheless, this artifact has not yet been investigated with connection to the FMO problems although it has appeared in published results, e.g., [88]. Since we only utilize the elastic properties for a heuristic guidance, we adopt a simple post-processing step and postpone the rigorous treatment of the checkerboard patterns in FMO to future works.

The objective of the clustering algorithms is to partition a given dataset of points into a specific number of clusters. However, standard clustering algorithms have several disadvantages: they are usually stochastic and, therefore, require multiple evaluations to generate quality outputs. Moreover, they also generally fail to maintain structural symmetries. To overcome these shortcomings, we have developed a novel greedy agglomerative clustering algorithm that is deterministic while preserving structural symmetry of the elasticity field, when present (Section 5.2.1.3). Using this clustering algorithm on a module interface-based discretization produces a valid Wang tiling.

For appropriate assembly plans obtained via the clustering procedure, the module topologies must also be optimized. Unfortunately the emergent optimization problems appear to be strongly dependent on the provided starting point. To make the initial choice problem-specific yet well-defined, we propose to use a solution to a macro-scale FMO problem with modularity to construct initial guesses based on the traces of the optimal stiffness tensors obtained from FMO (Section 5.2.2.1). A single-scale topology optimization extended to structural modularity is finally used for a computation of the optimized modular topologies in Section 5.2.2.2.

We illustrate the developed theory on four selected examples, including minimum-compliance optimization of the MBB beam problem, Section 5.3.1, and three compliant mechanisms:

inverter (Section 5.3.2), gripper (Section 5.3.3), and a module set reusable among both the inverter and gripper mechanisms (Section 5.3.4). We conclude that our approach to designing modular structures provides fairly efficient and well-behaving designs.

In this contribution, we adopt the following notation: scalar variables are written in plain, lower-case italic letters (a). First- and second-order tensors appear in lower-case, bold italic ($\boldsymbol{a}$) and bold regular ($\mathbf{a}$) letters, respectively, and the fourth-order tensors are denoted as bold capital regular letters ($\mathbf{A}$). Finally, column vectors and matrices appear in sans-serif lower and upper-case bold characters (a, A).

5.2 Methods

This section is devoted to establishing a sequential method that provides a heuristic solution to the modular-topology optimization problem with the modular assembly plan parameterized as valid Wang tilings, recall Section 5.1.4. Our approach to designing modular structures and mechanisms encompasses six sequential steps: (i) optimum free material design performed on a module interface-based mesh, Section 5.2.1.1, (ii) removal of potential modules whose interfaces lack any significant stiffness, (iii) smoothing of the material stiffness field to suppress the checkerboard patterns, Section 5.2.1.2, and (iv) clustering of elasticity matrices by a newly-developed agglomerative clustering algorithm to output valid Wang tilings, Section 5.2.1.3. For the resulting assembly plan, we (v) optimize the material properties of individual modules via an FMO extended to modularity, Section 5.2.2.1, which provides an appropriate initial guess for (vi) the standard continuum topology optimization extended to modularity in Section 5.2.2.2.

5.2.1 Assembly plan design

Based on the inherent bilevel nature of the modular-topology optimization problem, we split the above six steps into two parts—a heuristic procedure for assembly plan design, and then topological optimization of the modules. In this section, we aim to answer the question of *how should we design efficient assembly plans*? To this goal, we introduce free material optimization in Section 5.2.1.1, the necessary post-processing steps in Section 5.2.1.2, and a novel clustering procedure in Section 5.2.1.3.

5.2.1.1 Free material optimization

Let us consider a two-dimensional continuum body Ω under appropriate static and kinematic boundary conditions. In this domain, the free material optimization (FMO) method seeks the spatially-varying material elasticity tensors $\mathbf{E}$ that make the structural response optimal with respect to a specified performance functional $a(\boldsymbol{u})$ [231] of a displacement field $\boldsymbol{u}$.

In this manuscript, we assume a linear strain-displacement relation

$$\varepsilon_{ij}(\boldsymbol{x}) = \frac{1}{2}\left(\frac{\partial u_i(\boldsymbol{x})}{\partial x_j} + \frac{\partial u_j(\boldsymbol{x})}{\partial x_i}\right), \quad \forall \boldsymbol{x} \in \Omega\,, \tag{5.1}$$

where ε denotes a strain tensor. The sought elastic material properties are represented by the effective elasticity tensor $\mathbf{E}$ that satisfies linear Hooke's law written in indicial notation as

$$\sigma_{ij}(\boldsymbol{x}) = E_{ijkl}(\boldsymbol{x})\,\varepsilon_{kl}(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \Omega\,, \tag{5.2}$$

in which σ is a stress tensor.

In what follows, we adopt the (engineering) Voigt notation that allows us to rewrite the tensors ε, σ, and $\mathbf{E}$ as column and square matrices, respectively. To this goal, let

$$\boldsymbol{\varepsilon}(\boldsymbol{x}) = \begin{pmatrix} \varepsilon_{11}(\boldsymbol{x}) & \varepsilon_{22}(\boldsymbol{x}) & 2\varepsilon_{12}(\boldsymbol{x}) \end{pmatrix}^{\mathrm{T}} = \begin{pmatrix} \varepsilon_{11}(\boldsymbol{x}) & \varepsilon_{22}(\boldsymbol{x}) & \gamma_{12}(\boldsymbol{x}) \end{pmatrix}^{\mathrm{T}} \tag{5.3}$$

be the strain column matrix, with γ_{12} being the engineering shear strain, and let

$$\sigma(\boldsymbol{x}) = \begin{pmatrix} \sigma_{11}(\boldsymbol{x}) & \sigma_{22}(\boldsymbol{x}) & \sigma_{12}(\boldsymbol{x}) \end{pmatrix}^{\mathrm{T}} = \begin{pmatrix} \sigma_{11}(\boldsymbol{x}) & \sigma_{22}(\boldsymbol{x}) & \tau_{12}(\boldsymbol{x}) \end{pmatrix}^{\mathrm{T}} \tag{5.4}$$

denote the stress matrix. Then, the matrix form of the Hooke law (5.2) reads as

$$\boldsymbol{\sigma}(\boldsymbol{x}) = \mathsf{E}(\boldsymbol{x})\,\boldsymbol{\varepsilon}(\boldsymbol{x})\,, \quad \forall \boldsymbol{x} \in \Omega\,, \tag{5.5}$$

with the material stiffness matrix

$$\mathsf{E}(\boldsymbol{x}) = \begin{pmatrix} E_{1111}(\boldsymbol{x}) & E_{1122}(\boldsymbol{x}) & E_{1112}(\boldsymbol{x}) \\ E_{1122}(\boldsymbol{x}) & E_{2222}(\boldsymbol{x}) & E_{2212}(\boldsymbol{x}) \\ E_{1112}(\boldsymbol{x}) & E_{2212}(\boldsymbol{x}) & E_{1212}(\boldsymbol{x}) \end{pmatrix} \tag{5.6}$$

constructed from the (to-be-optimized) components E_{ijkl} of the forth-order elasticity tensor $\mathbf{E}$. To simplify the further developments, we introduce a vector form of E as

$$\mathsf{e}(\boldsymbol{x}) = \begin{pmatrix} E_{1111}(\boldsymbol{x}) & \sqrt{2}E_{1122}(\boldsymbol{x}) & E_{2222}(\boldsymbol{x}) & \sqrt{2}E_{1112}(\boldsymbol{x}) & \sqrt{2}E_{2212}(\boldsymbol{x}) & E_{1212}(\boldsymbol{x}) \end{pmatrix}^{\mathrm{T}} \tag{5.7}$$

such that

$$\|\mathsf{E}(\boldsymbol{x})\|_{\mathrm{F}} = \|\mathsf{e}(\boldsymbol{x})\|_2\,, \tag{5.8}$$

where $\|\bullet\|_2$ denotes the standard Euclidean norm of the vector $\bullet$, and $\|\bullet\|_{\mathrm{F}}$ is the Frobenius norm.

Our approach to a numerical solution of the FMO optimization problems relies on approximate solutions to the governing equation obtained with the finite element method: we search for the kinematically admissible displacement field $\mathsf{u} \in \mathbb{R}^{n_{\mathrm{dof}}}$ satisfying the equilibrium equations

$$\mathsf{K}\left(\mathsf{E}^{(1)}, \ldots, \mathsf{E}^{(n_\ell)}\right)\mathsf{u} = \mathsf{f}, \tag{5.9}$$

where $\mathsf{f} \in \mathbb{R}^{n_{\mathrm{dof}}}$ denotes the column matrix of equivalent nodal forces, n_{dof} stands for the number of degrees of freedom, the super-script $\bullet^{(\ell)}$ expresses an assignment of $\bullet$ to the element ℓ, and $\mathsf{K}\left(\mathsf{E}^{(1)}, \ldots, \mathsf{E}^{(n_\ell)}\right) \in \mathbb{S}^{n_{\mathrm{dof}}}_{\succ 0}$ is the symmetric, positive definite structural stiffness matrix—a function of the discretized field of elasticity matrices $\mathsf{E}^{(\ell)} \in \mathbb{S}^3_{\succ 0}$ of n_ℓ finite elements. In this notation, $\mathbb{S}^\bullet$ refers to a space symmetric square matrices of the size $\bullet \times \bullet$, and $\succ$ denotes positive definiteness. In addition, we define $\mathsf{X} \succeq \epsilon$ to express that the eigenvalues of X are grater than or equal to ϵ.

The structural stiffness matrix follows from an assembly of element stiffness matrices $\mathsf{K}^{(\ell)}$ based on the element gather matrices $\mathsf{L}^{(\ell)}$ and a design-independent term $\mathsf{K}_0 \in \mathbb{S}^{n_{\mathrm{dof}}}_{\succeq 0}$,

$$\mathsf{K}\left(\mathsf{E}^{(1)}, \ldots, \mathsf{E}^{(n_\ell)}\right) = \mathsf{K}_0 + \sum_{\ell=1}^{n_\ell} \mathsf{L}^{(\ell)\mathrm{T}}\mathsf{K}^{(\ell)}(\mathsf{E}^{(\ell)})\mathsf{L}^{(\ell)}, \tag{5.10}$$

where the element contributions $\mathsf{K}^{(\ell)}\left(\mathsf{E}^{(\ell)}\right) \in \mathbb{S}_{\succeq 0}$ evaluate as

$$\mathsf{K}^{(\ell)}\left(\mathsf{E}^{(\ell)}\right) = \int_{\Omega_\ell} \mathsf{B}^{(\ell)\mathrm{T}}(\boldsymbol{x})\, \mathsf{E}^{(\ell)}(\boldsymbol{x})\, \mathsf{B}^{(\ell)}(\boldsymbol{x}) \mathrm{d}\Omega. \tag{5.11}$$

In (5.11), $\mathsf{B}^{(\ell)}$ denotes the strain-displacement matrix arising from the adopted displacement approximation, and Ω_ℓ constitutes the element domain. From now on, we assume $\mathsf{E}^{(\ell)}$ to be constant over each element, so that $\mathsf{K}^{(\ell)}$ is a linear function of $\mathsf{E}^{(\ell)}$.

In the spirit of standard topology optimization (TO) [22], FMO seeks the most efficient distribution of a limited amount of material, constrained by Eqs. (5.12c)–(5.12e), that satisfies the (elastic) equilibrium, Eq. (5.12b). However, compared to TO, the FMO setting opens up a broader design freedom, because the full material stiffness matrix is the subject of optimization. Here, we formalize the FMO problems of our interest as

$$\min_{s,\mathsf{E}^{(1)},\dots,\mathsf{E}^{(n_\ell)}} \quad s \tag{5.12a}$$

$$\text{s.t. } s - \mathsf{a}^{\mathrm{T}} \left[\mathsf{K}\left(\mathsf{E}^{(1)},\dots,\mathsf{E}^{(n_\ell)}\right)\right]^{-1} \mathsf{f} \geq 0, \tag{5.12b}$$

$$\overline{V}\,\overline{\epsilon} \sum_{\ell=1}^{n_\ell} v^{(\ell)} - \sum_{\ell=1}^{n_\ell} \left[v^{(\ell)} \mathrm{Tr}\left(\mathsf{E}^{(\ell)}\right)\right] \geq 0, \tag{5.12c}$$

$$\overline{\epsilon} - \mathrm{Tr}\left(\mathsf{E}^{(\ell)}\right) \geq 0, \;\; \forall \ell \in \{1,\dots,n_\ell\}, \tag{5.12d}$$

$$\mathsf{E}^{(\ell)} \succeq \underline{\epsilon}, \;\; \forall \ell \in \{1,\dots,n_\ell\}. \tag{5.12e}$$

In (5.12), $s \in \mathbb{R}$ is a slack variable and $\mathsf{a} \in \mathbb{R}^{n_{\mathrm{dof}}}$ stands for a fixed vector representing the discretized performance of the functional a. Furthermore, $\underline{\epsilon} \in \mathbb{R}_{>0}$ and $\overline{\epsilon} \in \mathbb{R}_{>0}$ are lower- and upper-bounds on the (sum of the) eigenvalues of $\mathsf{E}^{(\ell)}$, $\forall \ell \in \{1,\dots,n_\ell\}$, $\overline{V} \in \langle 0,1\rangle$ stands for the volume fraction, $v^{(\ell)} \in \mathbb{R}_{>0}$ represents the volume of the ℓ-th element, and Tr is the trace operator, amounting to the sum of its argument's eigenvalues. Notice that in contrast to established formulations [231, 110, 88], we use an element volume-based weighted average in (5.12c) to facilitate optimization with non-structured spatial discretizations.

When setting $\mathsf{a} = \mathsf{f}$, the slack variable s becomes an upper bound on the compliance functional, (5.12b), rendering (5.12) to be a compliance minimization problem. Such optimization problem is convex: the functions in (5.12a) and (5.12c)–(5.12e) are linear (matrix) functions, and (5.12b) possesses a positive definite Hessian, see Appendix 5.B.2, and it is therefore also convex. For the case of $\mathsf{a} \neq \mathsf{f}$, the second term in function (5.12b) amounts to $\mathsf{a}^{\mathrm{T}}\mathsf{u}$. Consequently, formulation (5.12) facilitates optimization of compliant mechanisms by minimizing weighted sums of the displacement vector entries.

Considering Eqs. (5.12c) and (5.12e), a natural question arises of how to choose $\underline{\epsilon}$ and $\overline{\epsilon}$. Clearly, the lower bound for eigenvalues must be sufficiently small to mimic void materials but should be large enough to avoid problems with a possibly non-existing inverse in (5.12b). In our case, we fix $\underline{\epsilon} = 10^{-3}\overline{\epsilon}$. To determine the upper-bound on the sum of eigenvalues, $\overline{\epsilon}$, we link the FMO problem (5.12) with our ultimate goal of designing microstructures made of a mixture of an isotropic material and void, optimized with standard continuum topology optimization. Assuming a family of isotropic materials under plane stress that are parameterized by the Young modulus $E^{(\ell)} \in \langle 0,1\rangle$ and a fixed Poisson ratio ν, the trace of the isotropic material stiffness matrix $\mathsf{E}^{(\ell)}_{\mathrm{iso}}$ amounts to

$$\mathrm{Tr}(\mathsf{E}^{(\ell)}_{\mathrm{iso}}) = \frac{5-\nu}{2(1-\nu^2)} E^{(\ell)}. \tag{5.13}$$

If the entire design domain is filled with the solid material, $\overline{V} = 1$, the trace constraint (5.12c) implies that

$$\overline{\epsilon} = \frac{5 - \nu}{2(1 - \nu^2)}. \tag{5.14}$$

With the theoretical bounds on the Poisson ratio $\nu \in (-1, 0.5\rangle$, Eq. (5.14) implies that for all solid isotropic materials under plane stress $\overline{\epsilon} \in \langle \frac{5}{4} + \frac{\sqrt{6}}{2}, \infty)$.

The optimization problem (5.12) can readily be solved by nonlinear semidefinite programming solvers such as PENLAB [62] or PENNON [109]. However, for the special case of $\mathsf{a} = \mathsf{f}$, one can also adopt any of the more widely available (convex) linear semidefinite programming optimizers. In this contribution, we rely on the PENNON solver kindly provided by its authors. For compliance minimization problems, we use both the first- and second-order derivatives, see Appendix 5.B. For compliant mechanism problems, it appeared to be more efficient to use an identity matrix instead of the exact Hessian term from Eq. (5.48). In addition, practical computations revealed that it is more beneficial to solve a scaled version of (5.12), we refer the reader to Appendix 5.C for details.

5.2.1.2 Checkerboard pattern

The FMO generates (locally-)optimal elasticity matrices $\mathsf{E}^{(\ell)}$ that are, however, prone to the checkerboard pattern phenomenon [52], especially in the case of the bilinear quadrilateral finite elements, which are the most widely used elements in the TO community [9]. In Fig. 5.2, we show this emerging issue on traces of the optimal material stiffnesses in compliance optimization of a bar under uniaxial compression. Please note that while this issue is apparent at the element stiffness traces in this case, it may be difficult to recognize in general because FMO operates directly over stiffness components and, hence, it has no scalar unknown field as in standard density-based TO, and the checkerboard may be lost as a result of visualization.

There are probably multiple reasons why the checkerboard issue occurs. First, we optimize over element design variables, whereas the finite element system (5.9) operates over a nodal-based displacement field. Therefore, the output of (5.12) shall be interpreted as optimal

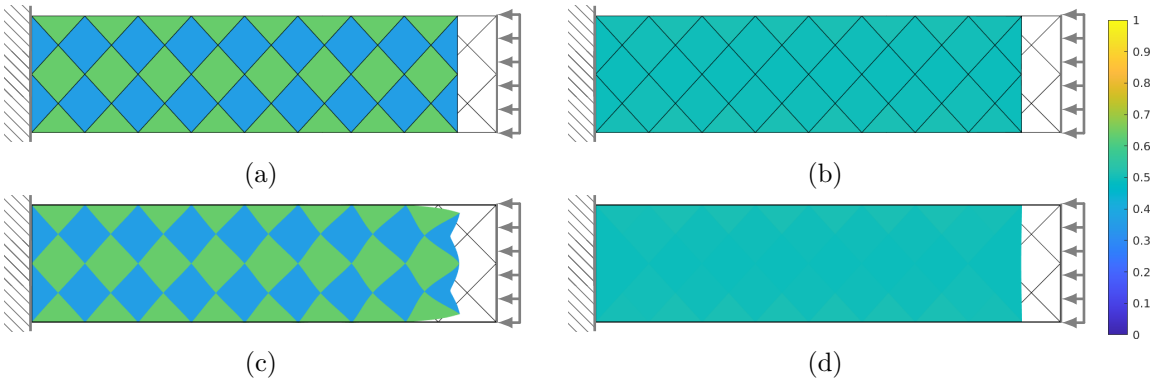


Figure 5.2: A bar of length 4 and height 1 under uniaxial compression, designed for $\overline{V} = 0.5$ and $\overline{\epsilon} = 1$ with visualized traces of elasticity matrices and deformed shape. (a) Optimum design obtained by solving a FMO problem ($s_{\mathrm{FMO}} = 32.13$) and (c) its projection on a finer mesh ($s_{\mathrm{FMO}}^{\mathrm{refined}} = 34.33$), and design after post-processing (b) on the optimization mesh ($s_{\mathrm{FMO,f}} = 32.13$) and (d) projected onto a refined mesh ($s_{\mathrm{FMO,f}}^{\mathrm{refined}} = 32.13$).

discretized (nodal) stiffnesses in the form of the stiffness matrix $\mathsf{K}\left(\mathsf{E}^{(1)},\dots,\mathsf{E}^{(n_\ell)}\right)$, which seems to be checkerboard-free. However, discretized nodal stiffnesses result from a (weighted) sum of the stiffnesses of connected elements, recall (5.10). Clearly, the element stiffnesses do not play any role unless they change a value in their assembled sum $\mathsf{K}\left(\mathsf{E}^{(1)},\dots,\mathsf{E}^{(n_\ell)}\right)$. Hence, the element stiffnesses may indeed form a checkerboard pattern, which is often falsely recognized in finite element analyses as structurally more efficient [52]. We note here that associating the design variables with the nodes does not resolve the problem, because the element stiffnesses (5.11) are usually evaluated at the Gauss points inside the finite element interior, not at the nodal points. Furthermore, the bilinear quadrilateral elements used in this manuscript are known to be prone to locking, and, therefore, they tend to overestimate the shear effects [155, 230]. Again, this may result in a load transfer that is nonphysical.

To our knowledge, the checkerboard issue has not been investigated with connection to the FMO problem (5.12) yet, probably because only regular orthogonal mesh discretizations have been adopted, and, in that case, the issue is usually not visually significant but indeed occurs, see e.g., [88, Figs. 4b and 4e]. For another discretizations, such as the rotated one used in this manuscript, Fig. 5.2, the checkerboard is more apparent.

Traditionally, two approaches have been adopted in the density-based topology optimization to mitigate the checkerboard patterns: the sensitivity and density filters [22]. These filters suppress the checkerboard pattern by a convolution of the first-order derivatives, or of the pseudo-density field. They can also be applied to the FMO problem directly by filtering the entries of the elasticity matrices independently because the space of positive semi-definite matrices is convex. However, we adopt a different, simpler approach. Because we use FMO as a heuristic tool only, we perform a single post-processing step. In addition to being computationally more efficient, our post-processing maintains near-optimality in the structural performance for sufficiently fine discretizations.

With n_n denoting the number of nodes, let $\mathsf{F}\in\mathbb{R}_{\geq 0}^{n_\ell\times n_\mathrm{n}}$ be a *filtering* matrix, whose ℓ-th row is associated with the element ℓ and j-th column belongs to the node j. Components of F follow from

$$F_{\ell j}=\int_{\Omega_\ell} N_j(\boldsymbol{x})\mathrm{d}\Omega\,, \tag{5.15}$$

where N_j is the shape function associated with the j-th node. Operating coefficient-wise, we define a vector $\mathsf{e}_k=\left(e_k^{(1)},\dots,e_k^{(n_\ell)}\right)^\mathrm{T}$ collecting k-th component across all elements ℓ and perform post-processing according to

$$\widehat{\mathsf{e}}_k=\left[\mathrm{diag}\left(\mathsf{F}\mathsf{1}\right)^{-1}\mathsf{F}\right]\left[\mathrm{diag}\left(\mathsf{F}^\mathrm{T}\mathsf{1}\right)^{-1}\mathsf{F}^\mathrm{T}\right]\mathsf{e}_k, \tag{5.16}$$

with $\mathsf{1}$ being an all-one column vector of appropriate dimensions and the diag() operator constructing a square matrix with its argument vector placed at the major diagonal. The procedure in (5.16) consists of two projections: interpolating element material stiffnesses onto nodal material stiffnesses by $\left[\mathrm{diag}\left(\mathsf{F}^\mathrm{T}\mathsf{1}\right)^{-1}\mathsf{F}^\mathrm{T}\right]$, and a backwards averaging of nodal material stiffnesses onto the element ones, $\left[\mathrm{diag}\left(\mathsf{F}\mathsf{1}\right)^{-1}\mathsf{F}\right]$. Because both these projections are convex, such post-processing preserves the trace value in the resource constraint (5.12c) and the convex eigenvalue constraints (5.12e) and (5.12d).

Nevertheless, the post-processing step may impair the quality of the objective function value (5.12a). Our numerical experiments with projecting the originally element-wise constant

material properties onto a refined mesh, e.g. 800×200 elements in Fig. 5.2c, leads to significant discrepancy in the predicted objective when the post-processing is not used. On the other hand, with post-processed densities being projected, the detoriation of the performance is less pronounced, and more importantly the objective is lower compared to a refined design based on the non-filtered design. In addition, the increase caused by the post-processing is mostly aligned with the fineness of the finite element discretization. Because optimal material designs are locally periodic in the limit [20], we have $\widehat{\mathsf{e}}_k \to \mathsf{e}_k$. In fact, a zero increase occurs for any periodic problem, even with a coarse discretization, recall Fig. 5.2. Therefore, there is a zero increase in the objective function value for infinitely-fine discretizations.

5.2.1.3 Clustering for modular Wang tile design

Having smoothed the material stiffness field, our next goal is to determine structurally-efficient modular assemblies in the form of valid Wang tilings. In this work, we follow the direction of the clustering of the material stiffnesses because it is the most general option: it generalizes both the density clustering for isotropic materials with a fixed Poisson ratio and the strain-based clustering for problems possessing optimal design in the form of rank-3 laminates. Our approach thus extends to the remaining linear elasticity problems including multi-load designs and compliant mechanisms. To this goal, this section develops a greedy agglomerative clustering algorithm that maintains structural symmetries when present.

As discussed already in Section 5.1.4, valid Wang tilings are determined uniquely by the edge color codes. Therefore, instead of the modules, Fig. 5.3a, we must cluster material stiffnesses corresponding to the horizontal and vertical edges. These stiffnesses are obtained by solving the free material optimization problem (5.12) on a rotated, module interface-based finite element mesh, see Fig. 5.3b. When comparing with Fig. 5.3a, it clearly follows that each of the modules is defined by a quadruple of (edge-related) elements in Fig. 5.3b. Hence, after clustering, the edge labels automatically define Wang tiles and a valid tiling, see Figs. 5.3g–h. We note here that a similar approach relying on rotated mesh has been applied to generate Wang tile-based textures in computer graphics [41] and microstructure modeling [56].

Using the post-processed FMO outputs, it may happen that there would exist modules without any stiffness in their edges, i.e., the maximum eigenvalue of the edge-related stiffness tensors is small. Such modules can conversely be interpreted as voids, and we remove them prior to the clustering procedure and leave their locations empty.

The common aim of the clustering algorithms is to partition a set of points $\mathcal{P} := \left\{\mathsf{e}^{(1)}, \ldots, \mathsf{e}^{(n_\ell)}\right\}$ into (at most) m groups, such that a specified distance-based criterion is minimized [68]. Among these algorithms, `kmeans` is probably the most widely used. It starts with a random assignment of points to clusters, computes a centroid of each cluster based on a specified distance function, and re-assigns each point to the cluster with the nearest centroid. This procedure repeats until convergence, but the resulting clustering strongly depends on the initial partitioning.

The second group of clustering algorithms, spectral clustering, exploits a representation of the points $\mathcal{P}$ in a graph form. For this representation, one computes the graph Laplacian matrix and finds a partitioning based on its eigenvectors. Based on our testing (not included), neither `kmeans`, nor spectral clustering led to convincing designs.

In this manuscript, we rely on the third class of algorithms, the hierarchical clustering, because this class may avoid randomization and can be very simply tailored to satisfy additional

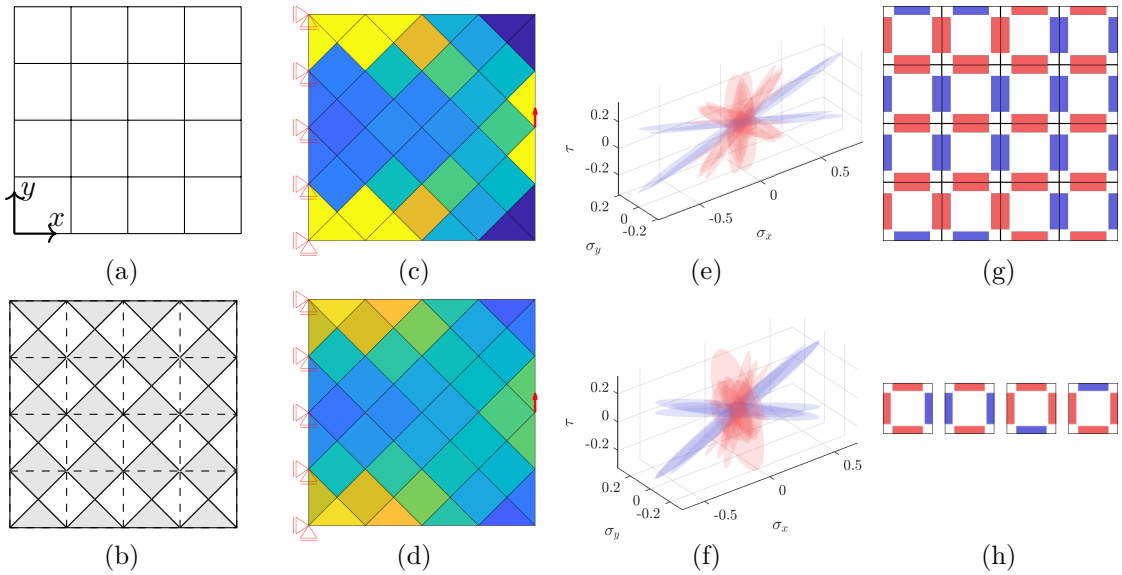


Figure 5.3: Illustration of the proposed heuristic approach to generate efficient valid Wang tilings. First, we (a) discretize a design domain into modules, and (b) associate a finite element with each edge: elements belonging to horizontal and vertical edges are shown in gray and white, respectively. For this mesh, we (c) solve the free material optimization problem (5.12), represented in this figure by the traces of material stiffnesses, and (d) post-process them. Finally, the material stiffnesses visualized in the (e) horizontal and (f) vertical edges as envelopes of the stress components σ_x, σ_y and τ induced by arbitrary unit strain vectors, are clustered independently based on the Frobenius norm, establishing a (g) Wang tiling assembly plan and the (h) module set.

constraints imposed on the clustering data. In particular, the latter feature appears to be very difficult to achieve with the two previous classes of algorithms. Hierarchical clustering algorithms can be categorized into *divisive* clustering—starting with all points belonging to a single cluster, partitioned recursively—and *agglomerative* clustering—where each point initially defines a cluster, and these are joined repetitively based on a linkage criterion.

In our implementation, we follow the agglomerative clustering approach and adopt the squared Frobenius norm to measure the distances of the material stiffness matrices, which is equivalent to the Euclidean norm among the stiffness vectors $\mathsf{e}^{(\ell)}$, recall Section 5.2.1.1. To compute a distance between clusters, we represent each cluster $\mathcal{C}_i$ by its centroid with components

$$\mathsf{t}_{i,j} = \frac{\sum_{\ell \in \mathcal{C}_i} w_\ell e_j^{(\ell)}}{|\mathcal{C}_i|}, \tag{5.17}$$

where $\mathsf{w} \in \mathbb{R}^{n_\ell}_{>0}$ are weights of the input data and $|\mathcal{C}_i|$ denotes the cardinality of points in $\mathcal{C}_i$, and by the cluster cost

$$c_i = \sum_{\ell \in \mathcal{C}_i} w_\ell \|\mathsf{t}_i - \mathsf{e}^{(\ell)}\|_2^2. \tag{5.18}$$

To determine which clusters shall be merged, we compute for all pairs of clusters (i, j) the deterioration cost

$$d_{i,j} = \begin{cases} c_{i,j} - c_i - c_j & \text{if } i \text{ and } j \text{ are allowed to merge,} \\ \infty & \text{otherwise,} \end{cases} \tag{5.19}$$

where $c_{i,j}$ stands for the cost of the cluster $\mathcal{C}_i \cup \mathcal{C}_j$. We accept the merger that least deteriorates the objective function. Notice that this linkage criterion is not equivalent to merging the nearest clusters, as more distant clusters containing fewer points may be cheaper to merge than the nearer ones with more data points. In this sense, we believe that this clustering procedure provides a more balanced number of points inside clusters.

Assume now that the material stiffness distribution exhibits symmetry with respect to a vertical or horizontal axis. For such a setting, we find it reasonable that the resulting clusterings preserve this symmetry pattern. In our implementation, this can be guaranteed by restricting which clusters cannot join, i.e., the deterioration cost (5.19) is set to ∞ for some non-admissible pairs (i, j).

To this goal, we split the material stiffnesses into three groups $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}'$ and $\boldsymbol{\beta}$. $\boldsymbol{\beta}$ contains material stiffnesses of the elements that are either located on the axis of symmetry or do not have a symmetric counterparts, and groups $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$ consists of the symmetric complements. More precisely, let us assume a fixed $i \in \{1, \dots, |\mathcal{C}|\}$ and $j \in \{1, \dots, |\mathcal{C}|\} \setminus i$ such that the entries of these centroids satisfy $t_{i,1} = t_{j,1}$, $t_{i,2} = t_{j,2}$, $t_{i,3} = t_{j,3}$, $t_{i,4} = -t_{j,4}$, $t_{i,5} = -t_{j,5}$, and $t_{i,6} = t_{j,6}$, which certifies symmetry of the corresponding material stiffness centroids with respect to the vertical or horizontal axis, cf. Appendix 5.A for details. If no such j exists for a fixed i or when $t_{i,4} = t_{i,5} = 0$, we set $\mathtt{t}_i \in \boldsymbol{\beta}$. If the corresponding pair exists, we set $\mathtt{t}_i \in \boldsymbol{\alpha}$ if $t_{i,4} > 0$ or $t_{i,4} = 0 \wedge t_{i,5} > 0$, and $\mathtt{t}_i \in \boldsymbol{\alpha}'$ otherwise.

Using these groups, a finite deterioration cost is associated only with (i) a merger of the corresponding pair $i \in \boldsymbol{\alpha}$ and $j \in \boldsymbol{\alpha}'$, producing a cluster belonging to $\boldsymbol{\beta}$, (ii) joining clusters $\{i, j : i, j \in \boldsymbol{\beta}, i \neq j\}$, with the result also belonging to $\boldsymbol{\beta}$, and (iii) a merger of $\{i, j : i, j \in \boldsymbol{\alpha}, i \neq j\}$, which requires to merge concurrently the pairs $\{i, j : i, j \in \boldsymbol{\alpha}', i \neq j\}$. In this last case, the number of clusters is decreased by two, with the merged outputs belonging to $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$, respectively. All the possible aggregations are illustrated in Fig. 5.4.

In each iteration, the aggregation procedure reduces $|\mathcal{C}|$ by 1 at least, building an aggregation tree called dendrogram. When $|\mathcal{C}| \leq m$, the clustering algorithm terminates, see Algorithm 5.1 for the pseudo-code. As an output, we obtain an assignment of the input data (horizontal or vertical edges) into clusters, Fig. 5.3e–f, and this partitioning is uniquely representable by a valid Wang tiling over (at most) m colors, Fig. 5.3g–h.

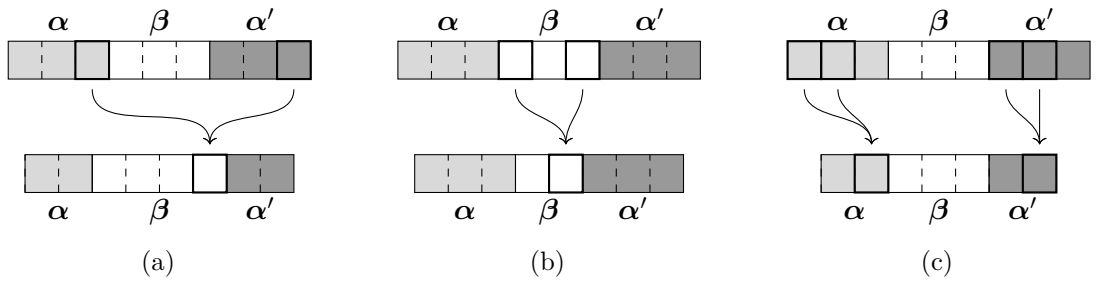


Figure 5.4: Illustration of all allowed cluster aggregations to generate Wang tilings preserving the symmetric distribution of material stiffness matrices when present. The set $\boldsymbol{\beta}$ consists of material stiffnesses of the elements located on the axis of symmetry and those that do not have a symmetric counterparts, and $\boldsymbol{\alpha}$ with $\boldsymbol{\alpha}'$ contain the symmetric complements.

Algorithm 5.1 Pseudo-code of the proposed greedy agglomerative clustering algorithm.

$\boldsymbol{\alpha}, \boldsymbol{\alpha}', \boldsymbol{\beta}, \mathcal{C} \leftarrow$ `initialize`(e)
while $|\mathcal{C}| > m$ **do**
  pair_1, $\text{cost}_1 \leftarrow$ `findCheapestMerge`($\boldsymbol{\alpha}$, $\boldsymbol{\alpha}'$)
  pair_2, $\text{cost}_2 \leftarrow$ `findCheapestMerge`($\boldsymbol{\beta}$, $\boldsymbol{\beta}$)
  pairs_3, $\text{cost}_3 \leftarrow$ `findCheapestMerge`($\boldsymbol{\alpha}$, $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}'$, $\boldsymbol{\alpha}'$)
  if $|\mathcal{C}| - m > 1$ **then**
    $\boldsymbol{\alpha}, \boldsymbol{\alpha}', \boldsymbol{\beta}, \mathcal{C} \leftarrow$ `mergeCheapest`(pair_1, cost_1, pair_2, cost_2, pairs_3, $\text{cost}_3/2$)
  else
    $\boldsymbol{\alpha}, \boldsymbol{\alpha}', \boldsymbol{\beta}, \mathcal{C} \leftarrow$ `mergeCheapest`(pair_1, cost_1, pair_2, cost_2, pairs_3, cost_3)
  end if
end while
return $\mathcal{C}$

5.2.2 Design of topologies of modules

Once an assembly plan—in the form of valid Wang tiling—is obtained, the dimension of design space in TO usually reduces drastically because $n_\mathrm{m} < n_\ell$, with n_m being the number of Wang tiles or modules and n_ℓ denoting the number of module positions in Fig. 5.3a. However, it seems that—at least for compliant mechanism design problems—the landscape of the objective function used in TO is very complex and optimized topologies depend strongly on a supplied initial guess. This issue becomes even more relevant when dealing with the design of modules reusable among multiple structures (Section 5.3.4), since an inappropriate initial guess may prevent the mechanisms to work at all.

In this section, we first recall a variable linking procedure [192] to describe modular structures mathematically, and apply it to the FMO problem (5.12), which allows for an inexpensive computation of the initial guesses for TO based on the element traces, see Section 5.2.2.1, and also a simple comparison of the efficiency of different assembly plans. Description of the adopted TO method is given in Section 5.2.2.2.

5.2.2.1 Modular free material optimization

Let n_m denote the number of modules and $\{\mathsf{g} \in \mathbb{N}^{n_\ell}, \forall \ell \in \{1,\dots,n_\ell\} : g_\ell \leq n_\mathrm{m}\}$ be a group vector assigning a module number $m \in \{1,\dots,n_\mathrm{m}\}$ to each element $\ell \in \{1,\dots,n_\ell\}$ of the module scale, recall Fig. 5.3a. In addition, let us define a group matrix $\mathsf{G} \in \mathbb{B}^{n_\ell \times n_\mathrm{m}}$ as

$$G_{\ell,j} = \begin{cases} 0 & \text{if } j \neq g_\ell, \\ 1 & \text{if } j = g_\ell, \end{cases} \quad \forall \ell \in \{1,\dots,n_\ell\}, \forall j \in \{1,\dots,n_\mathrm{m}\}. \tag{5.20}$$

Using this definition, we can construct the full-domain elasticity properties e_k, $k \in \{1,\dots,6\}$, by mapping the *grouped* modular elastic properties $\mathsf{e}_{\mathrm{g},k} = \left(e_k^{\langle 1\rangle},\dots,e_k^{\langle n_\mathrm{m}\rangle}\right)^\mathrm{T}$ using the group matrix G,

$$\mathsf{e}_k = \mathsf{G}\mathsf{e}_{\mathrm{g},k}, \quad \forall k \in \{1,\dots,6\}, \tag{5.21}$$

with the notation $\bullet^{\langle m\rangle}$ expressing an association of $\bullet$ with the module m. Notice that an association with an element ℓ remains denoted by $\bullet^{(\ell)}$.

Then, the optimization problem (5.12) extended to structural modularity reads as

$$\min_{s,\mathsf{E}^{\langle 1\rangle},\dots,\mathsf{E}^{\langle n_\mathrm{m}\rangle}} \quad s \tag{5.22a}$$

$$\text{s.t. } s - \mathsf{a}^\mathrm{T}\left[\mathsf{K}\left(\mathsf{E}^{\langle 1\rangle},\dots,\mathsf{E}^{\langle n_\mathrm{m}\rangle}\right)\right]^{-1}\mathsf{f} \geq 0, \tag{5.22b}$$

$$\overline{V}\overline{\epsilon}\sum_{\ell=1}^{n_\ell} v^{(\ell)} - \sum_{\ell=1}^{n_\ell}\left[v^{(\ell)}\mathrm{Tr}\left(\mathsf{E}^{\langle g_\ell\rangle}\right)\right] \geq 0, \tag{5.22c}$$

$$\overline{\epsilon} - \mathrm{Tr}\left(\mathsf{E}^{\langle m\rangle}\right) \geq 0, \ \forall m \in \{1,\dots,n_\mathrm{m}\}, \tag{5.22d}$$

$$\mathsf{E}^{\langle m\rangle} \succeq \underline{\epsilon}, \ \forall m \in \{1,\dots,n_\mathrm{m}\}, \tag{5.22e}$$

with the stiffness matrix assembled as

$$\mathsf{K}\left(\mathsf{E}^{\langle 1\rangle},\dots,\mathsf{E}^{\langle n_\mathrm{m}\rangle}\right) = \mathsf{K}_0 + \sum_{\ell=1}^{n_\ell}\mathsf{L}^{(\ell)\mathrm{T}}\mathsf{K}^{(\ell)}(\mathsf{E}^{\langle g_\ell\rangle})\mathsf{L}^{(\ell)}. \tag{5.23}$$

Because of $n_\mathrm{m} < n_\ell$, the optimization problem (5.22) is usually solved much faster than (5.12). After its solution, we obtain the optimal elasticity matrices of individual modules, and their traces are used to build an initial guess for TO as constant pseudo-density fields of individual modules according to

$$\rho_\mathrm{init}^{\langle m\rangle} := \mathrm{Tr}(\mathsf{E}^{\langle m\rangle})/\overline{\epsilon}. \tag{5.24}$$

5.2.2.2 Topology optimization

The above-described procedures based on the free material optimization yield an efficient assembly plan and related optimal material stiffnesses that are constant over individual modules. Ultimately, though, we want to obtain explicit design of modules manufactured from a single material. As mentioned already in Section 5.1, one option is to perform inverse homogenization for each module separately, optionally with enforcing geometrical or mechanical continuity across their boundaries [71, 88]. Here, we opted for a different approach in which the functional continuity is resolved automatically by performing the single-scale topology optimization with a modularity projection following from the optimized assembly plan.

In particular, we adopted the widely-used Solid Isotropic Material with Penalization (SIMP) approach [21, 22]. The design is thus parameterized by a pseudo-density field $0 \leq \rho(\boldsymbol{x}) \leq 1$ that, in turn, dictates the distribution of the material stiffness according to

$$\mathbf{E}(\boldsymbol{x}) = \mathbf{E}_\mathrm{min} + \rho^p(\boldsymbol{x})\left(\mathbf{E}_0 - \mathbf{E}_\mathrm{min}\right), \tag{5.25}$$

where $\mathbf{E}_0 \succ 0$ and $\mathbf{E}_\mathrm{min} \succ 0$ are the stiffness tensors of the bulk material and void space (which attain small but positive eigenvalues in order to avoid a singular Hessian), respectively, and the penalization parameter $p = 3$ is introduced to promote "black-white" designs [22]. Throughout all the simulations, we assumed an isotropic material under plane stress conditions with fixed Poisson ratio $\nu = 0.3$; the pseudo-density ρ thus effectively scales only the Young modulus.

Similarly to the free material optimization problem (5.12), the formulation of the topology optimization builds on the finite element discretization of the governing equation of

linear elasticity with unknown vector $\boldsymbol{\rho}$ containing element-wise constant densities $\rho^{(\ell)}$. For notation simplicity, we assume that each of n_m modules is discretized with $n_{\mathrm{m}\ell}$ finite elements. The modularity is then introduced in the formulation via a boolean group matrix $\mathsf{G}_\mathrm{TO} \in \{0,1\}^{n_\ell \times (n_\mathrm{m} \cdot n_{\mathrm{m}\ell})}$ which maps the module-related densities collected in vector $\boldsymbol{\rho}_\mathrm{g} = \left(\boldsymbol{\rho}^{\langle 1 \rangle}, \dots, \boldsymbol{\rho}^{\langle n_\mathrm{m} \rangle}\right)^\mathrm{T}$ onto full-domain densities $\boldsymbol{\rho}$,

$$\boldsymbol{\rho} = \mathsf{G}_\mathrm{TO}\, \boldsymbol{\rho}_\mathrm{g}\,, \tag{5.26}$$

recall Eq. (5.21). Keeping the same structure as in Eq. (5.12), the modular topology optimization is eventually formalized as

$$\min_{s_\mathrm{TO},\, \boldsymbol{\rho} = \mathsf{G}_\mathrm{TO}\boldsymbol{\rho}_\mathrm{g}} \quad s_\mathrm{TO} \tag{5.27a}$$

$$\text{s.t.} \quad s_\mathrm{TO} - \mathsf{a}^\mathrm{T} \left[\mathsf{K}\left(\boldsymbol{\rho}\right)\right]^{-1} \mathsf{f} \geq 0, \tag{5.27b}$$

$$\overline{V} \sum_{\ell=1}^{n_\ell} v^{(\ell)} - \sum_{\ell=1}^{n_\ell} \rho^{(\ell)} v^{(\ell)} \geq 0. \tag{5.27c}$$

$$1 \;\geq\; \rho^{(\ell)} \geq 0, \;\; \forall \ell \in \{1, \dots, n_\ell\}. \tag{5.27d}$$

The structural stiffness matrix $\mathsf{K}\left(\boldsymbol{\rho}\right)$ follows from Eq. (5.10) (including the design-independent part K_0) with $\mathsf{E}^{(\ell)}$ depending on $\rho^{(\ell)}$ through Eq. (5.25). The slack variable s_TO is introduced here only formally to denote the objective. Contrary to the formulation (5.12), the volume constraint (5.27c) is here formulated directly in element densities $\rho^{(\ell)}$ and no analogue to the eigenvalue constraints (5.12d) and (5.12e) is needed because they are already satisfied due to the interpolation scheme (5.25). On the other hand, this comes at the price of a restricted design space compared to the FMO in which the whole material stiffness matrix is unknown.

Having effectively only volume and bound constraints, we solve the problem (5.27) with the first-order Optimality Criteria (OC) method [22] due to its simplicity and practical performance. Moreover, we modify the standard OC scheme with the gray-scale suppression proposed by Groenwold and Etman [78] to further promote faster convergence towards designs with clearly distinguished phases.

The sensitivity of the objective function $\frac{\partial s_\mathrm{TO}}{\partial \boldsymbol{\rho}_\mathrm{g}}$ under satisfied equality equation (5.27b) follows from the chain rule

$$\frac{\partial s_\mathrm{TO}}{\partial \boldsymbol{\rho}_\mathrm{g}} = \frac{\partial s_\mathrm{TO}}{\partial \boldsymbol{\rho}} \frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{\rho}_\mathrm{g}}\,, \tag{5.28}$$

where $\frac{\partial s_\mathrm{TO}}{\partial \boldsymbol{\rho}}$ is obtained similarly to FMO, see Appendix 5.B.1, and $\frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{\rho}_\mathrm{g}} = \mathsf{G}_\mathrm{TO}$, recall Eq. (5.26). The same chain rule applies also to the sensitivity of the volume constraint (5.27b), denoted as $\frac{\partial V}{\partial \boldsymbol{\rho}_\mathrm{g}}$ further on. In order to avoid mesh-dependency and checkerboard patterns discussed already in Section 5.2.1.2 for FMO, we opt for the sensitivity filter [178] that modifies the objective sensitivity by weight averaging. In the discrete setup, where each $\rho^{(\ell)}$ belongs to the center of gravity $\boldsymbol{x}^{(\ell)}$ of the corresponding element ℓ, the filtered non-modular sensitivity $\widehat{\frac{\partial s_\mathrm{TO}}{\partial \boldsymbol{\rho}}}$ is defined as

$$\widehat{\frac{\partial s_\mathrm{TO}}{\partial \rho^{(i)}}} = \frac{1}{\rho^{(i)} \sum_{j=1}^{n_\ell} w(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})} \sum_{j=1}^{n_\ell} \rho^{(j)} \frac{\partial s_\mathrm{TO}}{\partial \rho^{(i)}}\, w(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})\,, \quad \forall \ell \in \{1, \dots, n_\ell\}\,, \tag{5.29}$$

which leads to a linear relation

$$\widehat{\frac{\partial s_{\mathrm{TO}}}{\partial \boldsymbol{\rho}}} = \mathsf{F}_{\mathrm{w}}(\boldsymbol{\rho}) \frac{\partial s_{\mathrm{TO}}}{\partial \boldsymbol{\rho}} \,, \tag{5.30}$$

with the filtering matrix F_{w} following from Eq. (5.29). Here, we assume a linear weight function

$$w(\boldsymbol{x}, \boldsymbol{y}) = \max(r - \|\boldsymbol{x} - \boldsymbol{y}\|, 0) \tag{5.31}$$

with the radius r.

Denoting the sensitivity vectors $\mathsf{g}^s := \widehat{\frac{\partial s_{\mathrm{TO}}}{\partial \boldsymbol{\rho}_{\mathrm{g}}}} = \mathsf{G}_{\mathrm{TO}}^{\mathrm{T}} \mathsf{F}_w(\boldsymbol{\rho}) \frac{\partial s_{\mathrm{TO}}}{\partial \boldsymbol{\rho}}$ and $\mathsf{g}^V := \frac{\partial V}{\partial \boldsymbol{\rho}_{\mathrm{g}}} = \mathsf{G}_{\mathrm{TO}}^{\mathrm{T}} \frac{\partial V}{\partial \boldsymbol{\rho}}$ for terseness, the update of the design variables $\boldsymbol{\rho}_{\mathrm{g}}$ within the modified OC scheme takes the form

$$\rho_{\mathrm{g},i}^{\mathrm{new}} = \begin{cases} \underline{\rho}_i & \text{if} \quad (\rho_{\mathrm{g},i} \, (B_i)^{\eta})^q \leq \underline{\rho}_i \,, \\ \overline{\rho}_i & \text{if} \quad \overline{\rho}_i \geq (\rho_{\mathrm{g},i} \, (B_i)^{\eta})^q \,, \\ (\rho_{\mathrm{g},i} \, (B_i)^{\eta})^q & \text{otherwise,} \end{cases} \tag{5.32}$$

where $\underline{\rho}_i = \max(\rho_{\mathrm{g},i} - \delta, 0.0)$, $\overline{\rho}_i = \min(\rho_{\mathrm{g},i} + \delta, 1.0)$, and $B_i = \max(-\frac{g_i^s}{\Lambda g_i^v}, \mu)$. The lower index $\bullet_{,i}$ denotes the ith component of the appropriate vector $\bullet$, ranging over the modular unknowns, i.e. $i \in \{1, \ldots, (n_{\mathrm{m}} \cdot n_{\mathrm{m}\ell})\}$. Coefficients η and δ are the damping factor and the move limit of the original OC method, respectively, q is the power coefficient pertinent to the power-law gray-scale suppression, and μ facilitates the inconsistent modification [22] such that μ is set to $-\infty$ for the compliance minimization and 10^{-10} for mechanisms. Finally, Λ denotes the Lagrange multiplier related to the volume constraint (5.27b), whose optimal value is found with a bisection algorithm; see e.g. [22] for more details.

The topology optimization described above was implemented in an in-house, object-oriented C++ code [55], which was linked against Intel® oneAPI Math Kernel Library and included PARDISO direct sparse solver.

5.3 Results

We illustrate the presented theoretical developments with three selected classical problems in topology optimization. In particular, we show the results for the minimum-compliance optimization of the Messerschmitt-Bölkow-Blohm (MBB) beam in Section 5.3.1, Sections 5.3.2 and 5.3.3 cover the design of compliant mechanisms (an inverter and a gripper, respectively), and a novel reusable modular gripper-inverter design is demonstrated in Section 5.3.4.

For all these problems, we limited the relative volume with $\overline{V} = 0.4$. The empty modules identified in the FMO part were not explicitly discretized in TO; consequently, the volume fraction limit considered in TO was modified to match the volume fraction relative to the original domain size. Modules were discretized with a regular grid of 100×100 bilinear fully integrated quadrilateral finite elements. We assumed isotropic, linear-elastic materials under the plane stress with $E = 10^{-9}$ and $\nu = 0.3$ for the voids and $E = 1$ and $\nu = 0.3$ for the bulk material, which in turn lead to $0 \leq \mathrm{Tr}\left(\mathsf{E}^{(\ell)}\right) \leq \overline{\epsilon}$ with the upper-bound on the sum of eigenvalues $\overline{\epsilon} = 235/91$, recall Eq. (5.14). In the topology optimization part, radius r of the sensitivity filter was set to $r = 3.5 h_\ell$ with h_ℓ being the length of an element edge. Optimality Criteria's parameters were as follows: move limit $\delta = 0.1$, damping coefficient $\eta = 0.5$ for

compliance and $\eta = 0.3$ for mechanisms. We used a continuation strategy for the power coefficient q of the gray-scale suppression; starting with the 20^{th} iteration, the initial value $q = 1.0$ was iteratively updated as $q_{\text{new}} = \min(1.01q, 2.0)$. Within all examples, the maximal number of iterations was set to 150. The convergence criteria were based on the change in the objective, $|\delta s_{\text{TO}}| < 10^{-12}$, and the change in the design variables, $\|\delta \boldsymbol{\rho}_{\text{g}}\|_{\infty} < 10^{-2}$. To avoid premature convergence with significant gray areas, quantified with the measure [178]

$$M_{\text{nd}} = \frac{\sum_{\ell=1}^{n_\ell} 4\rho^{(\ell)}(1-\rho^{(\ell)})}{n_\ell}, \tag{5.33}$$

we added a convergence requirement $M_{\text{nd}} < 10^{-3}$. All topology optimization problems were initialized using the module-wise constant guess provided by FMO, as specified in Eq. (5.24).

5.3.1 Messerschmitt-Bölkow-Blohm beam

First, we examine the performance of our scheme for the Messerschmitt-Bölkow-Blohm (MBB) beam problem, i.e., a simply-supported beam with a vertical unit force at its mid-span. The beam is 0.375 in height and 1.0 in width, and discretized by 32×12 equisized square modules as shown in Fig. 5.5.

We start with discretizing the MBB beam with a rotated mesh, recall Section 5.2.1.3, and solve the FMO problem (5.12) that provides us with an optimal compliance of $s_{\text{FMO}} = 5.60$. As is clearly visible in the plot of $\text{Tr}\left(\mathsf{E}^{(\ell)}\right)$ in Fig. 5.6a, the obtained design exhibits pronounced checkerboard pattern near the beam supports indicating that the optimization exploited the coarseness of the discretization and low order of the used elements. Indeed, when using the optimal elasticity matrices projected on a finer discretization containing $3,200 \times 1,200$ elements, i.e., a discretization that is approximately matching the one used subsequently in TO, the compliance significantly increases to $s_{\text{FMO}}^{\text{refined}} = 189.16$. When a simple post-processing step described in Section 5.2.1.2 is performed, see Fig. 5.6b, the objective function increases to $s_{\text{FMO,f}} = 6.92$ on a coarse mesh, but the refined-mesh compliance amounts to $s_{\text{FMO,f}}^{\text{refined}} = 16.15$ only. We believe that the latter increase is justifiable as the bilinear quadrilateral finite elements are prone to the shear locking phenomenon, so a very fine mesh is usually required in the case of structures under bending.

With the post-processed optimized material stiffnesses, we proceed by removing the modules whose maximum eigenvalue of all edge stiffnesses is below 1% of $\bar{\epsilon}$, and follow the clustering procedure explained in Section 5.2.1.3 for a hierarchical merging of the varying material

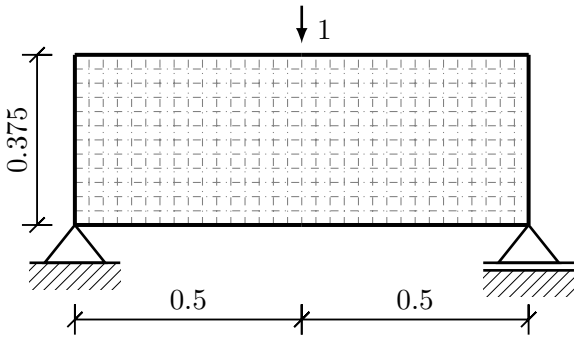
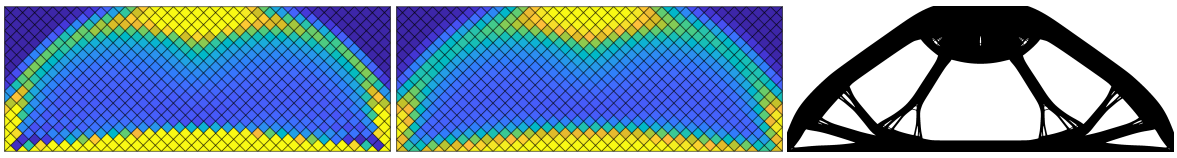


Figure 5.5: Boundary conditions and considered modular discretization of the MBB beam.



(a) (b) (c)

Figure 5.6: Lower-bound designs for the MBB beam problem: (a) optimal traces of material stiffness matrices susceptible to the checkerboard issue ($s_{\mathrm{FMO}} = 5.60$), (b) post-processed material stiffness distribution ($s_{\mathrm{FMO,f}} = 6.92$), and (c) non-modular design obtained by TO ($s_{\mathrm{TO},\infty} = 29.08$).

stiffness matrices into a predefined number of groups, one to four in our case. This clustering procedure is applied twice: once for horizontal and once for vertical edges. In the two-dimensional setting, this process can easily be visualized as follows: the stiffness matrices $\mathsf{E}^{(\ell)}$ are a linear mapping of the engineering strain vector $\boldsymbol{\varepsilon}^{(\ell)}$ onto the stress vector $\boldsymbol{\sigma}^{(\ell)}$, Eq. (5.2). Hence, a sphere of all unit engineering strains is mapped by the elasticity matrix onto a 3-dimensional ellipsoid whose surface corresponds to all admissible stresses induced in the material by arbitrary unit-sized strains, determined using the 2-norm. The clustering process partitions the ellipsoid of all elements in FMO into a given number of bins, trying to replace similar ellipsoids with an approximate one corresponding to the cluster centroid. Consequently, the lengths and orientations of the ellipsoid's axes characterize the material stiffness. Partitioning induced by the clustering is denoted by coloring of these ellipsoids, see first two columns in Fig. 5.7.

After clustering, each module position encompasses a quadruple of edge labels, defining therefore a Wang tile, and these tiles are placed in an assembly plan satisfying the Wang tiling formalism, recall Section 5.1.4 and the developments in [41]. For such modular assemblies, we solve a reduced-sized modular-FMO optimization problem (5.22) to receive effective stiffnesses of individual modules, and use their traces to construct initial guesses for modular TO, recall Section 5.2.2.1.

Although the TO algorithm described in Section 5.2.2.2 usually converges to a neighborhood of a locally-optimal solution, we believe that it is reasonable to assume that the performance of all modular designs shall be bounded by a PUC solution from above, and by a non-modular design from below. Please note, however, that despite the fact that we have not observed any violations of this assumption, it is not guaranteed in general due to the inherent non-convexity of the optimization problems. Using the parameters of TO listed at the beginning of the section, we discretize the interior of each module type using a regular 100×100 mesh, implying that the MBB beam contains 3.84 millions bilinear quadrilateral finite elements. Using the filter radius of 3.5 elements, standard TO converges to the design in Fig. 5.6c of compliance $s_{\mathrm{TO},\infty} = 29.08$. As expected, imposing modularity degrades structural performance: the worst-case single-module, i.e., PUC, design obtained for a single-color clustering possesses the compliance $s_{\mathrm{TO},1} = 63.31$, see the first row in Fig. 5.7. Using two colors, we obtain a design in the second row of Fig. 5.7 composed of 14 modules, of an improved compliance of $s_{\mathrm{TO},2} = 48.18$. The three-color-based clustering produces a design with 25 modules, exhibiting the compliance $s_{\mathrm{TO},3} = 38.26$, see the third row in Fig. 5.7. Finally, the four-color clustering provides us with the design in the fourth row of Fig. 5.7, composed of 46 modules and compliance $s_{\mathrm{TO},4} = 34.30$, which is only 18% more compliant than the non-modular

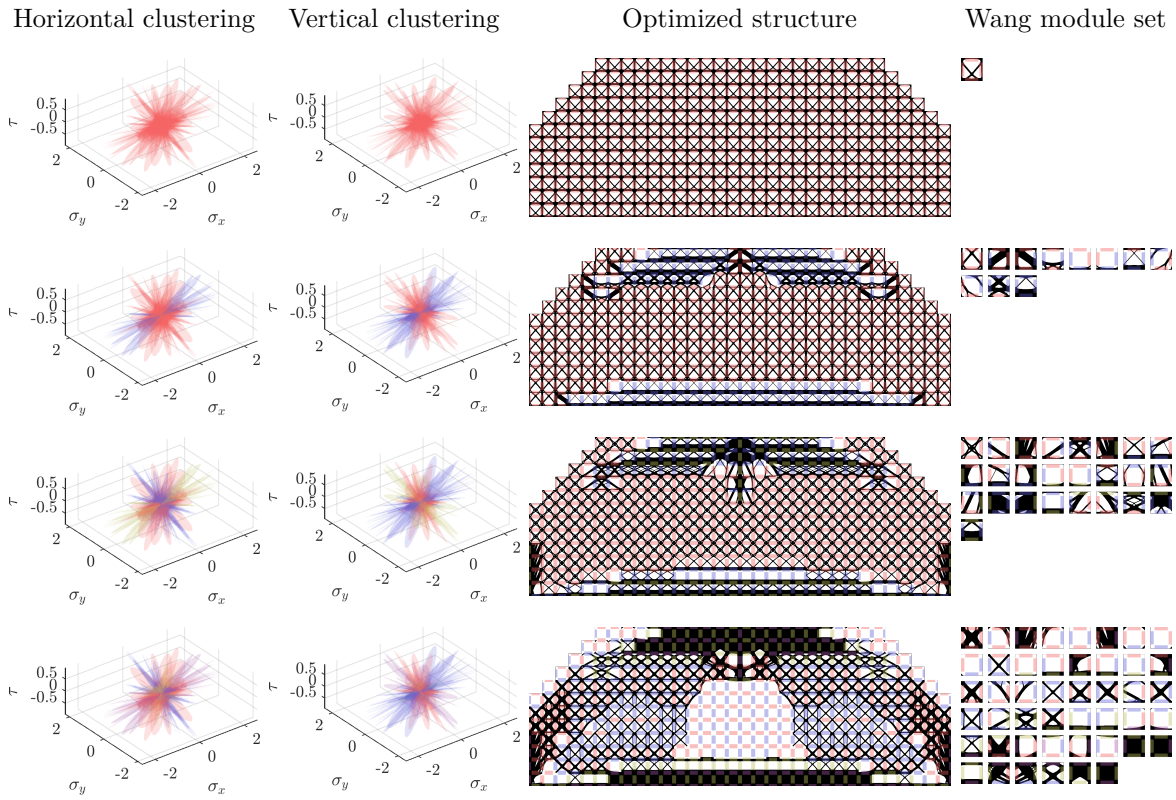


Figure 5.7: Modular designs of the MBB beam problem for the numbers of edge colors ranging from 1 to 4. The first two columns illustrate the partitioning produced by the hierarchical clustering algorithm, the third and fourth columns visualize optimized modular topologies and their assembly.

design. We wish to emphasize that, because our hierarchical clustering algorithm respects the symmetric stiffness distribution, the resulting module sets contain multiple module pairs that have symmetric topologies.

5.3.2 Force inverter

Our second illustrative problem involves designing a modular force inverter mechanism. We consider the design domain to be 1.0 both in width and height and to consist of 10×10 square-sized modules, as shown in Fig. 5.8a. In this domain, 0.01-long segments at the top and bottom of the left edge are fixed. For a coarse, module-based mesh adopted in the FMO problems (5.12) and (5.22), these boundary conditions can not be reproduced exactly, so we collapse these segments into single corner points. The loading is posed by a horizontal unitary force acting in the middle of the left edge height. Similarly to [177], we add an horizontal string of the stiffness $k_{\text{in}} = 1$ to this point of loading and another, output horizontal spring of the stiffness $k_{\text{out}} = 0.05$ is placed at the mid-height of the right edge of the domain.

For these boundary conditions, we search a material distribution that makes the unitary input force minimize the motion of the output point to the right, denoted by u_{out} in Fig. 5.8a. When solving the FMO formulation (5.12) on a rotated mesh and without post-processing, we receive $s_{\text{FMO}} = -1.000$ on a coarse mesh, Fig. 5.8b, and $s_{\text{FMO}}^{\text{refined}} = -0.072$ on a 100-times

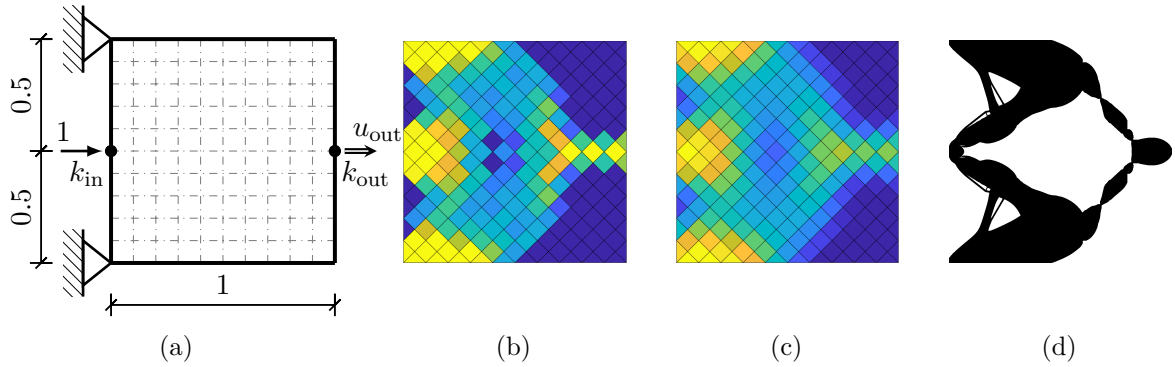


Figure 5.8: (a) Boundary conditions of the inverter problem and assumed modular discretization, and lower-bound designs: (b) optimal traces of material stiffness matrices ($s_{\text{FMO}} = -1.000$), (c) post-processed material stiffness distribution ($s_{\text{FMO,f}} = -0.282$), and (d) non-modular design obtained by TO ($s_{\text{TO},\infty} = -0.354$).

refined mesh with accurate boundary conditions. After application of the post-processing step, we obtain $s_{\text{FMO,f}} = -0.282$ on the coarse mesh, Fig. 5.8c, and $s^{\text{refined}}_{\text{FMO,f}} = -0.128$ on the refined one. These values indicate that even though any visual checkerboard-like patterns is not present in the traces of the stiffness matrices in Fig. 5.8b, the design performance is not physical due to the enormous increase of the output displacement with discretization refinement of optimal design. With the post-processing step, the output displacement inevitably increases, but the mesh shows a much more stable response with refinement.

The clustering procedure, illustrated within the first two columns in Fig. 5.9, produced assembly plans shown in the third column of the same figure. For these assembly plans, we solved the modular-FMO problem (5.22) to receive the initial guesses for TO. Similarly to the MBB beam design, we may provide performance bounds of topologically-optimized designs established by a non-modular design, $s_{\text{TO},\infty} = -0.354$ in Fig. 5.8d, and a single color clustering PUC, $s_{\text{TO},1} = 0.000$, see the top row of Fig. 5.9 showing that the PUC design does not connect the unitary input force to the output properly. However, based on our tests, we believe that such a PUC inverter design does not exist for the selected combination of materials used in TO and for the given modular discretization. Please note that although a PUC force inverter was presented in [213] for a symmetric part of the design domain, its extension to the full domain would require a second module type in our formalism.

As shown in Fig. 5.9, the two-color-based clustering provides a design composed of 8 modules with the performance $s_{\text{TO},2} = -0.211$. Three color-based design possesses $s_{\text{TO},3} = -0.288$, and requires 23 different module types. Finally, 25 module types in the four-color based clustering leads to the output displacement $s_{\text{TO},4} = -0.290$. From the performance point of view, 3- and 4-color designs result in objectives that are only by 18% worse than that of the non-modular design.

5.3.3 Gripper

As the third problem, we consider the traditional gripper mechanism design [177]. Similarly to the force inverter, we adopt a square design domain of the size 1.0×1.0 and split it using 10×10 equisized square modules, see Fig. 5.10a. Following [177] and similarly to the force inverter, we also fix the top and bottom 0.01-long segments of the left edge of the domain,

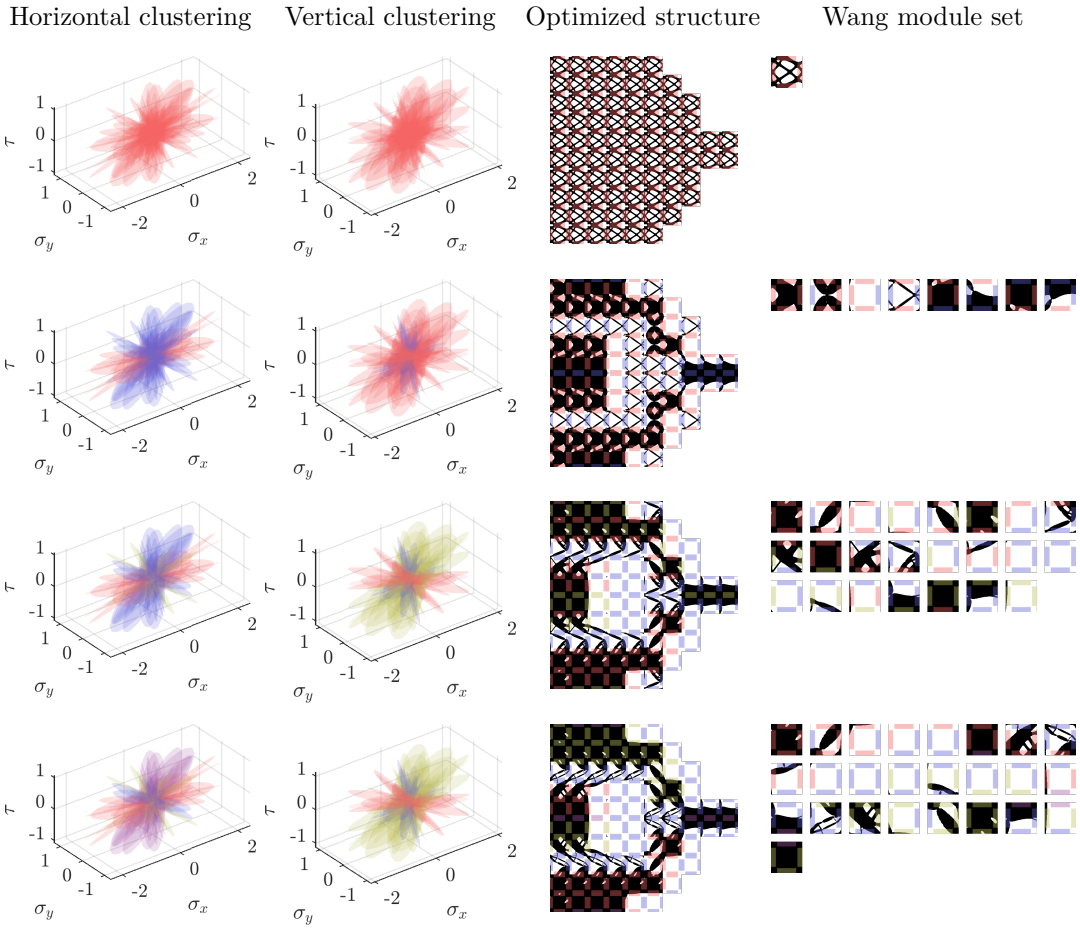


Figure 5.9: Modular designs of the force inverter problem for the numbers of edge colors ranging from 1 to 4. The first two columns illustrate the partitioning produced by the hierarchical clustering algorithm, the third and fourth columns visualize optimized modular topologies and their assembly.

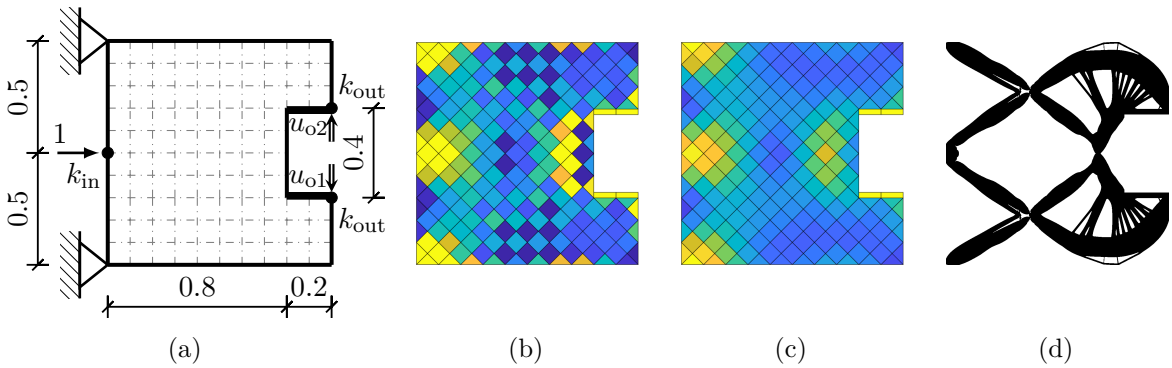


Figure 5.10: Boundary conditions of the gripper problem and assumed modular discretization, and lower-bound designs: (b) optimal traces of material stiffness matrices ($s_{\text{FMO}} = -2.096$), (c) post-processed material stiffness distribution ($s_{\text{FMO,f}} = -0.665$), and (d) non-modular design obtained by TO ($s_{\text{TO},\infty} = -0.616$).

and collapse these supports into single points for coarse discretizations adopted in the FMO problems (5.12) and (5.22).

At the center of the left edge, we place a spring with the stiffness $k_{\text{in}} = 1$ and an input force of a magnitude 1, both aligned with the horizontal direction. In the central part of the opposite, right edge, we prescribe an empty space of 2×4 modules to facilitate a placement of an item that shall be grasped by the mechanism. The grasping occurs through fixed jaws placed at the top and bottom of the empty space. These jaws are 0.025 in height and 0.2 in width and made of a solid isotropic material with $E = 1$ and $\nu = 0.3$, matching so the material adopted in TO and the maximum trace constraint in FMO (5.12d). We note here that to use the edge/module-based discretization and these differently-sized fixed elements in FMO problems, we interpolate their stiffness into the neighboring nodes.

For the mechanism functionality, the input force must trigger closing of the gripper's jaws, which we quantify by minimizing the difference of the displacements $u_{\text{o2}} - u_{\text{u1}}$ of the rightmost points of the jaws, each equipped with a stiffness $k_{\text{out}} = 0.025$ in the vertical direction.

Again, we first solve the FMO optimization problem (5.12) on the rotated mesh, providing us with $s_{\text{FMO}} = -2.096$ on the coarse mesh and $s_{\text{FMO}}^{\text{refined}} = -0.098$ after 100 times refinement. After

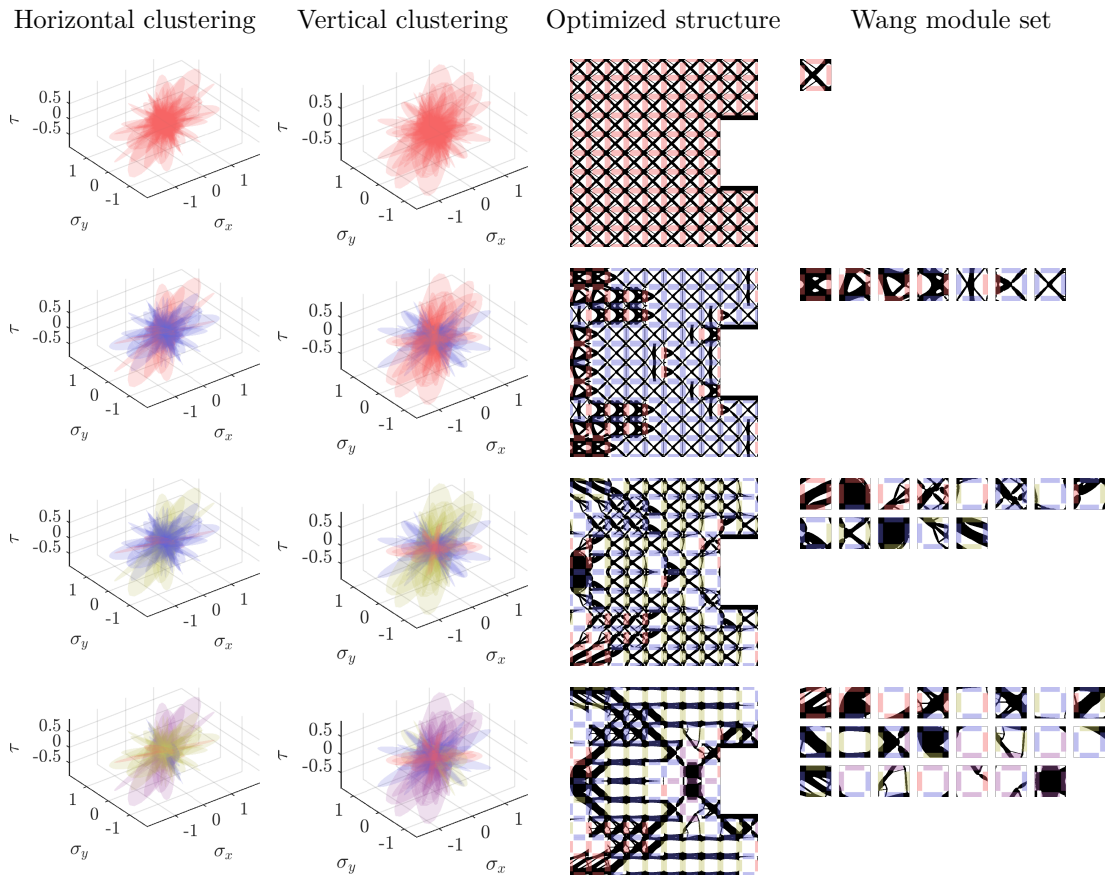


Figure 5.11: Modular designs of the gripper problem for the numbers of edge colors ranging from 1 to 4. The first two columns illustrate the partitioning produced by the hierarchical clustering algorithm, the third and fourth columns visualize optimized modular topologies and their assembly.

the post-processing step, we obtain $s_{\mathrm{FMO,f}} = -0.665$ on the coarse mesh and $s^{\mathrm{refined}}_{\mathrm{FMO,f}} = -0.411$ on the refined mesh; compare Figs. 5.10b and 5.10c for a visualization of the post-processing step. Similarly to the first two examples, also here the refined mesh certifies the importance of filtering or post-processing phase to avoid nonphysical designs.

For the case of the gripper mechanism, the non-modular design produces $s_{\mathrm{TO},\infty} = -0.616$ and the design shown in Fig. 5.10d. The other extreme case, PUC single-color-based clustering at the top row of Fig. 5.11, possesses $s_{\mathrm{TO},1} = -0.080$. While a two-color-based design contains 7 module types and an objective function $s_{\mathrm{TO},2} = -0.151$, a three-color design encompasses 13 modules that induce $s_{\mathrm{TO},3} = -0.196$. Finally, the four-color clustering possesses $s_{\mathrm{TO},4} = -0.399$ and consists of 23 distinct modules; see Fig. 5.11 for a visualization of all modular designs. Notice that for this problem, four-color clustering led to a 35% loss in the performance compared to the non-modular design.

5.3.4 Module reusability in gripper and inverter

To demonstrate merits of modularity, we address a reusable simultaneous design of the previous described inverter and gripper mechanisms as our last example. Based on the multi-functionality of the resulting module designs, it becomes apparent that modularity paves a way from structure-based designs to meta-*material* based ones.

For the reusability design problem, we adopt discretizations and boundary conditions described in Sections 5.3.2 and 5.3.3. However, the reusable designs add another level of complexity as they allow for prioritizing of individual subproblems, i.e., they generate a multi-objective optimization problem. In what follows, we reduce this setting to a single-objective one by introducing weights w_{inv} and w_{grip}, and write the resulting objective as

$$s = w_{\mathrm{inv}} s_{\mathrm{inv}} + w_{\mathrm{grip}} s_{\mathrm{grip}}, \tag{5.34}$$

with s_{inv} and s_{grip} denoting the objective values of the inverter and gripper problems, respectively. Based on a comparison of the objectives of modular gripper and inverter designs and extensive tests (not included), we believe that the choice of $w_{\mathrm{inv}} = 1.0$ and $w_{\mathrm{grip}} = 0.60$ is reasonable, because it balances the terms $w_{\mathrm{inv}} s_{\mathrm{inv}}$ and $w_{\mathrm{grip}} s_{\mathrm{grip}}$ for the non-modular designs, recall Sections 5.3.2 and 5.3.3.

The first step in all previous examples was a solution to the FEM problem on the rotated FE mesh; however, we have already solved these initial steps for both the subproblems in Sections 5.3.2 and 5.3.3 and, hence, we can recycle them here. Thus, instead of another solution, we cluster the former material distributions concurrently, Figs. 5.8c and 5.10c. In the clustering procedure, recall Section 5.2.1.3, we weight the stiffness matrices of the gripper mechanism by w_{grip} and of inverter by w_{inv}. Similarly to TO, these weights maintain

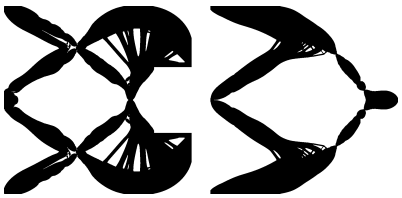


Figure 5.12: Non-modular design of coupled inverter-gripper mechanisms obtained by TO.

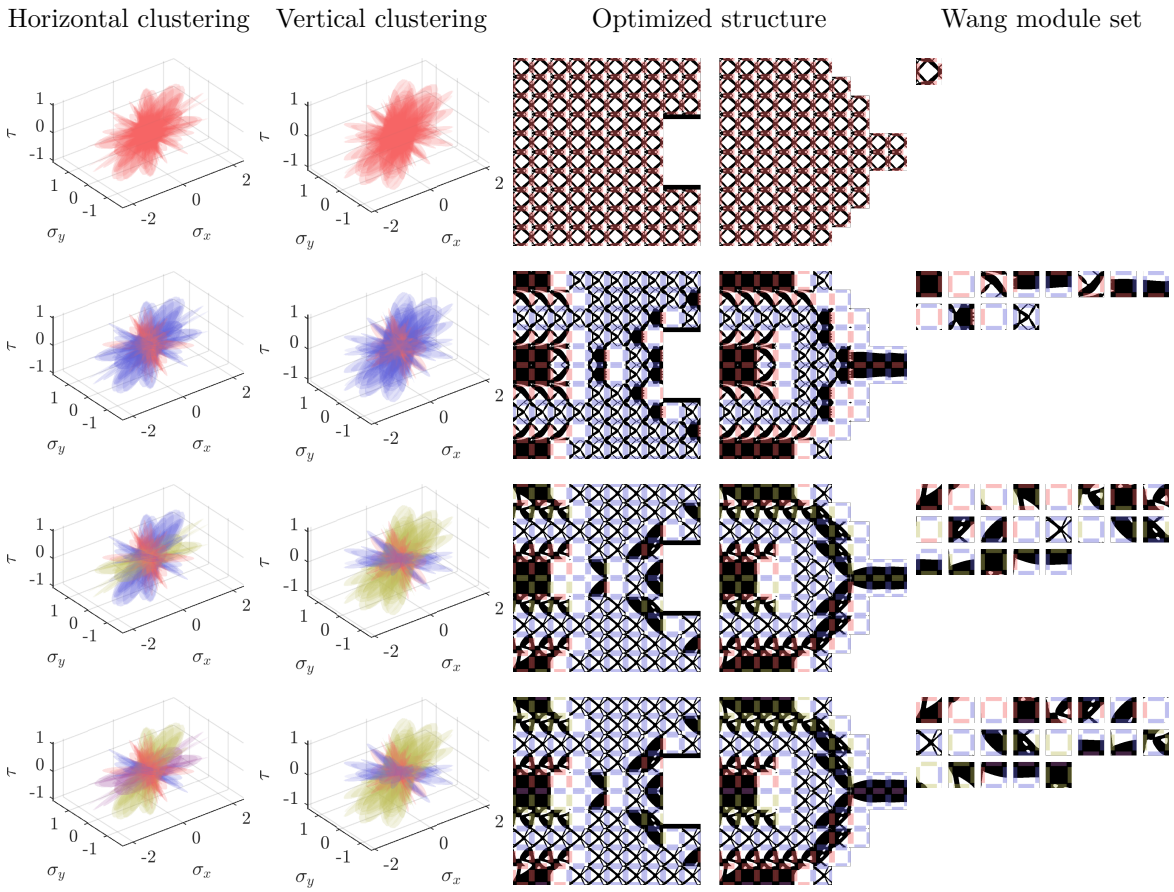


Figure 5.13: Modular designs of reusable inverter-gripper problem for the numbers of edge colors ranging from 1 to 4. While the first two columns illustrate the partitioning produced by the hierarchical clustering algorithm, the third and fourth columns visualize optimized modular topologies and their assembly.

more balanced performance. Finally, we reach the assembly plans and module sets shown in Fig. 5.13.

For TO, we aggregate the independent volume constraints and adopt $\overline{V} = 0.4$ for the union of the domains. Then, the optimized non-modular designs in Fig. 5.12 possess $s_{\mathrm{TO},\infty} = -0.731$ (with $s_{\mathrm{TO},\infty,\mathrm{inv}} = -0.320$ and $s_{\mathrm{TO},\infty,\mathrm{grip}} = 0.687$), and the single-color-based PUC design in the first row of Fig. 5.13 provides $s_{\mathrm{TO},1} = -0.043$ (with $s_{\mathrm{TO},1,\mathrm{inv}} = 0$ and $s_{\mathrm{TO},1,\mathrm{grip}} = -0.072$). In both these settings, the performance of reusable designs naturally approach the independent ones.

The two-color-based design in the second row of Fig. 5.13 contains 12 modules and exhibits $s_{\mathrm{TO},2} = -0.225$ (with $s_{\mathrm{TO},2,\mathrm{inv}} = -0.189$ and $s_{\mathrm{TO},2,\mathrm{grip}} = -0.060$). Here, the resulting assembly seem to be slightly more effective for the inverter mechanism.

Final improvement is obtained for 3-color clustering, which allows for a design in the third row of Fig. 5.13 composed of 21 modules and $s_{\mathrm{TO},3} = -0.275$ (with $s_{\mathrm{TO},3,\mathrm{inv}} = -0.205$ and $s_{\mathrm{TO},3,\mathrm{grip}} = -0.118$). Interestingly, almost the same performance $s_{\mathrm{TO},4} = -0.279$ (with $s_{\mathrm{TO},4,\mathrm{inv}} = -0.208$ and $s_{\mathrm{TO},4,\mathrm{grip}} = -0.118$) results from 4-color clustering, because the 3- and 4-color module sets result in the same assembly plan.

5.4 Conclusions

In this contribution, we have developed a novel heuristic sequential framework for designing modular structures and mechanisms based on Wang tiling formalism. To the best of our knowledge, this manuscript presents the first study which allows modular compliant mechanisms with non-uniform modules to be designed, thus answering one of the challenges laid out in [192, 213].

Our strategy relies on a solution to the Free Material Optimization (FMO) problem (5.12) on a coarse finite element mesh, smoothing the optimal material stiffness field to obtain physical results, and clustering the resulting elasticity matrices. Topologies of individual modules in the assembly plans generated by the previous steps are eventually found by standard Topology Optimization (TO) extended to structural modularity, which we initialize using a well-defined yet problem-specific initial guess. Compared to our prior study on modular-topology optimization of truss structures [192], this framework constitutes a computationally more attractive scheme that neither requires a repetitive solution to the module topology optimization problem nor its convexity. Consequently, the new approach also applies to non-convex problems arising, e.g., in continuum topology optimization.

As the first step of our method, we proposed solving the FMO problem (5.12) on a module-interface-based mesh discretization: using the Pennon [109] optimizer, we solve for the (locally-)optimal elasticity matrices in the discretized domain. Interestingly, they are often prone to the checkerboard phenomenon, at least when the most common bilinear quadrilateral finite elements are employed. Although the resulting degradation of structural responses is very significant and can be verified simply by projecting the optimal material properties onto a refined mesh, the checkerboard issue has surprisingly never been studied or even reported with connection to FMO problems, and thus it remains open to rigorous examination. The reason behind this can be attributed to the fact that the checkerboard patterns in tensorial field are hard to identify in general—contrary to, e.g., the scalar density fields in TO. However, the checkerboard issue in FMO is often discernible when visualizing traces of the optimized material stiffnesses, recall Figs. 5.2 or cf. [88, Figs. 4b and 4e].

Because we use the FMO problem (5.12) only as a heuristic tool only, we eliminate the checkerboard using a single, computationally inexpensive post-processing step performed on the optimization outputs. The essence of post-processing adopts the idea of averaging discretized element stiffnesses onto nodal ones and a backward projection onto element stiffnesses. Because these mappings eliminate the checkerboard-like distribution of stiffnesses, the post-processed designs are more robust with respect to mesh refinement. Furthermore, due to its convexity, the post-processing step preserves both volume and the convex eigenvalue-based constraints. Admittedly, however, the objective function usually increases with the increase magnitude depending on the level of refinement, approaching zero for infinitely fine discretizations.

The particular shape of the finite element discretization used in the problem (5.12) was adopted to provide a simple interface-based definition of individual modules, i.e., each module being defined by four edges and the corresponding quadruple of elasticity matrices. This definition, which is equivalent to Wang tiles, allows us to control the number of interface types among the modules. In this manuscript, we have forbidden rotations of modules, so the clustering of the vertical and horizontal edges were performed independently.

To this goal, we developed a deterministic agglomerative clustering algorithm tailored

to satisfy structural symmetries when present. While in very specific instances of linear elasticity one may rely on partitioning simpler datasets than the stiffness tensors, recall Section 5.1.3, our clustering covers the most general class of problems and, therefore, also applies to multiple-load case minimum-compliance optimization or compliant mechanism design problems.

The resulting clustered designs form modular assembly plans in the form of valid Wang tilings, and their performance can readily be assessed using FMO formulation extended to structural modularity (5.22). However, the modular FMO problem also appears appealing from another viewpoint: its outputs in the form of the traces of module stiffnesses can be advantageously supplied to TO as initial guesses. Subsequently, TO—with the Optimality Criteria method with sensitivity filtering and a continuation—was used to arrive at optimized topologies.

We demonstrated the applicability of the developed framework on three classical problems: design of a minimum-compliant modular MBB beam, a force inverter, and a modular gripper mechanism. Ultimately, we introduced a new problem of designing topologies and assembly of modules that can be reused in both the gripper and inverter mechanisms.

For all these problems, we have shown that with an increasing number of colors, the number of module types not only grows, but the overall structural performance also improves. In the ultimate limit, we would reach the performance of a non-modular design.

This work can be extended in several directions. In addition to the straightforward extension to three-dimensional problems, we plan to focus on the proper handling of the checkerboard phenomenon, e.g., via a filtering scheme inside FMO or by using higher-order finite elements. With simpler manufacturability being one benefit of modularity, we will try to develop constraints to secure a continuous material distribution inside individual modules. Another promising extension constitutes the design of multi-functional mechanisms under finite strains. Such extension, however, would require extending the clustering procedure in Section 5.2.1.3 to deal with time-dependent datasets. In addition, modularity can also be exploited inside optimization and in finite element analyses by efficient preconditioners or domain decomposition methods. Finally, we plan to manufacture the selected modular designs and validate the structural response of the prototypes against predictions made by the model.

5.A Material stiffness matrix in symmetric designs

Let $\boldsymbol{\sigma}^{(i)}, \boldsymbol{\varepsilon}^{(i)}$, and $\boldsymbol{\sigma}^{(j)}, \boldsymbol{\varepsilon}^{(j)}$ be the stress and engineering strain column vectors in the element i and j, respectively. Consider now a symmetric distribution of material stiffnesses and displacement fields in a design domain, and the axis of symmetry aligned with the 1 or 2 axis. Then, for the two elements i and j corresponding to each other in the symmetry, we have $\sigma_{11}^{(i)} = \sigma_{11}^{(j)}$, $\sigma_{22}^{(i)} = \sigma_{22}^{(j)}$, but $\sigma_{12}^{(i)} = -\sigma_{12}^{(j)}$. The strains possess similar symmetries, i.e., $\varepsilon_{11}^{(i)} = \varepsilon_{11}^{(j)}$, $\varepsilon_{22}^{(i)} = \varepsilon_{22}^{(j)}$, and $\gamma_{12}^{(i)} = -\gamma_{12}^{(j)}$.

Considering five test symmetric strain column vectors, the stress symmetry requires

$$\mathsf{E}^{(i)} \begin{pmatrix} \varepsilon_{11}^{(i)} & 0 & 0 & \varepsilon_{11}^{(i)} & \varepsilon_{11}^{(i)} \\ 0 & \varepsilon_{22}^{(i)} & 0 & \varepsilon_{22}^{(i)} & \varepsilon_{22}^{(i)} \\ 0 & 0 & \gamma_{12}^{(i)} & 0 & \gamma_{12}^{(i)} \end{pmatrix} = \mathsf{E}^{(j)} \begin{pmatrix} \varepsilon_{11}^{(i)} & 0 & 0 & \varepsilon_{11}^{(i)} & \varepsilon_{11}^{(i)} \\ 0 & \varepsilon_{22}^{(i)} & 0 & \varepsilon_{22}^{(i)} & \varepsilon_{22}^{(i)} \\ 0 & 0 & -(-\gamma_{12}^{(i)}) & 0 & -(-\gamma_{12}^{(i)}) \end{pmatrix}. \tag{5.35}$$

From the first three test strain fields it follows that the diagonal entries of the elastic material stiffness matrix must match, i.e., $e^{(i)}_{1111} = e^{(j)}_{1111}$, $e^{(i)}_{2222} = e^{(j)}_{2222}$, and $e^{(i)}_{1212} = e^{(j)}_{1212}$. Using this observation, the fourth test strain yields $e^{(i)}_{1122} = e^{(j)}_{1122}$. The relation of the two remaining components of the elastic tensor are determined from the stress balance induced by the fifth field. Then, $e^{(i)}_{1112} = -e^{(j)}_{1112}$ and $e^{(i)}_{2212} = -e^{(j)}_{2212}$.

5.B Derivation of sensitivities

Let us assume an unknown field denoted by y, which encompasses the elements of E in the case of (5.12) and $\boldsymbol{\rho}$ in (5.27). In addition we define

$$a\left(\mathsf{y}\right) \coloneqq \mathsf{a}^{\mathrm{T}}\mathsf{K}(\mathsf{y})^{-1}\mathsf{f} - s, \tag{5.36}$$

so that the inequalities (5.12b) or (5.27b) are equivalent to

$$a(\mathsf{y}) \leq 0. \tag{5.37}$$

In the following subsections, we derive analytical expressions for the first- and second-order derivatives of $a(\mathsf{y})$ with respect to y.

5.B.1 Gradient

From Eq. (5.36) it follows that

$$\frac{\partial a(\mathsf{y})}{\partial y_i} = \mathsf{a}^{\mathrm{T}}\frac{\partial \mathsf{K}(\mathsf{y})^{-1}}{\partial y_i}\mathsf{f}. \tag{5.38}$$

After differentiating the identity

$$\mathsf{I} = \mathsf{K}(\mathsf{y})^{-1}\mathsf{K}(\mathsf{y}), \tag{5.39}$$

with respect to y_i and re-arranging the terms, we obtain

$$\frac{\partial \mathsf{K}(\mathsf{y})^{-1}}{\partial y_i} = -\mathsf{K}(\mathsf{y})^{-1}\frac{\partial \mathsf{K}(\mathsf{y})}{\partial y_i}\mathsf{K}(\mathsf{y})^{-1}. \tag{5.40}$$

Inserting (5.40) into (5.38), we obtain

$$\frac{\partial a(\mathsf{y})}{\partial y_i} = -\mathsf{a}^{\mathrm{T}}\mathsf{K}(\mathsf{y})^{-1}\frac{\partial \mathsf{K}(\mathsf{y})}{\partial y_i}\mathsf{K}(\mathsf{y})^{-1}\mathsf{f} = -\boldsymbol{\lambda}^{\mathrm{T}}\frac{\partial \mathsf{K}(\mathsf{y})}{\partial y_i}\mathsf{u}, \tag{5.41}$$

with $\boldsymbol{\lambda}$ and u being solutions to

$$\mathsf{K}(\mathsf{y})\mathsf{u} = \mathsf{f}, \tag{5.42a}$$
$$\mathsf{K}(\mathsf{y})\boldsymbol{\lambda} = \mathsf{a}. \tag{5.42b}$$

5.B.2 Hessian

In order to compute the Hessian matrix, which we apply only in solutions of the FMO problems (5.12) and (5.22), we first differentiate (5.38) with respect to y_j, i.e.,

$$\frac{\partial^2 a(\mathsf{y})}{\partial y_i \partial y_j} = \mathsf{a}^{\mathrm{T}} \frac{\partial^2 \mathsf{K}(\mathsf{y})^{-1}}{\partial y_i \partial y_j} \mathsf{f} \tag{5.43}$$

To compute $\frac{\partial \mathsf{K}(\mathsf{y})^{-1}}{\partial y_i \partial y_j}$, we differentiate (5.40) with respect to y_j to obtain

$$\frac{\partial^2 \mathsf{K}(\mathsf{y})^{-1}}{\partial y_i \partial y_j} = -\frac{\partial \mathsf{K}(\mathsf{y})^{-1}}{\partial y_j} \frac{\partial \mathsf{K}(\mathsf{y})}{\partial y_i} \mathsf{K}(\mathsf{y})^{-1} - \mathsf{K}(\mathsf{y})^{-1} \frac{\partial \mathsf{K}(\mathsf{y})}{\partial y_i} \frac{\partial \mathsf{K}(\mathsf{y})^{-1}}{\partial y_j} - \mathsf{K}(\mathsf{y})^{-1} \frac{\partial^2 \mathsf{K}(\mathsf{y})}{\partial y_i \partial y_j} \mathsf{K}(\mathsf{y})^{-1}. \tag{5.44}$$

Using the identity (5.40) and exploiting that the stiffness matrix is symmetric, we receive

$$\frac{\partial^2 \mathsf{K}(\mathsf{y})^{-1}}{\partial y_i \partial y_j} = 2\mathsf{K}(\mathsf{y})^{-1} \frac{\partial \mathsf{K}(\mathsf{y})}{\partial y_i} \mathsf{K}(\mathsf{y})^{-1} \frac{\partial \mathsf{K}(\mathsf{y})}{\partial y_j} \mathsf{K}(\mathsf{y})^{-1} - \mathsf{K}(\mathsf{y})^{-1} \frac{\partial^2 \mathsf{K}(\mathsf{y})}{\partial y_i \partial y_j} \mathsf{K}(\mathsf{y})^{-1}. \tag{5.45}$$

In the FMO problem (5.12), the stiffness matrix is a linear function of y, so the second term vanishes. Therefore, equations (5.40), (5.45) combined with (5.43) yield

$$\frac{\partial^2 a(\mathsf{y})}{\partial y_i \partial y_j} = 2\mathsf{a}^{\mathrm{T}} \mathsf{K}(\mathsf{y})^{-1} \frac{\partial \mathsf{K}(\mathsf{y})}{\partial y_i} \mathsf{K}(\mathsf{y})^{-1} \frac{\partial \mathsf{K}(\mathsf{y})}{\partial y_j} \mathsf{K}(\mathsf{y})^{-1} \mathsf{f} = 2\boldsymbol{\lambda}^{\mathrm{T}} \frac{\partial \mathsf{K}(\mathsf{y})}{\partial y_i} \mathsf{K}(\mathsf{y})^{-1} \frac{\partial \mathsf{K}(\mathsf{y})}{\partial y_j} \mathsf{u}. \tag{5.46}$$

Let us now define matrices $\mathsf{U}, \boldsymbol{\Lambda}$ with the i-th columns computed as

$$\mathsf{U}_i(\mathsf{y}) = \frac{\partial \mathsf{K}(\mathsf{y})}{\partial y_i} \mathsf{u}, \tag{5.47a}$$

$$\boldsymbol{\Lambda}_i(\mathsf{y}) = \frac{\partial \mathsf{K}(\mathsf{y})}{\partial y_i} \boldsymbol{\lambda}. \tag{5.47b}$$

Then, the Hessian matrix $\mathsf{H}(\mathsf{y})$ reads as

$$\mathsf{H}(\mathsf{y}) = 2\left[\boldsymbol{\Lambda}(\mathsf{y})\right]^{\mathrm{T}} \mathsf{K}(\mathsf{y})^{-1} \mathsf{U}(\mathsf{y}). \tag{5.48}$$

When (5.12b) or (5.27b) is a compliance constraint, $\boldsymbol{\Lambda}(\mathsf{y}) = \mathsf{U}(\mathsf{y})$. Because $\mathsf{K}(\mathsf{y})^{-1} \succ 0$, we also have $\mathsf{H}(\mathsf{y}) \succ 0$ for all y. Consequently, the constraint (5.12b) is convex.

5.C Scaling of the optimization problem

In the case of $\mathsf{f} \neq \mathsf{a}$, numerical solution to (5.12) appears to be sensitive to scaling of the design variables. To make the solution process more robust, we substitute $\mathsf{E}^{(\ell)}$ with $\bar{\epsilon}\tilde{\mathsf{E}}^{(\ell)}$,

which implies that $\mathrm{Tr}(\tilde{\mathsf{E}}^{(\ell)}) \in \langle 3\underline{\epsilon}/\overline{\epsilon}, 1\rangle$. Consequently, we arrive at the modified formulation

$$\min_{s_{\mathrm{sc}},\tilde{\mathsf{E}}^{(1)},\ldots,\tilde{\mathsf{E}}^{(n_\ell)}} \quad s_{\mathrm{sc}} \tag{5.49a}$$

$$\text{s.t. } s_{\mathrm{sc}} - \mathsf{a}^{\mathrm{T}} \left[\frac{\mathsf{K}_0}{\overline{\epsilon}} + \sum_{\ell=1}^{n_\ell} \mathsf{K}^{(\ell)} \left(\tilde{\mathsf{E}}^{(\ell)} \right) \right]^{-1} \mathsf{f} \geq 0, \tag{5.49b}$$

$$\frac{\sum_{\ell=1}^{n_\ell} v^{(\ell)}}{\overline{V}} - \sum_{\ell=1}^{n_\ell} \frac{v^{(\ell)} \mathrm{Tr}\left(\tilde{\mathsf{E}}^{(\ell)} \right)}{\overline{V}} \geq 0, \tag{5.49c}$$

$$1 - \mathrm{Tr}\left(\tilde{\mathsf{E}}^{(\ell)} \right) \geq 0, \quad \forall e \in \{1, \ldots, n_\ell\}, \tag{5.49d}$$

$$\tilde{\mathsf{E}}^{(\ell)} \succeq \underline{\epsilon}/\overline{\epsilon}, \ \forall e \in \{1, \ldots, n_\ell\}, \tag{5.49e}$$

from which the (locally) optimal solution is recovered as

$$s = s_{\mathrm{sc}}/\overline{\epsilon} \tag{5.50a}$$

$$\mathsf{E}^{(\ell)} = \overline{\epsilon}\tilde{\mathsf{E}}^{(\ell)}, \quad \forall e \in \{1, \ldots, n_\ell\}. \tag{5.50b}$$

Chapter 6

Conclusions

This thesis aimed at developing methods for modular-topology optimization of structures and mechanisms, with the ultimate goal of bringing sustainability into topology optimization. In particular, we asked

1. How should the module topologies look like?
2. How to encode the compatibility information of modules?
3. How to arrange the modules in an assembly plan?

We provided partial answer to these questions in the form of the collection of five manuscripts.

In the first manuscript in Chapter 2, we considered Question 1 alone and investigated an industrial problem of designing a segmented minimum-weight truss internal structure of a thin-walled composite beam susceptible to elastic wall instabilities. To limit these instabilities manifested in low shear and wall buckling free-vibration eigenfrequencies, we have developed a linear semidefinite programming formulation to design a lightweight internal structure increasing the fundamental eigenfrequency and carrying the manufacturing loads. To accelerate the optimization, we proposed condensation of the beam surface stiffnesses with the Schur complement lemma. The optimization output—optimal segmented internal structure—was manufactured by 3D printing, assembled, and wound with carbon fibers saturated with the epoxy resin. After curing and finalizing the beam surface, we verified the model against industrial finite element equivalent, and validated it with the roving hammer test. We concluded that the accuracy was within the error range that is expected for composite structures. Thus, the modular prototype effectively reduced the occurrence of wall instabilities. The CompoTech company, which manufactured the beam prototype, exhibited the beam at several industrial expositions, including EMO 2017 in Hannover and JEC World 2019 in Paris. At the latter exhibition, the beam prototype was selected within the three finalists in the Additive Manufacturing category of the JEC World Innovation Awards competition[1].

The theme of Chapter 3 flourished from the outcomes of Chapter 2, where we modeled the internal structure using the truss finite elements. Although the truss elements underestimate structural stiffness when compared to frames (and thereby deliver reliable), we refer the reader to Section 3.A, modeling with frames would have been more accurate. Interestingly, we found that the topology optimization problems incorporating bending stiffnesses parameterized with continuous cross-section properties were unresolved. To circumvent this in Chapter 3, we derived a non-linear semidefinite programming formulation, minimizing a linear function over

[1] `https://www.jec-world.events/essential_grid/optimised-3d-printed-internal-beam-structure/`

a compact semi-algebraic feasible set. The resulting optimization problem can be solved to certified global optimality using the moment-sum-of-squares (Lasserre) hierarchy. In addition, we showed how to project the relaxed solutions onto the feasible set, hence generating feasible upper bounds. The lower bounds emergent in the relaxations and the projected upper bounds produced a simple and computationally inexpensive certificate of global ε-optimality. Finally, we have shown that the optimality gap approaches zero in the limit if there is a unique global minimizer.[2] We demonstrated the method capabilities on several illustrative minimum-compliance optimization problems with extensions towards the multiple loads and self-weight settings. In all our test cases, the convergence of hierarchy was finite, all global solutions were extracted, and only a low relaxation degree was required.

In Chapter 4, we dealt with Questions 1–3 concurrently for the first time, thus introducing the modular-topology optimization concept. We adopted the corner Wang tiling formalism—defining a fixed-sized module set with a predefined connectivity—to describe the modular assembly plans. Similarly to Chapter 2, we discretized the module interiors with a truss ground structure to allow for a global solution to the lower-level module design problem. However, here we formalized it as a second-order conic program that incorporated the minimum-compliance optimization, multiple load cases, constraints on axial stresses, and module reusability within multiple structures. Thanks to its convexity, the performance of individual assembly plans was assessed uniquely. To account for the bilevel problem structure, we developed a metaheuristics-mathematical programming framework, where a genetic algorithm optimized the assembly plans. We illustrated our method with several problems, ultimately including module reusability within L-shaped and simply-supported beam structures. The illustrative problems revealed that while modularity degrades structural performance, the loss is fairly small for optimized assemblies.

Our final exposition towards modular-topology optimization in Chapter 5 aimed to overcome two shortcomings of the method from Chapter 4: convexity assumption for the module topology design problem, and computationally intensive bilevel optimization. To these goals, we have developed a sequential heuristic method by combining the free material design, clustering, and continuum topology optimization. As its first step, we proposed solving the free material optimization on an module-interface-based discretization. However, its output of (locally-)optimal constitutive parameters exhibited checkerboard patterns. Although these patterns also appear in published literature, surprisingly, to the best of our knowledge, this discretization issue has neither been considered nor commented with the free material optimization problems yet. In Chapter 5, we developed a single post-processing step to eliminate the checkerboards. The post-processed elasticity matrices were subsequently partitioned into a predefined number of groups by a novel hierarchical clustering algorithm accounting for symmetric distribution of the elastic properties when present. After interpreting the cluster labels as the colors of Wang tiles, the clustering produces an assembly plan in the form of Wang tilings. Finally, starting with an initial guess based on the free material optimization, we adopted conventional continuum topology optimization extended to modularity to arrive at optimized module topologies. We illustrated this procedure with a problem set covering the design of modular structures and compliant mechanisms, both possibly supplemented with module reusability.

In contrast to Chapter 2 that searched for module topologies only, in Appendix A, we have considered solely the assembly plan design problem (Question 3). Observing that

[2] After publication, we observed that the assumption for the zero optimality gap can be generalized with the same argument to requiring the set of the global minimizers to be convex.

generation of bounded Wang tilings is generally $\mathcal{NP}$-complete, we developed four integer programming formulations for the tiling production: a decision variant, maximum rectangular tiling, maximum cover, and maximum adjacency constraint satisfaction. For the maximum cover version, which seems to be the closest to our requirements in Chapter 5, we proposed a simple heuristic scheme initialized with one of three developed algorithms. Among them, one was shown to provide the 1/2 approximation guarantee for arbitrary tilesets, and another a 2/3 guarantee for tilesets with cyclic transducers. While concluding the chapter with numerical performance of the formulations and of the heuristic algorithms, we revealed two unexpected errors in well-known aperiodic tilesets: an unusable tile in the Knuth tile set [103], and periodicity of the Lagae *et al.* [121] corner tileset.

6.1 Perspectives for future research

The content of this thesis suggests several possible future extensions. Considering modular-topology optimization, these extensions may involve finite strains, multi-stable and multi-functional complaint mechanisms, approaching mechanical computers or digital metamaterials [90]. With respect to practical applicability, manufacturing constraints need to be incorporated and the approach generalized to 3D.

Second direction of future research may extend the developments in Chapter 3. Although the history of the topology optimization field can be traced back by more than a century [136], several important problems still remain for which no global solution method has been developed, e.g., the forced and free-vibration problems of bending-resistant structures. We believe that the moment-sum-of-squares approach may be a viable option for their solution. However, also scalability of the solution method must be considered, e.g., by exploiting the optimization problem structure [107, 82].

Finally, we believe that the checkerboard issue in free material optimization deserves a more rigorous investigation and treatment, beyond the heuristic procedure introduced in Chapter 5.

Appendix A

On bounded Wang tilings

Abstract: *Determining tilability of a fixed domain with Wang tiles is known to be an $\mathcal{NP}$-complete problem. Driven by emerging applications in materials engineering, we develop four novel integer programming formulations to generate valid or nearly-valid Wang tilings: a decision, maximum-rectangular tiling, maximum cover, and maximum adjacency constraint satisfaction variants. Furthermore, we supplement these programs with practical extensions to tile-based, color-based, packing, and variable-sized periodic constraints to facilitate a more delicate control over the resulting tilings. Second, we introduce an efficient heuristic algorithm for the maximum-cover optimization variant and derive simple modifications to (i) provide a $1/2$ approximation guarantee for arbitrary tile sets, and (ii) a $2/3$ approximation factor for tile sets with cyclic transducers. Finally, we assess the performance of the integer programming formulations and of the heuristic algorithms, showing that the heuristics provides very competitive outputs. We conclude with two theoretical discoveries: the Knuth tile set contains a tile unusable in two-way infinite tilings, and the Lagae corner tile set is not aperiodic as was claimed.*

A.1 Introduction

Wang tiles, non-rotatable unit squares with colored edges, constitute a formalism introduced by Wang to popularize the $\forall\exists\forall$ decidability problem of predicate calculus [202]. Formulating an equivalent domino problem, Wang considered an infinite number of copies of an arbitrary set of Wang tiles and investigated whether there exists a simply-connected valid tiling of the infinite plane.

Wang conjectured that only the tile sets that form a torus, i.e., cover a periodic simply-connected rectangular domain with identical coloring of the opposite edges, generate infinite valid tilings [201]. However, Berger [26] disproved the conjecture by finding a tile set that covers the infinite plane only aperiodically by exploiting Kahr's reduction of the Turing (machine) halting problem [190, 46] to the origin-constrained domino problem [94]. Hence, the domino problem was proven to be *undecidable* and, consequently, no general finite algorithm for producing infinite valid tilings exists.

Far less attention has been paid to the finite version of the domino problem, *bounded tiling*, i.e., searching for a fixed-sized valid tiling generated by an arbitrary tile set. Although the problem is known to be $\mathcal{NP}$-complete in general [125] and [124, Theorem 7.2.1]—and indeed decidable—most of the available approaches exploit specific properties of particular tile sets, e.g., [41, 50, 118, 150]. However, several closely related works address the (tile) packing problem for edge-matching puzzles, in which all tiles from the tile set are placed exactly once, see [114, 120, 166] and [168] for an approach aiming at the famous Eternity II puzzle.

In this chapter, we investigate the bounded Wang tiling problem in its full generality. To this goal, we first survey the most significant *aperiodic* tile sets in Section A.1.1 and applications of Wang tiles in Section A.1.2. In Section A.1.3, we list available algorithms for generation of Wang tilings. Finally, our aims and contributions appear summarized in Section A.1.4.

A.1.1 Aperiodic tile sets

The originally unexpected property of Wang tile sets—aperiodicity—resulted in a long-term competition among scientists in mathematical logic, computer science, discrete mathematics, and even recreational mathematicians to find the aperiodic tile set of the minimum cardinality [79, Chapter 11]. Starting from the Berger tile set containing $20,426$ tiles in 1964 [25, 26], it took almost 50 years until the two sets of 11 tiles were found and proved to be minimal [92]; see Fig. A.1 for a graphical overview of the selected historical developments.

In 1966, Läuchli sent to Wang an aperiodic set of 40 tiles over 16 colors, but it remained unpublished until 1975 [204]. Meanwhile, unaware of the Läuchli's result, Knuth [103] simplified Berger's set to 92 tiles over 26 colors; and Robinson developed sets of 104 and 52 tiles over 8 colors in 1967 [154], of 56 tiles over 12 colors in 1971 [161], and noted an existence of a set of 35 tiles [161].

In 1973, Penrose developed a new approach based on kites and darts tiling, leading to a set of 34 tiles. Robinson, being in contact with Penrose, modified the Penrose's approach to reach a reduced set of 32 tiles over 16 colors [79]. Using the same technique together with Penrose rhombs tiling, Grünbaum and Shephard [79] obtained a set of 24 tiles over 9 colors in 1987.

Another two tile sets were discovered by Ammann. In 1978, he used the so-called Ammann bars to reach 16 tiles over 6 colors [162]. Building on the Ammann's A2 tiling, see, e.g., [79],

1964 – 20, 426 and 104 Wang tiles by Berger [25, 26]
1966 – 40 Wang tiles over 16 colors by Läuchli [204]
1967 – 104 and 52 Wang tiles over 8 colors by Robinson [160, 154]
1968 – 92 Wang tiles over 26 colors by Knuth [103] (reducible to 86 tiles [105])
1971 – 56 Wang tiles over 12 colors and 35 tiles by Robinson [161]
1973 – 34 Wang tiles by Penrose [79]
1973 – 32 Wang tiles over 16 colors by Robinson [79]
1977 – 24 Wang tiles over 24 colors by Robinson [79]
1978 – 16 Wang tiles over 6 colors by Ammann [162]
1987 – 24 Wang tiles over 9 colors by Grünbaum and Shephard [79]
1995 – 64 Wang tiles by Senechal [172]
1996 – 14 Wang tiles over 6 colors by Kari [98]
1996 – 13 Wang tiles over 5 colors by Čulík [44]
1999 – 696 Wang tiles by Kari and Papasoglu [99] (deterministic tile set)
2008 – 104 Wang tiles by Ollinger [150]
2018 – 19 Wang tiles over 16 colors by Labbé [117] (self-similar tile set)
2021 – 11 Wang tiles over 5 and 4 colors by Jeandel and Rao [92]

Figure A.1: List of aperiodic Wang tile sets.

Robinson obtained a set of 24 tiles over 24 colors in 1977.

Subsequent size reduction of the smallest aperiodic set occurred in 1996, as Kari [98] developed a new method based on Mealy machines multiplying Beatty sequences, and presented a set of 14 tiles over 6 colors. Čulík [44], using the same approach, reduced the set even further to 13 tiles over 5 colors.

The search for the minimal aperiodic set has been concluded by Jeandel and Rao [92], who used an enumeration approach to find aperiodic sets of 11 tiles over 4 and 5 colors and proved non-existence of an aperiodic set either containing 10 or fewer tiles or labeled by less than 4 colors.

In addition to the original edge-based Wang tiles, in 2006 Lagae and Dutré [119] described a subset of the Wang tiles, the *corner* tiles (we refer to Appendix A.A for their relation to edge-based Wang tiles), with the matching information stored in the colored corners instead of the edges. In the same year, they constructed multiple aperiodic sets of corner tiles [121], out of which the set of 44 corner tiles over 6 colors was the smallest one. The set was further simplified by Nurmi [145] to 30 corner tiles over 6 colors and both were claimed to be aperiodic.

A.1.2 Applications of Wang Tiles

Thanks to the property of particular tile sets to generate aperiodic tilings, Wang tiles gained interest among several disciplines. Building on the original purpose of Wang tiles, proofs in the first-order logic [201], they were also used in cellular automata theory [97], topology,

group theory [42], and symbolic dynamical systems [139].

In computer graphics, Stam [181] adopted Wang tiles to generate aperiodic textures. After Cohen *et al.* [41] recognized that stochastic nonperiodic tilings are easier to handle, the Wang-tile-based approach to generating seamless textures became popular, including also the generation of point patterns [41, 86].

In science, Wang tiles and other related aperiodic tilings served as the key tool to understand the 5-fold symmetry of electron diffraction patterns of quasicrystals [158, 172]. Another applications at the nanoscale involve molecular DNA-based realization of Wang tiles, introduced by Winfree *et al.* [209], which provided a self-assembly of biological nanostructures into aperiodic patterns. The self-assembly process of DNA Wang tiles also powered custom DNA-based computations [171], fueled by Turing completeness of Wang tiles [26, 204].

Besides nanoscale, Wang tiles were also used for efficient compression [143] and reconstruction [56] of nonperiodic microstructures, speeding up finite element analyses of random heterogeneous materials [58], such as open-cell metal foams [54]. Furthermore, we have developed a bilevel optimization approach to design modular truss materials based on the corner Wang tiling formalism (Chapter 4) and a clustering-based method for designing modular structures and mechanisms with continuum topology optimization (Chapter 5). Finally, Jílek *et al.* [93] developed a centimeter-scale self-assembly procedure.

A.1.3 Wang tiling generation algorithms

To the best of our knowledge, nor general approaches to solving the bounded tiling problem have been reported in the literature; the only available results are specific to a single family of tile sets [41, 50, 118, 150], or consider infinite thin strips [92].

In what follows, we describe the gist of three tiling algorithms: substitution-based, stochastic, and transducer-based.

A.1.3.1 Substitution-based tiling algorithm

Substitution is a map $S : \mathcal{T} \mapsto \mathfrak{T}$ that assigns a tiling $\mathfrak{T}_k$ to each tile $k \in \mathcal{T}$. Consequently, arbitrary-sized tilings are generated by placing a single tile k, and then repeating iteratively the substitution rule on all tiles that have already been placed [150]. Hence, the tiling "grows" in each iteration. Clearly, such procedure has a low complexity, but only very specific tile sets allow for a substitution rule that generates valid tilings.

A.1.3.2 Stochastic tiling algorithms

In computer graphics, Wang tiles are mostly used for a generation of visually appealing yet compressed textures. To this goal, it is essential to generate these nonperiodic patterns quickly, which is best achieved with stochastic tile sets—usually containing all combinations of edge labels for a given number of colors. For example, in the stochastic tiling algorithm [41] the tiling is generated row-wise, such that the neighbor of any tile that has already been placed can always be selected from at least two tiles at random. This approach was further extended towards the hash-based direct stochastic tiling algorithm [118]. Note that stochastic algorithms enable straightforward enforcement of several tile- or edge-based constraints.

A.1.3.3 Transducer-based tiling algorithm

The transducer-based tiling algorithm [92] builds on the fact that the 1D domino problem is decidable and can be solved in a polynomial time because the bi-infinite path is formed by an arbitrary cycle in the transducer graph $G_{\mathrm{t,h}}$, see Section A.2 for clarification. To generate valid tilings of multiple rows, it is required, however, to compute the product of several transducers. Hence, we must enumerate all feasible valid tilings of requested height and unit width, and then find a path of the given length in the transducer graph of the just-formed tile set. Obviously, this approach works well for tiling of thin strips; however, it is impractical for larger nearly-square domain.

A.1.4 Aims and novelty

In this contribution, we consider the bounded Wang tiling in its general form, thereby allowing for an arbitrary tile set and control over the resulting tilings. As follows from the above state-of-the-art survey, no such algorithm has been published yet.

We believe that development of such algorithms is important from multiple reasons. First, we have already investigated modeling of non-periodic and stochastic microstructures with Wang tilings, e.g., [143, 56, 54, 57, 58]. We hope that extension of our methods to more general tile sets would allow characterizing a broader class of non-periodic conventional materials and meta-materials [43, 219, 141].

In the reverse direction, we have also applied the Wang tiling formalism to perform a bi-level design of modular materials and structures [192].

Our current results remain limited to stochastic tile sets and broadening the design space has the potential to improve upon the performance of optimized designs.

Apart from emerging applications in materials engineering, we believe that developing a unified methodology is of independent interest, e.g., for the verification of available results there, we justify this claim by finding two errors in well-established aperiodic tile sets.

To these goals, we first provide the necessary definitions in Section A.2 to make the chapter self-contained. The subsequent part of the chapter is devoted to four integer programming formulations for generation of valid tilings: decision variant in Section A.3.1, maximum rectangular valid tiling in Section A.3.2, maximum-cover in Section A.3.3, and maximum adjacency constraint satisfaction in Section A.3.4. To allow for a delicate control over the resulting tilings, we also include simple extensions to prescribe tile- and color-based boundary conditions, (a variable-sized) periodic constraint, and the tile-packing constraint, Section A.3.5.

Due to the complexity of the proposed formulations, in Section A.4 we propose a heuristic graph-based algorithm to tackle the maximum-cover optimization variant from Section A.3.3. The developed algorithm relies on solutions to shortest path problems in directed acyclic graphs, hence possesses a low asymptotic complexity. Further, we show that the algorithm provides an approximation ratio of 2/3 for the tile sets whose transducer graphs are cyclic.

Section A.5.1 collects results on the computational assessment of the integer programming formulations and heuristics (Section A.4), and on the benchmarking of the periodic tile packing formulation against the algorithm of Lagae and Dutré [120] in Section A.5.3. We close the section with two surprising observations when using integer programming formulations for two well-known aperiodic tile sets: the Knuth [103] tile set of 92 tiles contains a tile unusable

in infinite simply-connected valid tilings, Section A.5.4, and the Lagae *et al.* [121] tile set of 44 corner tiles is not aperiodic, Section A.5.5. Finally, we summarize our results in Section A.6.

A.2 Notation and preliminaries

Assuming a finite set of *color codes* $\mathcal{C} = \{1, 2, \ldots, n_\mathrm{c}\} \subset \mathbb{N}$, the *(Wang) tile* k is a quadruple of the color codes $(c_k^\mathrm{n}, c_k^\mathrm{w}, c_k^\mathrm{s}, c_k^\mathrm{e})$, with $c_k^\mathrm{n}, c_k^\mathrm{w}, c_k^\mathrm{s}$, and $c_k^\mathrm{e} \in \mathcal{C}$ standing for the color codes of the north, west, south, and east edge of the tile k, respectively. Tiles can, therefore, be represented graphically as non-rotatable squares shown in Fig. A.2. Without loss of generality, we further consider these squares to be of the unit size.

$$(c_k^\mathrm{n}, c_k^\mathrm{w}, c_k^\mathrm{s}, c_k^\mathrm{e}) \;\Leftrightarrow\; \boxed{\begin{array}{ccc} & c_k^\mathrm{n} & \\ c_k^\mathrm{w} & & c_k^\mathrm{e} \\ & c_k^\mathrm{s} & \end{array}}$$

Figure A.2: Graphical representation of a Wang tile k.

A *tile set* $\mathcal{T}$ represents a finite collection of n_t tiles, see Fig. A.3. When $\forall (c^\mathrm{n}, c^\mathrm{w}, c^\mathrm{s}, c^\mathrm{e}) \in \mathcal{C}^4 : (c^\mathrm{n}, c^\mathrm{w}, c^\mathrm{s}, c^\mathrm{e}) \in \mathcal{T}$, we call the tile set *complete*.

$$\mathcal{T} = \left\{ \boxed{\begin{array}{ccc} & c_1^\mathrm{n} & \\ c_1^\mathrm{w} & & c_1^\mathrm{e} \\ & c_1^\mathrm{s} & \end{array}}, \boxed{\begin{array}{ccc} & c_2^\mathrm{n} & \\ c_2^\mathrm{w} & & c_2^\mathrm{e} \\ & c_2^\mathrm{s} & \end{array}}, \ldots, \boxed{\begin{array}{ccc} & c_{n_\mathrm{t}}^\mathrm{n} & \\ c_{n_\mathrm{t}}^\mathrm{w} & & c_{n_\mathrm{t}}^\mathrm{e} \\ & c_{n_\mathrm{t}}^\mathrm{s} & \end{array}} \right\}$$

Figure A.3: A tile set $\mathcal{T}$.

Using the notation $\tilde{\bullet} = \bullet \cap \mathbb{N}^2$ to denote an intersection of the set $\bullet$ with the integer lattice points, *tiling* $\mathfrak{T}^\mathcal{A}$ of a bounded domain $\mathcal{A} \supset \tilde{\mathcal{A}} = \mathcal{H} \times \mathcal{W}$, where $\mathcal{H} \subset \mathbb{N}$ and $\mathcal{W} \subset \mathbb{N}$ denote the set of vertical (height) and horizontal (width) coordinates, constitutes an arrangement of copies of the tiles from the tile set $\mathcal{T}$ such that the tiles are placed at $\tilde{\mathcal{A}}$, they are contiguous, do not overlap, and cover the entire domain, cf. Fig. A.4. More formally, tiling is a mapping $M : \tilde{\mathcal{A}} \to \mathcal{T}$ assigning a single tile $k \in \mathcal{T}$ to every $(i, j) \in \tilde{\mathcal{A}}$ coordinate. Consequently, we call tilings $\mathfrak{T}^\mathcal{A}$ *simply connected* iff the domain $\mathcal{A}$ is so.

The tiling $\mathfrak{T}^\mathcal{A}$ is *rectangular* if $\forall i \in \{1, \ldots, n_\mathrm{h}\}, \forall j \in \{1, \ldots, n_\mathrm{w}\}$ it holds that $(i, j) \in \tilde{\mathcal{A}}$.

A *valid tiling* (Wang tiling) of $\mathcal{A}$, denoted by $\mathfrak{T}_\mathrm{v}^\mathcal{A}$, is a tiling with equal color codes at the shared edges between all pairs of adjoining tiles. Therefore, the mapping $M_\mathrm{v} : \tilde{\mathcal{A}} \to \mathcal{T}$ satisfies, in addition to the requirements for M, the additional constraints

$$c^\mathrm{s}_{M_\mathrm{v}(i,j)} = c^\mathrm{n}_{M_\mathrm{v}(i+1,j)}, \quad \forall i \in \mathcal{H} \setminus \{n_\mathrm{h}\}, \forall j \in \mathcal{W}, \tag{A.1a}$$

$$c^\mathrm{e}_{M_\mathrm{v}(i,j)} = c^\mathrm{w}_{M_\mathrm{v}(i,j+1)}, \quad \forall i \in \mathcal{H}, \forall j \in \mathcal{W} \setminus \{n_\mathrm{w}\}, \tag{A.1b}$$

provided that the axes are oriented accordingly to Fig. A.4. If M_v exists, we say that the domain $\mathcal{A}$ admits a valid $\mathcal{T}$-tiling, or that it is *tileable* by $\mathcal{T}$.

Consider that $\mathcal{B} \subseteq \mathcal{A}$ and $\mathcal{B}_{\max\,\mathrm{rect}} \subseteq \mathcal{A}$ are simply connected, rectangular, and $\mathcal{T}$-tileable. Then, the *maximum rectangular valid tiling* $\mathfrak{T}^\mathcal{A}_{\mathrm{v},\max\,\mathrm{rect}}$ is a valid tiling of the domain $\mathcal{B}_{\max\,\mathrm{rect}}$,

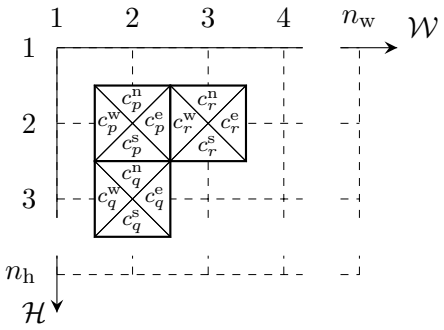


Figure A.4: Color matching among tiles p, q, and $r \in \mathcal{T}$.

where $\{\mathcal{B}_{\max\,\mathrm{rect}} \subseteq \mathcal{A}, \forall \mathcal{B} \subseteq \mathcal{A} : |\tilde{\mathcal{B}}_{\max\,\mathrm{rect}}| \geq |\tilde{\mathcal{B}}|\}$. Here, the notation $|\bullet|$ denotes cardinality of the set $\bullet$.

The *maximum cover* $\mathfrak{T}^{\mathcal{A}}_{\mathrm{v},\max\,\mathrm{cov}}$ is a valid tiling of $\mathcal{B}_{\max\,\mathrm{cov}}$, where $\mathcal{B}$ and $\mathcal{B}_{\max\,\mathrm{cov}}$ are arbitrary $\mathcal{T}$-tileable subdomains in $\mathcal{A}$ and $\{\mathcal{B}_{\max\,\mathrm{cov}} \subseteq \mathcal{A}, \forall \mathcal{B} \subseteq \mathcal{A} : |\tilde{\mathcal{B}}_{\max\,\mathrm{cov}}| \geq |\tilde{\mathcal{B}}|\}$.

A rectangular valid tiling is said to be *periodic*, if the color codes at the opposite sides of the rectangle match. If the valid tiling is not periodic, but the considered tile set allows for at least one periodic pattern to emerge, we call it *nonperiodic*. Finally, if no such periodic pattern exists and the tile set still allows for a valid tiling of the infinite plane, it is referred to as *aperiodic*. Similarly, the tile set $\mathcal{T}$ is *periodic* if it permits periodic valid tilings; and *aperiodic* if all feasible valid tilings are aperiodic.

Transducer graph[98] $G_{\mathrm{t,h}}$ of the tile set $\mathcal{T}$ is a directed (multi-)graph representation of a Mealy machine without any initial nor terminal state. It consists of $|\mathcal{C}|$ states (graph vertices) and $|\mathcal{T}|$ transitions (directed edges) $\mathcal{E}_\mathrm{h}$, where

$$\mathcal{E}_\mathrm{h} \coloneqq \bigcup_{k \in \mathcal{T}} \left(c_k^\mathrm{w} \xrightarrow{c_k^\mathrm{s} | c_k^\mathrm{n}} c_k^\mathrm{e} \right). \tag{A.2}$$

For the *dual transducer graph* $G_{\mathrm{t,v}}$, composed of the dual Wang tiles [117] reflecting $\mathcal{T}$ along the major diagonal of the tiles, the edge set is defined as

$$\mathcal{E}_\mathrm{v} \coloneqq \bigcup_{k \in \mathcal{T}} \left(c_k^\mathrm{n} \xrightarrow{c_k^\mathrm{e} | c_k^\mathrm{w}} c_k^\mathrm{s} \right). \tag{A.3}$$

To illustrate the construction, we include a visual example in Fig. A.5.

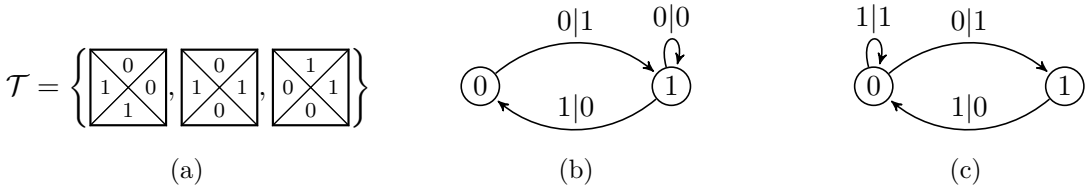


Figure A.5: (b) Transducer and (c) dual transducer graphs of the tile set (a).

A.3 Integer programming formulations

In this section, we introduce four integer programming formulations for the generation of valid tilings. The first one, in Section A.3.1, develops a decision variant. In the later sections, we investigate the maximum rectangular tiling (Section A.3.2), maximum cover (Section A.3.3), and the maximum adjacency constraints satisfaction (Section A.3.4). Finally, Section A.3.5 proposes several extensions to facilitate finer control over the resulting tilings.

A.3.1 Rectangular valid tiling

Let us now consider the fundamental problem of finding $\mathfrak{T}_{\mathrm{v}}^{\mathcal{A}}$ or proving it does not exist. From now on, we restrict $\mathcal{A}$ to be rectangular to simplify notation. However, the presented approach also extends to the general unrestricted case.

To this goal, we introduce $\forall(i,j,k) \in \mathcal{H} \times \mathcal{W} \times \mathcal{T}$ a binary decision variable $x_{i,j,k} \in \{0,1\}$ denoting the placement of the tile k at the (i,j) coordinate such that

$$x_{i,j,k} = \begin{cases} 1 & \text{iff the tile } k \text{ lies at coordinate } (i,j), \\ 0 & \text{otherwise.} \end{cases} \tag{A.4}$$

Consequently, the mapping $M(i,j)$ is expressed as

$$M(i,j) = \sum_{k \in \mathcal{T}} k x_{i,j,k}, \tag{A.5}$$

together with the requirement that every (i,j) coordinate is occupied by one tile,

$$\sum_{k \in \mathcal{T}} x_{i,j,k} = 1, \quad \forall(i,j) \in \mathcal{H} \times \mathcal{W}. \tag{A.6}$$

Similarly, the color codes of a tile placed at (i,j) are expressed using the binary variables as

$$c^{\mathrm{n}}_{M(i,j)} = \sum_{k \in \mathcal{T}} c^{\mathrm{n}}_k x_{i,j,k}, \tag{A.7a}$$

$$c^{\mathrm{w}}_{M(i,j)} = \sum_{k \in \mathcal{T}} c^{\mathrm{w}}_k x_{i,j,k}, \tag{A.7b}$$

$$c^{\mathrm{s}}_{M(i,j)} = \sum_{k \in \mathcal{T}} c^{\mathrm{s}}_k x_{i,j,k}, \tag{A.7c}$$

$$c^{\mathrm{e}}_{M(i,j)} = \sum_{k \in \mathcal{T}} c^{\mathrm{e}}_k x_{i,j,k}. \tag{A.7d}$$

Inserting (A.7) into (A.1a) and (A.1b) leads to the horizontal and vertical adjacency constraints expressed in terms of the decision variables, hence

$$\sum_{k \in \mathcal{T}} c^{\mathrm{s}}_k x_{i,j,k} - \sum_{k \in \mathcal{T}} c^{\mathrm{n}}_k x_{i+1,j,k} = 0, \quad \forall(i,j) \in \mathcal{H} \setminus \{n_{\mathrm{h}}\} \times \mathcal{W}, \tag{A.8a}$$

$$\sum_{k \in \mathcal{T}} c^{\mathrm{e}}_k x_{i,j,k} - \sum_{k \in \mathcal{T}} c^{\mathrm{w}}_k x_{i,j+1,k} = 0, \quad \forall(i,j) \in \mathcal{H} \times \mathcal{W} \setminus \{n_{\mathrm{w}}\}. \tag{A.8b}$$

Combining (A.4), (A.5), (A.6), and (A.8) then provides us with a complete binary linear programming representation of M_{v}.

For computational reasons, it proved to be advantageous to consider a tighter representation of the feasible design space and organize the constraints according to the color codes:

$$\sum_{k\in\mathcal{T}} x_{i,j,k}[c_k^{\mathrm{s}} = \ell] - \sum_{k\in\mathcal{T}} x_{i+1,j,k}[c_k^{\mathrm{n}} = \ell] = 0, \quad \forall (i,j,\ell) \in \mathcal{H}\setminus\{n_{\mathrm{h}}\}\times\mathcal{W}\times\mathcal{C}, \tag{A.9a}$$

$$\sum_{k\in\mathcal{T}} x_{i,j,k}[c_k^{\mathrm{e}} = \ell] - \sum_{k\in\mathcal{T}} x_{i,j+1,k}[c_k^{\mathrm{w}} = \ell] = 0, \quad \forall (i,j,\ell) \in \mathcal{H}\times\mathcal{W}\setminus\{n_{\mathrm{w}}\}\times\mathcal{C}, \tag{A.9b}$$

where the Iverson notation [104] $\sum_{k\in\mathcal{T}} x_{i,j,k}\,[c_k^{\mathrm{s}} = \ell]$ expresses that $x_{i,j,k}$ is added to the sum if and only if $c_k^{\mathrm{s}} = \ell$.

The constraint (A.9a) requires that the number of tiles at (i,j) with the south edge colored by ℓ equals to the number of tiles at $(i+1,j)$ with the north edge marked by the same ℓ, for all $\ell \in \mathcal{C}$. Because of (A.6) there are either no tiles with the shared edge colored by ℓ, or a single tile at each of the coordinates with its common edge labeled by ℓ. Analogously to the vertical adjacency constraint, also the horizontal constraint (A.9b) enforces equality among the number of tiles at (i,j) with the east edge colored by ℓ and the number of tiles at $(i,j+1)$ having the west edge colored by identical ℓ.

Finally, combining (A.4), (A.6), and (A.9), while noticing that the constraints (A.6) naturally propagate with the adjacency constraints from the domain boundaries (compare (A.10d,A.10e with (A.6)), leads to the binary programming formulation

$$\text{find } \mathbf{x} \tag{A.10a}$$

$$\text{s.t.} \sum_{k\in\mathcal{T}} x_{i,j,k}[c_k^{\mathrm{s}} = \ell] - \sum_{k\in\mathcal{T}} x_{i+1,j,k}[c_k^{\mathrm{n}} = \ell] = 0, \quad \forall (i,j,\ell) \in \mathcal{H}\setminus\{n_{\mathrm{h}}\}\times\mathcal{W}\times\mathcal{C}, \tag{A.10b}$$

$$\sum_{k\in\mathcal{T}} x_{i,j,k}[c_k^{\mathrm{e}} = \ell] - \sum_{k\in\mathcal{T}} x_{i,j+1,k}[c_k^{\mathrm{w}} = \ell] = 0, \quad \forall (i,j,\ell) \in \mathcal{H}\times\mathcal{W}\setminus\{n_{\mathrm{w}}\}\times\mathcal{C}, \tag{A.10c}$$

$$\sum_{k\in\mathcal{T}} x_{i,j,k} = 1, \quad \forall (i,j) \in \{1,n_{\mathrm{h}}\}\times\mathcal{W}, \tag{A.10d}$$

$$\sum_{k\in\mathcal{T}} x_{i,j,k} = 1, \quad \forall (i,j) \in \mathcal{H}\times\{1,n_{\mathrm{w}}\}, \tag{A.10e}$$

$$x_{i,j,k} \in \{0,1\}, \quad \forall (i,j,k) \in \mathcal{H}\times\mathcal{W}\times\mathcal{T}, \tag{A.10f}$$

that provides a complete representation of the bounded tiling problem, i.e., all valid tilings solve the integer program, and conversely, all feasible solutions to (A.10) are valid tilings. Moreover, observe that the problem consists of two totally unimodular constraints if considered independently: (A.10c,A.10e) representing row tilings, and (A.10b,A.10d) being column tilings. When considered simultaneously, the resulting problem becomes $\mathcal{NP}$-complete [125, 124].

A.3.2 Maximum rectangular valid tiling

In certain cases, when solution to (A.10) cannot be found in acceptable time or when no such solution exists, one can resort to relaxing the requirement of a valid tiling of $\mathcal{A}$ and search for a valid tiling of the as large rectangular subdomain as possible. Clearly, the most notable such subdomain is the largest one, as its valid tiling also solves (A.10) in the case that $\mathcal{A}$ is $\mathcal{T}$-tileable.

Without loss of generality, let us assume that the maximum rectangular valid tiling always contains an anchor tile placed at $(1, 1)$, i.e.,

$$\sum_{k \in \mathcal{T}} x_{1,1,k} = 1. \tag{A.11}$$

On the other hand, all the other coordinates may contain a tile, or be empty, thus

$$\sum_{k \in \mathcal{T}} x_{i,j,k} \leq 1, \quad \forall (i, j) \in \tilde{\mathcal{A}} \setminus (1, 1). \tag{A.12}$$

Let us now pick two vertically adjacent coordinates (i, j) and $(i + 1, j)$. If there is a tile q placed at $(i + 1, j)$, another tile p has to be placed at (i, j), as, otherwise, there is no simply-connected rectangular tiling containing both the tiles at $(1, 1)$ and at $(i+1, j)$. Validity of the tiling further requires identical color codes at the shared edges. On the other hand, if no tile is placed at $(i + 1, j)$, a coordinate (i, j) may be either occupied, or empty. The allowed and forbidden combinations are shown in Fig. A.6a–A.6d. Formally stated in terms of the decision variables, these considerations are expressed as

$$\sum_{k \in \mathcal{T}} x_{i,j,k}[c^{\mathrm{s}}_k = \ell] - \sum_{k \in \mathcal{T}} x_{i+1,j,k}[c^{\mathrm{n}}_k = \ell] \geq 0, \quad \forall (i, j, \ell) \in \mathcal{H} \setminus \{n_{\mathrm{h}}\} \times \mathcal{W} \times \mathcal{C}. \tag{A.13}$$

Similar considerations also apply to the case of the coordinates (i, j) and $(i, j+1)$, resulting in the constraints

$$\sum_{k \in \mathcal{T}} x_{i,j,k}[c^{\mathrm{e}}_k = \ell] - \sum_{k \in \mathcal{T}} x_{i,j+1,k}[c^{\mathrm{w}}_k = \ell] \geq 0, \quad \forall (i, j, \ell) \in \mathcal{H} \times \mathcal{W} \setminus \{n_{\mathrm{w}}\} \times \mathcal{C}. \tag{A.14}$$

For the graphical representation of the allowed and forbidden combinations, see Fig. A.6e–A.6h.

The developed constraints (A.11)–(A.14) enforce simple connectedness; however, they do not guarantee that the resulting tiling will be rectangular. For any 4 adjacent tiles p, q, r,

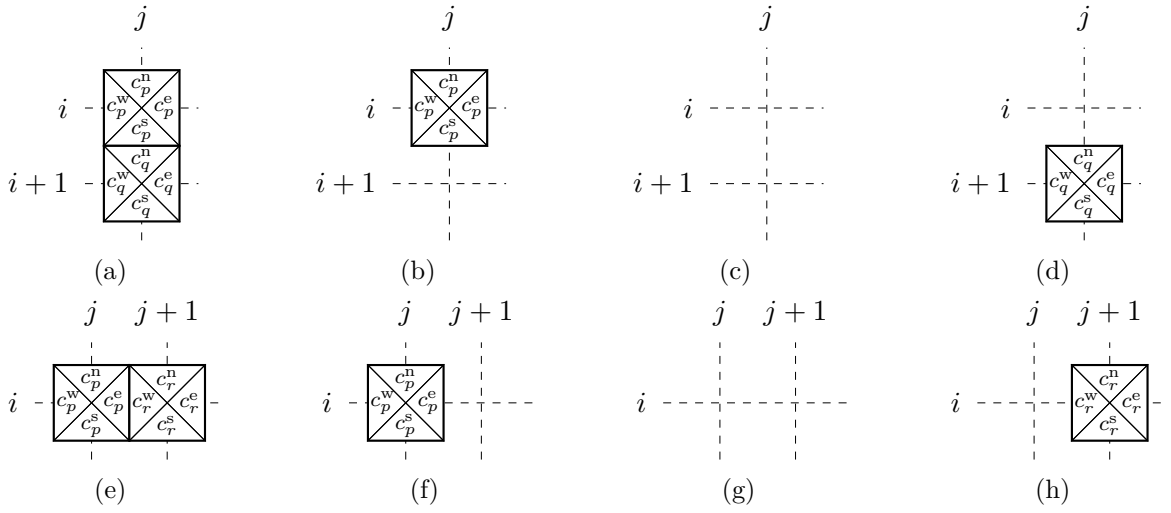


Figure A.6: Admissible tile placements (a)–(c) and (e)–(g), and forbidden placements (d) and (h) in the maximum rectangular valid tiling formulation.

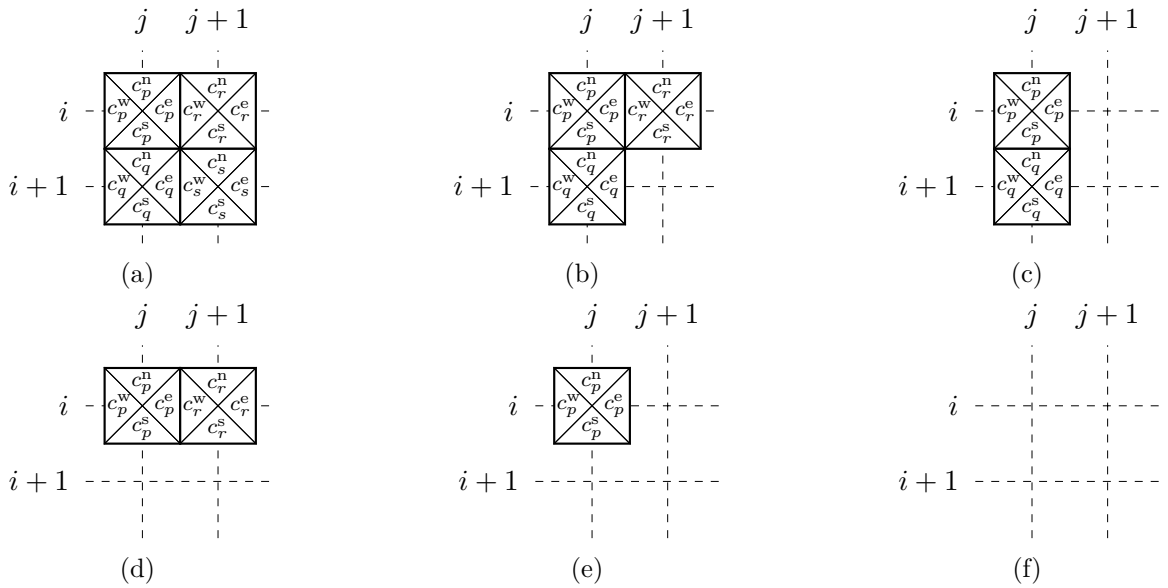


Figure A.7: Possible placement of tiles p, q, r, and s. The combination (b) cannot appear in any rectangular tiling.

and s placed at (i, j), $(i + 1, j)$, $(i, j + 1)$, and at $(i + 1, j + 1)$, respectively, these constraints allow for the assemblies shown in Fig. A.7. Because the combination A.7b cannot appear in any simply-connected rectangular tiling, we must exclude it from the feasible set,

$$\sum_{k\in\mathcal{T}} x_{i+1,j,k} + \sum_{k\in\mathcal{T}} x_{i,j+1,k} - \sum_{k\in\mathcal{T}} x_{i+1,j+1,k} \leq 1, \quad \forall (i,j) \in \mathcal{H} \setminus \{n_\mathrm{h}\} \times \mathcal{W} \setminus \{n_\mathrm{w}\}. \tag{A.15}$$

Finally, combining Eqs. (A.4), (A.11), (A.12), (A.13), (A.14), and (A.15) together with an objective function to maximize $|\tilde{\mathcal{B}}_{\max\,\mathrm{rect}}|$ provides us with the binary optimization program

$$\max_{\mathbf{x}} \sum_{i\in\mathcal{H}} \sum_{j\in\mathcal{W}} \sum_{k\in\mathcal{T}} x_{i,j,k} \tag{A.16a}$$

$$\text{s.t.} \sum_{k\in\mathcal{T}} x_{i,j,k}[c^{\mathrm{s}}_k = \ell] - \sum_{k\in\mathcal{T}} x_{i+1,j,k}[c^{\mathrm{n}}_k = \ell] \geq 0, \quad \forall (i,j,\ell) \in \mathcal{H} \setminus \{n_\mathrm{h}\} \times \mathcal{W} \times \mathcal{C}, \tag{A.16b}$$

$$\sum_{k\in\mathcal{T}} x_{i,j,k}[c^{\mathrm{e}}_k = \ell] - \sum_{k\in\mathcal{T}} x_{i,j+1,k}[c^{\mathrm{w}}_k = \ell] \geq 0, \quad \forall (i,j,\ell) \in \mathcal{H} \times \mathcal{W} \setminus \{n_\mathrm{w}\} \times \mathcal{C}, \tag{A.16c}$$

$$\sum_{k\in\mathcal{T}} x_{i+1,j,k} + \sum_{k\in\mathcal{T}} x_{i,j+1,k} - \sum_{k\in\mathcal{T}} x_{i+1,j+1,k} \leq 1, \ \forall (i,j) \in \mathcal{H} \setminus \{n_\mathrm{h}\} \times \mathcal{W} \setminus \{n_\mathrm{w}\}, \tag{A.16d}$$

$$\sum_{k\in\mathcal{T}} x_{1,1,k} = 1, \tag{A.16e}$$

$$\sum_{k\in\mathcal{T}} x_{i,j,k} \leq 1, \quad \forall (i,j) \in \tilde{\mathcal{A}} \setminus (1,1), \tag{A.16f}$$

$$x_{i,j,k} \in \{0,1\}, \quad \forall (i,j,k) \in \mathcal{H} \times \mathcal{W} \times \mathcal{T}. \tag{A.16g}$$

In contrast to (A.10), a feasible solution to the optimization program (A.16) can be found in a polynomial time, e.g., by tiling the first row or column of the one-dimensional bounded

tiling problem. However, obtaining an optimal solution to (A.16) is $\mathcal{NP}$-hard, because the optimization problem (A.16) is reducible to the decision version (A.10) by fixing the value of the objective function to $|\tilde{\mathcal{A}}|$. Indeed, this objective function value enforces equalities in (A.16b), (A.16c), and (A.16f), making consequently the constraint (A.16d) always satisfied, and thus redundant.

A.3.3 Maximum cover

Another option to avoid the infeasibility of (A.10) rests in neglecting the requirement of (simple) connectedness, hence allowing for a placement of empty tiles (voids). In this section, we therefore search the maximum cover of $\mathcal{A}$, or equivalently a valid tiling of the (possibly disconnected) domain $\mathcal{B}_{\max\,\mathrm{cov}} \subseteq \mathcal{A}$.

For the maximum cover formulation, we assume that any two adjacent tiles satisfy the edge-matching constraints of valid tilings, but these are also satisfied by any of the tile-void, void-tile, or void-void combination, where $\sum_{k\in\mathcal{T}} x_{i,j,k} = 0$ for a void located at $(i,j) \in \tilde{\mathcal{A}}$.

Thus, each coordinate (i,j) is occupied either by a tile or a void, implying that

$$\sum_{k\in\mathcal{T}} x_{i,j,k} \leq 1, \quad \forall (i,j) \in \mathcal{H}\times\mathcal{W}, \tag{A.17}$$

and the vertical and horizontal edge matching conditions become

$$\sum_{k\in\mathcal{T}} x_{i,j,k}[c_k^{\mathrm{e}} = \ell] + \sum_{k\in\mathcal{T}} x_{i,j+1,k}[c_k^{\mathrm{w}} \neq \ell] \leq 1, \quad \forall (i,j,\ell) \in \mathcal{H}\times\mathcal{W}\setminus\{n_{\mathrm{w}}\}\times\mathcal{C}, \tag{A.18a}$$

$$\sum_{k\in\mathcal{T}} x_{i,j,k}[c_k^{\mathrm{s}} = \ell] + \sum_{k\in\mathcal{T}} x_{i+1,j,k}[c_k^{\mathrm{n}} \neq \ell] \leq 1, \quad \forall (i,j,\ell) \in \mathcal{H}\setminus\{n_{\mathrm{h}}\}\times\mathcal{W}\times\mathcal{C}. \tag{A.18b}$$

Finally, the combination of Eqs. (A.17), (A.18a), (A.18b) with the objective function to maximize $|\tilde{\mathcal{B}}_{\max\,\mathrm{cov}}|$ leads to the binary optimization problem

$$\max_{\mathbf{x}} \sum_{i\in\mathcal{H}}\sum_{j\in\mathcal{W}}\sum_{k\in\mathcal{T}} x_{i,j,k} \tag{A.19a}$$

$$\text{s.t.} \sum_{k\in\mathcal{T}} x_{i,j,k}[c_k^{\mathrm{e}} = \ell] + \sum_{k\in\mathcal{T}} x_{i,j+1,k}[c_k^{\mathrm{w}} \neq \ell] \leq 1, \quad \forall (i,j,\ell) \in \mathcal{H}\times\mathcal{W}\setminus\{n_{\mathrm{w}}\}\times\mathcal{C}, \tag{A.19b}$$

$$\sum_{k\in\mathcal{T}} x_{i,j,k}[c_k^{\mathrm{s}} = \ell] + \sum_{k\in\mathcal{T}} x_{i+1,j,k}[c_k^{\mathrm{n}} \neq \ell] \leq 1, \quad \forall (i,j,\ell) \in \mathcal{H}\setminus\{n_{\mathrm{h}}\}\times\mathcal{W}\times\mathcal{C}, \tag{A.19c}$$

$$\sum_{k\in\mathcal{T}} x_{i,j,k} \leq 1, \quad \forall (i,j) \in \mathcal{H}\times\mathcal{W}, \tag{A.19d}$$

$$x_{i,j,k} \in \{0,1\}, \quad \forall (i,j,k) \in \mathcal{H}\times\mathcal{W}\times\mathcal{T}. \tag{A.19e}$$

The program (A.19) is trivially $\mathcal{NP}$-hard: Requiring the objective function (A.19a) to be at least $|\tilde{\mathcal{A}}|$ implies that

$$\sum_{k\in\mathcal{T}} x_{i,j,k} = 1, \quad \forall (i,j) \in \mathcal{H}\times\mathcal{W}, \tag{A.20}$$

i.e., all positions are occupied by a Wang tile. Moreover, (A.19b) and (A.19c) require for all adjacent tiles to share the color codes at their common edges. Consequently, the resulting tiling is void-free and valid, and solves the $\mathcal{NP}$-complete bounded tiling problem.

A.3.4 Maximum adjacency constraints satisfaction

Because the decision problem (A.10) also constitutes a specific instance of the constraint satisfaction problem (CSP), another optimization variant comes from the formulation of the max-CSP problem, maximizing the number of satisfied clauses, color matches in our case.

Therefore, for each vertical and horizontal edge we introduce a new variable $h^{\mathrm{v}}_{i,j} \in \mathbb{R}_{\geq 0}$, where $(i,j) \in \mathcal{H} \times \mathcal{W} \setminus n_{\mathrm{w}}$, and $h^{\mathrm{h}}_{i,j} \in \mathbb{R}_{\geq 0}$, with $(i,j) \in \mathcal{H} \setminus n_{\mathrm{h}} \times \mathcal{W}$, respectively. The adjacency constraints (A.9) are then relaxed by considering

$$\left| \sum_{k \in \mathcal{T}} x_{i,j,k}[c^{\mathrm{s}}_k = \ell] - \sum_{k \in \mathcal{T}} x_{i+1,j,k}[c^{\mathrm{n}}_k = \ell] \right| \leq h^{\mathrm{h}}_{i,j}, \quad \forall (i,j,\ell) \in \mathcal{H} \setminus \{n_{\mathrm{h}}\} \times \mathcal{W} \times \mathcal{C}, \tag{A.21a}$$

$$\left| \sum_{k \in \mathcal{T}} x_{i,j,k}[c^{\mathrm{e}}_k = \ell] - \sum_{k \in \mathcal{T}} x_{i,j+1,k}[c^{\mathrm{w}}_k = \ell] \right| \leq h^{\mathrm{v}}_{i,j}, \quad \forall (i,j,\ell) \in \mathcal{H} \times \mathcal{W} \setminus \{n_{\mathrm{w}}\} \times \mathcal{C} \tag{A.21b}$$

instead. Indeed, if $h^{\mathrm{h}}_{i,j} = 0$, the edge-matching requirement of the neighboring tiles at (i,j) and $(i+1,j)$ is satisfied; and it is violated otherwise. Similarly, $h^{\mathrm{v}}_{i,j} = 0$ guarantees color matches among the tiles at (i,j) and $(i,j+1)$.

Finally, rewriting (A.21) without absolute values while supplying an objective function to maximize the number of color matches yields the binary optimization problem

$$\max_{\mathbf{x}} \sum_{i \in \mathcal{H}} \sum_{j \in \mathcal{W} \setminus n_{\mathrm{w}}} \left(1 - h^{\mathrm{v}}_{i,j}\right) + \sum_{i \in \mathcal{H} \setminus n_{\mathrm{h}}} \sum_{j \in \mathcal{W}} \left(1 - h^{\mathrm{h}}_{i,j}\right) \tag{A.22a}$$

$$\text{s.t.} \sum_{k \in \mathcal{T}} x_{i,j,k}[c^{\mathrm{s}}_k = \ell] - \sum_{k \in \mathcal{T}} x_{i+1,j,k}[c^{\mathrm{n}}_k = \ell] \leq h^{\mathrm{h}}_{i,j}, \quad \forall (i,j,\ell) \in \mathcal{H} \setminus \{n_{\mathrm{h}}\} \times \mathcal{W} \times \mathcal{C}, \tag{A.22b}$$

$$\sum_{k \in \mathcal{T}} x_{i+1,j,k}[c^{\mathrm{n}}_k = \ell] - \sum_{k \in \mathcal{T}} x_{i,j,k}[c^{\mathrm{s}}_k = \ell] \leq h^{\mathrm{h}}_{i,j}, \quad \forall (i,j,\ell) \in \mathcal{H} \setminus \{n_{\mathrm{h}}\} \times \mathcal{W} \times \mathcal{C}, \tag{A.22c}$$

$$\sum_{k \in \mathcal{T}} x_{i,j,k}[c^{\mathrm{e}}_k = \ell] - \sum_{k \in \mathcal{T}} x_{i,j+1,k}[c^{\mathrm{w}}_k = \ell] \leq h^{\mathrm{v}}_{i,j}, \quad \forall (i,j,\ell) \in \mathcal{H} \times \mathcal{W} \setminus \{n_{\mathrm{w}}\} \times \mathcal{C}, \tag{A.22d}$$

$$\sum_{k \in \mathcal{T}} x_{i,j+1,k}[c^{\mathrm{w}}_k = \ell] - \sum_{k \in \mathcal{T}} x_{i,j,k}[c^{\mathrm{e}}_k = \ell] \leq h^{\mathrm{v}}_{i,j}, \quad \forall (i,j,\ell) \in \mathcal{H} \times \mathcal{W} \setminus \{n_{\mathrm{w}}\} \times \mathcal{C}, \tag{A.22e}$$

$$\sum_{k \in \mathcal{T}} x_{i,j,k} = 1, \quad \forall (i,j) \in \mathcal{H} \times \mathcal{W}, \tag{A.22f}$$

$$x_{i,j,k} \in \{0,1\}, \quad \forall (i,j,k) \in \mathcal{H} \times \mathcal{W} \times \mathcal{T}, \tag{A.22g}$$

that is $\mathcal{NP}$-hard due to the reduction to (A.10) after fixing all $h^{\mathrm{v}}_{i,j}$ and $h^{\mathrm{h}}_{i,j}$ to zeros. A simple feasible solution can be found in a polynomial time by finding valid row/column tilings for each row/column, so that either the term $\sum_{i \in \mathcal{H}} \sum_{j \in \mathcal{W} \setminus n_{\mathrm{w}}} h^{\mathrm{v}}_{i,j}$, or $\sum_{i \in \mathcal{H} \setminus n_{\mathrm{h}}} \sum_{j \in \mathcal{W}} h^{\mathrm{h}}_{i,j}$ equals zero.

A.3.5 Extensions

Until now, we have focused solely on the (re)formulations of the bounded tiling problem, searching for *arbitrary* valid tilings. However, it may be of interest for potential applications to have more detailed control over the resulting tilings. Thus, in this section, we state some simple extensions to enforce tile- and color-based boundary conditions, to solve the tile packing problem [120], and to enforce (variable-sized) periodic boundary conditions.

A.3.5.1 Tile-based boundary conditions

At first, we consider boundary conditions in the form of prescribed tiles. As the simplest one, we enforce the placement of a tile k at (i, j):

$$x_{i,j,k} = 1, \quad (i, j, k) \in \mathcal{H} \times \mathcal{W} \times \mathcal{T}. \tag{A.23}$$

Similarly, we may prevent the tile k to be placed there:

$$x_{i,j,k} = 0, \quad (i, j, k) \in \mathcal{H} \times \mathcal{W} \times \mathcal{T}. \tag{A.24}$$

Placement of an identical tile at the coordinates $(i, j) \in \tilde{\mathcal{A}}$ and $(p, q) \in \tilde{\mathcal{A}}$ requires

$$x_{i,j,k} - x_{p,q,k} = 0, \quad \{i, p\} \in \mathcal{H}, \{j, q\} \in \mathcal{W}, \forall k \in \mathcal{T}. \tag{A.25}$$

Conversely, different tiles at these coordinates are secured with

$$x_{i,j,k} + x_{p,q,k} \leq 1, \quad \{i, p\} \in \mathcal{H}, \{j, q\} \in \mathcal{W}, \forall k \in \mathcal{T}. \tag{A.26}$$

A.3.5.2 Color-based boundary conditions

In addition to the tile-based constraints, we may also enforce specific color codes of individual edges. To this goal, color of the north edge at $(i, j) \in \tilde{\mathcal{A}}$ is set to ℓ by

$$\sum_{k \in T} x_{i,j,k}[n_k = \ell] = 1, \quad (i, j, \ell) \in \mathcal{H} \times \mathcal{W} \times \mathcal{C}. \tag{A.27}$$

On the contrary, we may prevent this color by using

$$\sum_{k \in T} x_{i,j,k}[n_k = \ell] = 0, \quad (i, j, \ell) \in \mathcal{H} \times \mathcal{W} \times \mathcal{C}. \tag{A.28}$$

Further, the same color codes at the north edge of $(i, j) \in \tilde{\mathcal{A}}$ and at the west edge of $(p, q) \in \tilde{\mathcal{A}}$ are established by

$$\sum_{k \in T} x_{i,j,k}[n_k = \ell] - \sum_{k \in T} x_{p,q,k}[w_k = \ell] = 0, \quad \{i, p\} \in \mathcal{H}, \{j, q\} \in \mathcal{W}, \forall \ell \in \mathcal{C}, \tag{A.29}$$

and a different color with

$$\sum_{k \in T} x_{i,j,k}[n_k = \ell] + \sum_{k \in T} x_{p,q,k}[w_k = \ell] \leq 1, \quad \{i, p\} \in \mathcal{H}, \{j, q\} \in \mathcal{W}, \forall \ell \in \mathcal{C}. \tag{A.30}$$

A.3.5.3 Periodic tiling

In the domino problem, Wang [202] investigated existence of tile sets admitting an infinite aperiodic tilings. Here, we consider a similar setting for the finite domain $\mathcal{A}$: examining periodicity through periodic color-based boundary conditions.

We begin with a simpler setting of requiring equal coloring at the fixed opposite domain boundaries,

$$\sum_{k \in T} x_{1,j,k}[n_k = \ell] - \sum_{k \in T} x_{n_{t,h},j,k}[s_k = \ell] = 0, \quad \forall (j, \ell) \in \mathcal{W} \times \mathcal{C}, \tag{A.31a}$$

$$\sum_{k \in T} x_{i,1,k}[w_k = \ell] - \sum_{k \in T} x_{i,n_{t,w},k}[e_k = \ell] = 0, \quad \forall (i, \ell) \in \mathcal{H} \times \mathcal{C}. \tag{A.31b}$$

When adding (A.31) to the decision problem (A.10), we thus ask for an existence of a fixed-sized periodic Wang tiling.

In a natural generalization, we ask for an existence of bounded-sized periodic Wang tilings, relying so on the maximum rectangular valid tiling formulation (A.16). Naturally, the boundaries are not known in this case. Therefore, we must consider the constraints of the form

$$\sum_{k\in\mathcal{T}} x_{i,j,k}[e_k \neq \ell] + \sum_{k\in\mathcal{T}} x_{i,1,k}[w_k = \ell] - \sum_{k\in\mathcal{T}} x_{i,j+1,k}[j < n_{\mathrm{t},w}] \leq 1, \quad \forall (i,j,\ell) \in \mathcal{H}\times\mathcal{W}\times\mathcal{C}, \tag{A.32a}$$

$$\sum_{k\in\mathcal{T}} x_{i,j,k}[s_k \neq \ell] + \sum_{k\in\mathcal{T}} x_{1,j,k}[n_k = \ell] - \sum_{k\in\mathcal{T}} x_{i+1,j,k}[i < n_{\mathrm{t},h}] \leq 1, \quad \forall (i,j,\ell) \in \mathcal{H}\times\mathcal{W}\times\mathcal{C}. \tag{A.32b}$$

Here, (A.32a) prevents a color mismatch of the north edge of $(1,j) \in \tilde{\mathcal{A}}$ and the south edge of $(i,j) \in \mathcal{A}$ iff there is no tile placed at $(i,j+1) \in \tilde{\mathcal{A}}$. Similarly, in the case of (A.32b), we prevent a color mismatch of the west edge at $(i,1) \in \tilde{\mathcal{A}}$ and the east edge at $(i,j) \in \tilde{\mathcal{A}}$ iff the position $(i+1,j) \in \tilde{\mathcal{A}}$ is empty.

Finally, when adding the constraints (A.32) to (A.16), we usually search for the smallest periodic pattern rather than the largest,

$$\min_{\mathbf{x}} \sum_{i\in\mathcal{H}}\sum_{j\in\mathcal{W}}\sum_{k\in\mathcal{T}} x_{i,j,k}. \tag{A.33}$$

A.3.5.4 Tile packing problem

Our last extension constitutes the setting of the tile-packing problem [120]: we require each tile to be placed exactly once yet form a fixed-sized valid tiling,

$$\sum_{i\in\mathcal{H}}\sum_{j\in\mathcal{W}} x_{i,j,k} = 1, \quad \forall k \in \mathcal{T}. \tag{A.34}$$

Note here that this extension requires that $|\mathcal{T}| = |\tilde{\mathcal{A}}|$ as, otherwise, no solution exists.

A.4 Heuristic algorithm for the maximum cover tiling problem

In the previous sections, we have introduced several integer programming formulations for the bounded Wang tiling problem and their extensions. Because finding a valid tiling for a general tile set is $\mathcal{NP}$-complete, we further develop a simple heuristic algorithm for one of the optimization variants, the maximum cover.

A.4.1 Maximum cover tiling of rows

Let us start with revising the decision program (A.10). In this formulation, neglecting any pair of the constraints (A.10b, A.10d), or (A.10c,A.10e), provides a totally unimodular constraint matrix, recall Section A.3.1. Consequently, such simplified problems are solvable deterministically using the simplex method. Moreover, this setting agrees with the maximum

flow problem structure, as (A.10d) and (A.10e) are the flow balances in the source and sink, and (A.10b) with (A.10c) the Kirchhoff law equations. Further complexity reduction is possible by recognizing the (shortest) path problem structure, since the source and sink capacities equal to one, allowing only a single source-to-sink path with positive flow to emerge. Omitting any of these constraint pairs actually produces valid tilings of (finite) stripes, i.e., of rows or columns. However, the edges shared by the neighboring stripes may not comply with the edge matching rules. Starting with this observation, we first focus on a formulation of an efficient approach to generate valid tilings of the stripes, i.e. rows.

As follows from Section A.2, any valid tiling of a row can be visualized as a $|\mathcal{W}|$-long path in the transducer graph $G_{\mathrm{t,h}}$. To simplify subsequent developments, we represent the row-tiling problem by a transducer-based directed acyclic graph (DAG) composed of $|\mathcal{W}| + 3$ vertex layers. While both the first and the last layer contain only a single vertex (the source and terminal), the intermediate layers include $|\mathcal{C}|$ vertices to represent the vertical (east and west) color codes of the tiles, i.e., the states in the $G_{\mathrm{t,h}}$ transducer graph. The source vertex is connected to all vertices in the second layer, facilitating an arbitrary coloring of the west edge of the first tile, and, similarly, all the vertices in the penultimate layer are linked to the terminal to allow for all colors in the last east edge. The intermediate layers are bridged with the transducer edges $\mathcal{E}_{\mathrm{h}}$; see Fig. A.8 for a scheme. Consequently, any $s - t$ path in the yet established directed graph forms a valid tiling of the row, and conversely, all valid tilings build a $s - t$ path.

However, such paths do not exist for tile sets not admitting a valid tiling of the row, so we also need to incorporate voids. Clearly, we can add "void" tiles as edges that would interconnect the layers, i.e., any two consecutive layers would form a complete bipartite graph. However, such approach requires to add at most $|\mathcal{W}||\mathcal{C}|^2$ edges to the graph. Therefore, we add supplementary intermediate layers with a single vertex only, symbolizing the "void" tile type, and connect it to all vertices in the preceding and subsequent layer, see the dashed vertices and edges in Fig. A.9. Consequently, we generate at most $2|\mathcal{W}||\mathcal{C}|$ new edges altogether.

In addition, we assign unitary costs to the edges incoming to the void vertices and zero costs elsewhere. Hence, the path costs are equivalent to the number of voids in the row tiling. Furthermore, because the emergent graph is acyclic and single-sourced, the maximum row-cover tiling is found in $\mathcal{O}(|\mathcal{V}| + |\mathcal{E}|)$ time using the DAG-shortest-path algorithm, where

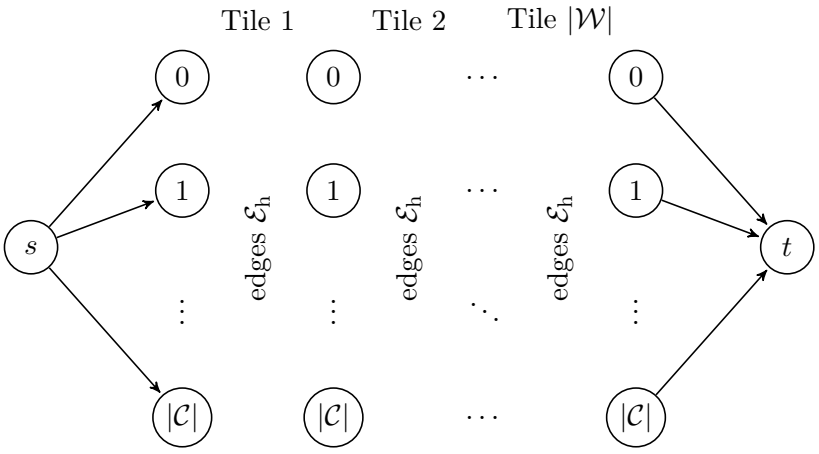


Figure A.8: Transducer-based directed acyclic graph for generation of valid row tilings.

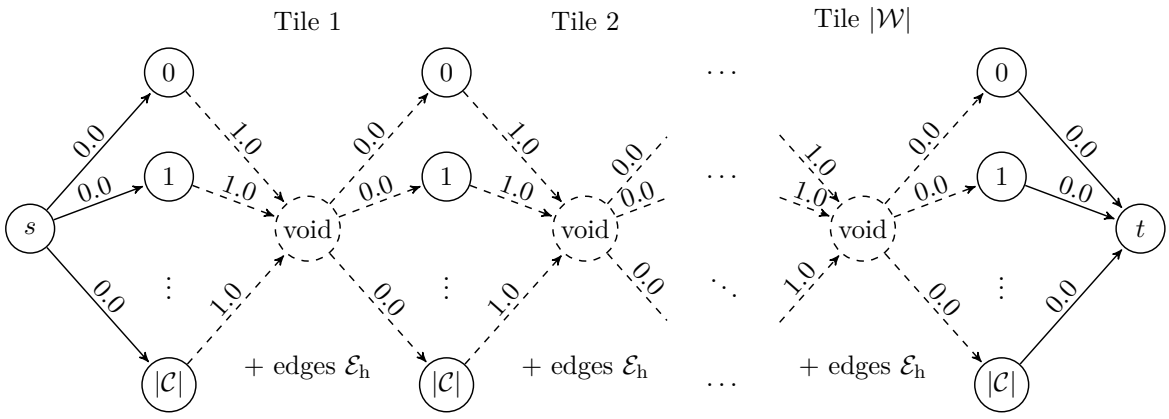


Figure A.9: Transducer-based directed acyclic graph for computing the maximum row cover.

$\mathcal{V}$ denotes the set of the graph vertices and $\mathcal{E}$ the set of the graph edges. In our case, we have

$$|\mathcal{V}| = 2 + (|\mathcal{W}| + 1)|\mathcal{C}| + |\mathcal{W}| = 2 + |\mathcal{W}| + |\mathcal{C}| + |\mathcal{W}||\mathcal{C}|, \tag{A.35a}$$

$$|\mathcal{E}| = 2|\mathcal{C}| + 2|\mathcal{W}||\mathcal{C}| + |\mathcal{W}||\mathcal{T}|. \tag{A.35b}$$

Thus, the overall asymptotic complexity to generate a maximum row cover tiling evaluates as

$$\mathcal{O}(|\mathcal{V}| + |\mathcal{E}|) = \mathcal{O}(2 + |\mathcal{W}| + 3|\mathcal{C}| + 3|\mathcal{W}||\mathcal{C}| + |\mathcal{W}||\mathcal{T}|) = \mathcal{O}(|\mathcal{W}||\mathcal{C}| + |\mathcal{W}||\mathcal{T}|). \tag{A.36}$$

Interestingly, the running time (but not the asymptotic complexity) of the DAG-shortest-path algorithm can be improved by recognizing that the topological order of the graph vertices—which is required for the DAG-shortest-path algorithm—is known from the graph construction method in advance.

Any path of the total cost c_t contain exactly c_t voids in the row tiling. Because the shortest path algorithm therefore minimizes the number of voids, it generates the maximum row cover as its output. These considerations are summarized in the following.

Proposition A.1. *The shortest path in graph in Fig. A.9 is equivalent to the maximum row cover.*

A.4.2 Tiling consecutive rows

Assuming already covered rows $i-1$ and $i+1$, e.g., initially by voids, we aim to generate the maximum cover of the i-th row. Interestingly, this requires only a minor modification of graph in Fig. A.9.

To this goal, we first check the north-east compatibility for each tile $k \in \mathcal{T}$ placed at (i, j). Notice that the compatibility is never violated when the neighbors are voids. For the cases of color mismatches, we remove the edges denoting these incompatible tiles from the graph.

Assume that the rows $(i-1)$ and $(i+1)$ are voids. Then, clearly, inappropriate tiles at the i-th row may prevent the vertically-adjacent positions to be populated by tiles. To limit the appearance of such introduced voids, we include a penalization of a small cost $\epsilon = 0.5(|\mathcal{W}|+1)^{-1}$ to the tiles that admit a single vertical neighbor only, and $\epsilon = (|\mathcal{W}|+1)^{-1}$ to tiles not admitting any vertical neighbor. Notice that these costs are selected such that, in the worst case, the total penalization due to these void-preventing weights amounts to

Algorithm A.1 Simple maximum cover heuristics

```
1: function simpleMaximumCoverHeuristics(𝒯, 𝒜)
2:     𝔗 ← initializeVoidTiling(𝒜)
3:     G_t,h ← getTransducerGraph(𝒯)
4:     for row ← {1, ..., |ℋ|} do
5:         G_DAG ← constructWeightedDAG(G_t,h, 𝒯, 𝔗, row)
6:         shortestPath ← solveDAGShortestPathProblem(G_DAG)
7:         𝔗 ← updateTiling(𝔗, shortestPath, row)
8:     end for
9:     return 𝔗
10: end function
```

$|\mathcal{W}|/(|\mathcal{W}|+1) < 1$, i.e., the maximum-row-cover property (for a given north and south edge coloring) in Proposition A.1 remains satisfied.

Consequently, we can build a simple heuristic algorithm, Algorithm A.1, that requires $|\mathcal{H}|$ maximum row-cover iterations, rendering the overall complexity to be $\mathcal{O}(|\tilde{\mathcal{A}}||\mathcal{C}| + |\tilde{\mathcal{A}}||\mathcal{T}|)$.

Although Algorithm A.1 usually generates relatively large ratio of the number of placed tiles $|\tilde{\mathcal{B}}_{\text{cov}}|$ to $|\tilde{\mathcal{A}}|$, it probably lacks a guaranteed lower bound. Such bounds can, however, be provided by fairly straightforward modifications introduced next.

A.4.3 1/2-approximation algorithm for general tile sets

In this section, we modify Algorithm A.1 to maintain the 1/2 approximation ratio. To this goal, we start with the following observation:

Proposition A.2. *Consider the maximum row-cover tiling of the odd rows of the initially void domain $\mathcal{A}$ given in Section A.4.1. Then, $|\tilde{\mathcal{B}}_{\text{cov}}| \geq 0.5|\tilde{\mathcal{B}}_{\max\text{cov}}|$.*

Proof. Consider that the maximum row-cover problem alone terminates with $|\tilde{\mathcal{B}}_{\max\text{rowcov}}|$ tiles. Based on the maximum row-cover property in Proposition A.1, none of the rows of $\mathcal{A}$ admits a tiling by more than $|\tilde{\mathcal{B}}_{\max\text{rowcov}}|$ tiles. Hence, we have $|\tilde{\mathcal{B}}_{\text{cov}}| \geq \lceil 0.5|\mathcal{H}|\rceil|\tilde{\mathcal{B}}_{\max\text{rowcov}}|$ and $|\tilde{\mathcal{B}}_{\max\text{cov}}| \leq |\mathcal{H}||\tilde{\mathcal{B}}_{\max\text{rowcov}}|$, so that $|\tilde{\mathcal{B}}_{\text{cov}}| \geq \lceil 0.5|\mathcal{H}|\rceil|\tilde{\mathcal{B}}_{\max\text{rowcov}}| \geq 0.5|\mathcal{H}||\tilde{\mathcal{B}}_{\max\text{rowcov}}| \geq 0.5|\tilde{\mathcal{B}}_{\max\text{cov}}|$. □

To exploit Proposition A.2 in Algorithm A.1, we modify the row processing order as $\{1, 3, 2, 5, 4 \dots\}$. Indeed, then each odd row contains exactly $|\tilde{\mathcal{B}}_{\max\text{rowcov}}|$ tiles. Nevertheless, covering the i-th (odd) row without acknowledging which tiles are placed in the $(i-2)$-th row may result in an unnecessarily empty $(i-1)$-th row. To avoid such situations, we do not check for the compatibility with the $(i-1)$-th row voids, but based on the dual transducer graph with the tiles in the $(i-2)$-th row. To this goal, for each south color code in the $(i-2)$-th row, we find admissible colors (states) in the dual transducer graph as the states reachable by an edge-long path. Indeed, the reached states are exactly the admissible north colors of compatible tiles in the i-th row. For the special case of voids in the $(i-2)$-th row, all color codes are assumed to be compatible. Finally, we penalize the incompatibilities with the cost $\epsilon = 0.5(|\mathcal{W}|+1)^{-1}$ as before. The final algorithm then reads as Algorithm A.2, allowing us to state the following, slightly stronger result:

Proposition A.3. *Assume a tile set $\mathcal{T}$ with the longest path in its transducer graph $G_{\text{t,h}}$ at least 2. Then, Algorithm A.2 terminates with $|\tilde{\mathcal{B}}_{\text{cov}}| \geq 0.5|\tilde{\mathcal{A}}|$.*

Algorithm A.2 1/2-approximation algorithm

```
1: function MAXIMUMCOVERAPPROXIMATION050(𝒯, 𝒜)
2:     𝔗 ← initializeVoidTiling(𝒜)
3:     G_t,h ← getTransducerGraph(𝒯)
4:     G_t,v ← getDualTransducerGraph(𝒯)
5:     for row ← {1, 3, 2, 5, 4, ...} do
6:         if row even then
7:             G_DAG ← constructWeightedDAG(G_t,h, 𝒯, 𝔗, row)
8:         else
9:             G_DAG ← constructWeightedDAGFromDTransducer(G_t,h, G_t,v, 𝔗, row, 1)
10:        end if
11:        shortestPath ← solveDAGShortestPathProblem(G_DAG)
12:        𝔗 ← updateTiling(𝔗, shortestPath, row)
13:    end for
14:    return 𝔗
15: end function
```

Proof. When $|\tilde{\mathcal{B}}_{\max \text{rowcov}}| = |\mathcal{W}|$, the proof follows directly from Proposition A.2. For the other cases, the odd rows must contain $|\tilde{\mathcal{B}}_{\max \text{rowcov}}|$ tiles due to Proposition A.1. Because these row-covers are maximal, the sequence of consecutive voids in these rows cannot exceed two, as we could have placed an additional tile otherwise, contradicting with the maximum row-cover property. Moreover, without loss of generality, the length of the shortest path in the i-th row is at most $|\tilde{\mathcal{B}}_{\max \text{rowcov}}| + (|\mathcal{W}| - |\tilde{\mathcal{B}}_{\max \text{rowcov}}|)\epsilon$, which occurs when the $(i-2)$-th and i-th row have the same tile-void patterns. Because the longest void sequence is at most two and the longest path in $G_{\text{t,h}}$ is at least two, we can always place tiles to the north of the voids of the i-th row. □

A.4.4 2/3-approximation algorithm for tilesets with cyclic transducers

Another improvement in the approximation factor of Algorithm A.2 follows from a further restriction to tile sets with all the states in the transducer graphs $G_{\text{t,h}}$ and $G_{\text{t,v}}$ being within at least one graph cycle. Notice that such situation occurs for all tile sets that tile the infinite plane.

To this goal, we modify the costs, and the row processing order to $\{1, 4, 3, 2, 3, 7, 6, 5, 6, \dots\}$. We begin with (i) tiling the maximum row-cover of the first row. Then, we (ii) find the maximum row-cover of the 4-th row such that we penalize possible incompatibilities with the first row based on the dual transducer graph by ϵ. The step (iii) encompasses finding a cover of the 3-rd row with penalized incompatibilities with the first row and enforced voids at even positions. Finally, we find the maximum covers of the rows 2 and 3. We repeat the procedure for the row numbers iteratively increased by 3, see Algorithm A.3. Then, we can make the following statement:

Lemma A.4. *Consider that all states in the transducer graphs $G_{\text{t,h}}$ and $G_{\text{t,v}}$ lie in cycles. Then, Algorithm A.3 terminates with at least $\frac{2}{3}|\tilde{\mathcal{A}}|$ placed tiles.*

Proof. Since the tile set allows for valid tiling of the row, the $\{1, 4, \dots\}$ rows are occupied by exactly $|\mathcal{W}|$ tiles. The $\{3, 6, \dots\}$ rows are then populated by at least $0.5|\mathcal{W}|$ tiles because

Algorithm A.3 2/3-approximation algorithm

```
1: function MAXIMUMCOVERAPPROXIMATION067(𝒯, 𝒜)
2:     𝔗 ← initializeVoidTiling(𝒜)
3:     G_t,h ← getTransducerGraph(𝒯)
4:     G_t,v ← getDualTransducerGraph(𝒯)
5:     setRowsNotVisited()
6:     for row ← {1, 4, 3, 2, 3, 7, 6, 5, 6 ...} do
7:         if mod(row−1,3)== 0 then
8:             G_DAG ← constructWeightedDAGFromDTransducer(G_t,h, G_t,v, 𝔗, row, 2)
9:         else if mod(row−2,3)== 0 then
10:            if rowVisited(row)==false then
11:                G_DAG ← constructWeightedDAGFromDTransducer(G_t,h, G_t,v, 𝔗, row, 1)
12:                G_DAG ← removeTilesAtEvenPositions(G_DAG)
13:            else
14:                G_DAG ← constructWeightedDAG(G_t,h, 𝒯, 𝔗, row)
15:            end if
16:        else
17:            G_DAG ← constructWeightedDAG(G_t,h, 𝒯, 𝔗, row)
18:        end if
19:        shortestPath ← solveDAGShortestPathProblem(G_DAG)
20:        𝔗 ← updateTiling(𝔗, shortestPath, row)
21:        setRowVisited(row)
22:    end for
23:    return 𝔗
24: end function
```

each tile from the rows $\{4, 7, \dots\}$ admits a vertical neighbor. Finally, the $\{2, 5, \dots\}$ rows contain at least the complement of the number of tiles used in the preceding row, because the tiles in the $\{1, 4, 6, \dots\}$ row admit a south neighbor. Depending on the row number, the algorithm places at least

$$|\tilde{\mathcal{B}}_{\text{cov}}| \geq \min\{|\tilde{\mathcal{A}}|, \frac{3}{4}|\tilde{\mathcal{A}}|, \frac{2}{3}|\tilde{\mathcal{A}}|, \frac{3}{4}|\tilde{\mathcal{A}}|, \frac{7}{10}|\tilde{\mathcal{A}}|, \frac{2}{3}|\tilde{\mathcal{A}}|, \dots\} = \frac{2}{3}|\tilde{\mathcal{A}}| \tag{A.37}$$

tiles. □

Remark A.5. If the tile set $\mathcal{T}$ allows for tiling the infinite plane, Algorithm A.3 terminates with $|\tilde{\mathcal{B}}_{\text{cov}}| \geq \frac{2}{3}|\tilde{\mathcal{A}}|$ based on Lemma A.4.

A.4.5 Iterative improvements

Similarly to finding the maximum row covers, we can search for the maximum cover of columns. When combining these two methods, we end up with our final algorithm that has the $\mathcal{O}(|\tilde{\mathcal{A}}|^2|\mathcal{C}| + |\tilde{\mathcal{A}}|^2|\mathcal{T}| + |\mathcal{C}|^2)$ complexity and provides the approximation ratios adjustable by algorithm choice (Algorithms A.1,A.2 or A.3) at line 2 of Algorithm A.4.

Proposition A.6. *Algorithm A.4 runs in a polynomial time and terminates in a finite number of steps.*

Algorithm A.4 Final maximum cover heuristics

```
1: function finalMaximumCoverHeuristics(𝒯, 𝒜)
2:     𝔗 ← generateInitialCover(𝒯,𝒜)
3:     G_t,h ← getTransducerGraph(𝒯)
4:     G_t,v ← getDualTransducerGraph(𝒯)
5:     improvement ← ∞
6:     method ← "columns"
7:     while improvement > 0 do
8:         numVoidsOld ← getNumVoids(𝔗)
9:         if method=="rows" then
10:            for row ← {1, ..., |ℋ|} do
11:                G_DAG ← constructSimpleDAG(G_t,h, 𝔗, row)
12:                shortestPath ← solveDAGShortestPathProblem(G_DAG)
13:                𝔗 ← updateTiling(𝔗, shortestPath, row)
14:                method ← "columns"
15:            end for
16:        else
17:            for column ← {1, ..., |𝒲|} do
18:                G_DAG ← constructSimpleDAG(G_t,v, 𝔗, column)
19:                shortestPath ← solveDAGShortestPathProblem(G_DAG)
20:                𝔗 ← updateTiling(𝔗, shortestPath, column)
21:                method ← "rows"
22:            end for
23:        end if
24:        improvement ← numVoidsOld − getNumVoids(𝔗)
25:    end while
26:    return 𝔗
27: end function
```

Proof. We have already shown that finding a maximum row-cover has $\mathcal{O}(|\mathcal{W}||\mathcal{C}| + |\mathcal{W}||\mathcal{T}|)$ complexity. Further, finding the 2-long paths in the transducer graph possesses the $|\mathcal{C}|^2$ complexity and can be run only once, prior to the algorithm main loop. Altogether, Algorithm A.3 requires at most $4/3|\mathcal{H}|$ inner iterations, so that we have the $\mathcal{O}(|\tilde{\mathcal{A}}||\mathcal{C}| + |\tilde{\mathcal{A}}||\mathcal{T}| + |\mathcal{C}|^2)$ overall complexity.

Regardless of the method at line 2 of Algorithm A.4, the improving loop runs at most $|\tilde{\mathcal{A}}|$ times. Consequently, the algorithm is finite and possesses the $\mathcal{O}(|\tilde{\mathcal{A}}|^2|\mathcal{C}| + |\tilde{\mathcal{A}}|^2|\mathcal{T}| + |\mathcal{C}|^2)$ complexity. □

A.5 Results

Having developed several exact and heuristic methods, this section is devoted to their numerical examination. We begin with assessing the performance of the integer programming formulations in Section A.5.1. Then, in Section A.4, we also relate these results with the outputs of the heuristic algorithms.

Extensions of the integer programs are investigated in subsequent sections. First, we

demonstrate the usefulness of the packing constraint by comparing the efficiency of solution to the tile-packing problem using our method with the times reported by Lagae and Dutré [120], Section A.5.3. Subsequently, we also reveal two unexpected discoveries observed during extensive tests of the formulations: the Knuth [103] tile set contains a tile unusable in infinite tilings, Section A.5.4, and the Lagae *et al.* [121] tile set of 44 corner tiles lacks aperiodicity, Section A.5.5.

We implemented all the above described methods in C++. As the integer programming solver, we adopt the state-of-the-art optimizer Gurobi 9.0, dynamically linked to the compiled binary. Numerical tests were evaluated on a workstation running the Ubuntu 16.04 operating system, equipped with 128 GB of RAM and Intel® Xeon® E5-2630v3 CPU clocked at 2.40 GHz.

A.5.1 Integer programming formulation

In this section, we investigate the performance of all integer programming formulations from Section A.3, i.e., the decision program (A.10), the maximum rectangular tiling (A.16), the maximum cover (A.19), and the maximum adjacency constraint satisfaction problem (A.22).

Unfortunately, there does not exist any standard set of bounded tiling problems, except for specific, mostly aperiodic tile sets listed in the literature, recall Section A.1.1. Hence, we consider a set of benchmark problems consisting of five aperiodic tile sets (11 tiles over 4 colors by Jeandel and Rao [92], 13 tiles over 5 colors by Čulík [44], 14 tiles over 6 colors by Kari [98], 16 tiles over 6 colors by Ammann [79], and 56 tiles over 12 colors by Robinson [161]), two stochastic tile sets introduced in computer graphics (8 tiles over 2 colors by Cohen *et al.* [41] and a set of 16 tiles over 4 edge colors by Lagae and Dutré [119]), two periodic tile sets (10 tiles over 4 colors by Wang [204], and the set of 30 tiles over 17 edge colors by Lagae *et al.* [121] and Nurmi [145]). In addition, in Fig. A.10, we introduce two tile sets that do not allow for a valid tiling of the infinite domain.

For all these tile sets, we aimed at generating valid tilings of the respective sizes 20×20, 25×25, and 30×30. The running time of the Gurobi solver was limited by 300 seconds while running in the single-threaded mode.

The results shown in Table A.1 illustrate that the performance of the decision program (A.10) surpasses any of the optimization variants. However, it failed four times to find a feasible solution in the time limit. In these cases, the output of the optimization problems (A.16, A.19, A.22) provides more usable data. Interestingly, the decision problem (A.10) also showed to be more efficient in the case of proving that the domain $|\mathcal{A}|$ lacks $\mathcal{T}$-tilability.

Comparison of the optimization variants hints that the maximum cover (A.19) and the maximum adjacency constraint satisfaction problem (A.22) are easier to manage than the maximum rectangular tiling (A.16). Indeed, generating any smaller rectangular domain

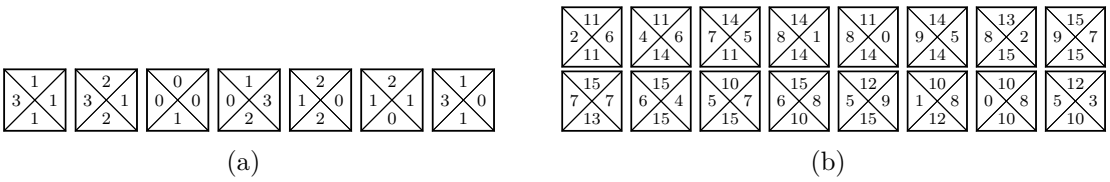


Figure A.10: New tile sets (a) Finite1 of 7 tiles over 4 colors, and (b) Finite2 of 16 tiles over 16 colors used in our algorithmic tests.

Tile set	Size	**Dec. prog.** (A.10)		**Max. rect.** (A.16)		**Max. cov.** (A.19)		**Max. CSP** (A.22)	
		Time [s]	*Objective*	*Time [s]*	*Objective*	*Time [s]*	*Objective*	*Time [s]*	*Objective*
Aperiodic1 (11/4) [92]	20 × 20	0.130	0	172.836	400	300.052	*385	300.079	*752
	25 × 25	99.698	0	300.060	*125	300.101	*596	300.110	*1108
	30 × 30	300.060	*infeasible	300.133	*60	300.160	*854	300.135	*1536
Aperiodic2 (13/5) [44]	20 × 20	0.184	0	300.067	*300	300.064	*399	300.090	*688
	25 × 25	32.383	0	300.116	*225	300.071	*611	300.135	*1174
	30 × 30	300.051	*infeasible	300.096	*60	300.147	*878	300.168	*1539
Aperiodic3 (14/6) [98]	20 × 20	14.743	0	300.067	*300	300.047	*395	300.052	*743
	25 × 25	300.045	*infeasible	300.067	*75	300.079	*612	300.083	*1108
	30 × 30	300.079	*infeasible	300.096	*90	300.099	*873	300.101	*1547
Aperiodic4 (16/6) [79]	20 × 20	0.189	0	256.778	400	300.072	*391	137.843	760
	25 × 25	0.161	0	300.298	*100	300.089	*543	300.091	*1028
	30 × 30	0.267	0	300.103	30	300.125	*784	300.133	*1529
Aperiodic5 (56/12) [161]	20 × 20	0.341	0	300.151	*20	300.293	*350	300.177	*616
	25 × 25	0.562	0	300.234	*25	300.429	553	300.230	*1017
	30 × 30	0.619	0	300.297	*30	300.613	*795	300.360	*1521
Stochastic1 (8/2) [41]	20 × 20	0.042	0	0.070	400	0.045	400	3.945	760
	25 × 25	0.059	0	0.085	625	0.067	625	5.711	1200
	30 × 30	0.081	0	0.204	900	0.094	900	13.035	1740
Stochastic2 (16/4) [119]	20 × 20	0.110	0	0.101	400	0.177	400	4.453	760
	25 × 25	0.163	0	0.155	625	0.276	625	5.429	1200
	30 × 30	0.228	0	0.221	900	0.430	900	10.194	1740
Periodic1 (10/4) [204]	20 × 20	0.085	0	44.474	400	37.095	400	38.462	760
	25 × 25	0.125	0	300.053	*50	168.901	625	300.060	*1084
	30 × 30	0.207	0	300.083	*90	300.112	*783	300.175	*1599
Periodic2 (30/17) [145]	20 × 20	0.821	0	195.723	400	300.244	*328	262.167	760
	25 × 25	1.079	0	300.143	*25	300.393	*508	187.188	1200
	30 × 30	1.328	0	300.198	*30	300.538	*733	300.436	*1522
Finite1 (7/4)	20 × 20	0.046	infeasible	300.028	*100	300.029	*379	300.033	*723
	25 × 25	0.069	infeasible	300.041	*125	300.054	*584	300.048	*1108
	30 × 30	0.086	infeasible	300.056	*60	300.059	*834	300.068	*1628
Finite2 (16/16)	20 × 20	0.081	infeasible	300.059	*60	300.269	*323	300.075	*682
	25 × 25	0.125	infeasible	300.084	*25	300.120	*493	300.124	*1027
	30 × 30	0.179	infeasible	300.120	*30	300.171	*690	300.135	*1525

Table A.1: Benchmark results. Values marked by an asterisk denote a premature termination of the solver.

remains $\mathcal{NP}$-complete, preventing any polynomial-time approximation algorithm to exist. On the other hand, both the formulations (A.19) and (A.22) admit a polynomial time approximation, recall Section A.3.

A.5.2 Heuristic algorithms

Second, we compare the performance of the maximum cover formulation (A.19) with the heuristic Algorithm A.4 supplied with three different initial coverings, i.e., based on Algorithms A.1, A.2 and A.3.

Algorithm A.4 ran sequentially. In order to limit the dependence of the heuristic algorithm on ordering of tiles, we made it stochastic by randomizing edge order in the directed acyclic graphs. Thus, we evaluated Algorithm A.4 100 times for each of the tested option. The best, worst, and mean results are listed in Table A.2.

From Table A.2, it follows that the initialization with the cover from Algorithm A.1 is the most efficient for the tested tile sets—both in terms of speed and performance. The remaining two initializations seem to be fairly comparable on average. While for Algorithm A.1, at least 82% tiles were always placed, only more than 60% followed from Algorithm A.2. Using Algorithm A.3, we obtained at least 70% tiles.

Tile set	Size	Alg. A.4 with Alg. A.1				Alg. A.4 with Alg. A.2				Alg. A.4 with Alg. A.3			
		t [s]	*min*	*avg*	*max*	*t [s]*	*min*	*avg*	*max*	*t [s]*	*min*	*avg*	*max*
Aperiodic1 (11/4) [92]	20 × 20	0.338	357	**368.99**	381	0.752	241	350.87	374	0.728	334	357.56	378
	25 × 25	0.472	566	**576.99**	588	1.237	376	550.28	580	1.214	526	556.18	577
	30 × 30	0.699	815	**829.97**	845	1.783	544	806.07	838	1.751	784	807.86	830
Aperiodic2 (13/5) [44]	20 × 20	0.462	359	**370.46**	378	0.756	308	360.34	377	0.625	301	342.38	368
	25 × 25	0.728	559	**577.18**	592	1.219	442	565.81	584	1.004	470	543.94	580
	30 × 30	1.080	815	**831.93**	852	1.641	668	814.39	838	1.581	756	787.72	829
Aperiodic3 (14/6) [98]	20 × 20	0.353	361	**375.74**	387	0.582	340	363.89	381	0.601	353	373.17	385
	25 × 25	0.566	566	**586.71**	604	0.903	528	570.58	589	0.955	567	584.40	602
	30 × 30	0.947	819	**846.96**	864	1.404	759	825.40	854	1.418	824	**846.90**	866
Aperiodic4 (16/6) [79]	20 × 20	0.307	355	**367.20**	382	1.026	297	343.01	365	1.069	296	345.38	363
	25 × 25	0.509	551	**572.00**	592	1.604	510	537.40	560	1.771	519	543.94	562
	30 × 30	0.766	803	**824.54**	849	2.385	757	776.87	799	2.531	752	785.05	810
Aperiodic5 (56/12) [161]	20 × 20	0.320	344	**360.17**	381	1.014	268	342.27	363	1.205	295	333.03	347
	25 × 25	0.450	544	**562.12**	589	1.513	399	531.13	568	2.202	495	527.95	547
	30 × 30	0.730	790	**808.30**	840	2.488	586	782.18	815	3.032	723	759.31	793
Stochastic1 (8/2) [41]	20 × 20	0.144	400	**400.00**	400	0.132	400	**400.00**	400	0.164	400	**400.00**	400
	25 × 25	0.155	625	**625.00**	625	0.155	625	**625.00**	625	0.188	625	**625.00**	625
	30 × 30	0.192	900	**900.00**	900	0.182	900	**900.00**	900	0.224	900	**900.00**	900
Stochastic2 (16/4) [119]	20 × 20	0.130	400	**400.00**	400	0.128	400	**400.00**	400	0.165	400	**400.00**	400
	25 × 25	0.158	625	**625.00**	625	0.157	625	**625.00**	625	0.197	625	**625.00**	625
	30 × 30	0.179	900	**900.00**	900	0.183	900	**900.00**	900	0.223	900	**900.00**	900
Periodic1 (10/4) [204]	20 × 20	0.319	342	**354.13**	378	0.946	298	337.34	352	0.936	304	341.29	354
	25 × 25	0.529	539	**552.72**	582	1.497	512	529.03	547	1.406	511	533.52	552
	30 × 30	0.704	775	**796.05**	835	2.068	736	761.70	786	1.945	751	771.05	800
Periodic2 (30/17) [145]	20 × 20	0.262	361	**380.41**	397	1.375	313	336.23	353	1.205	283	346.38	365
	25 × 25	0.412	559	**595.54**	619	2.232	504	532.26	553	1.858	545	564.73	579
	30 × 30	0.563	831	**858.33**	888	3.263	718	761.81	793	2.851	761	800.81	823
Finite1 (7/4)	20 × 20	0.291	353	**360.06**	367	0.660	346	358.55	369	0.764	322	355.54	370
	25 × 25	0.406	551	**561.29**	573	0.983	542	557.60	570	1.231	540	554.43	566
	30 × 30	0.587	795	**806.85**	821	1.369	782	802.77	814	1.731	776	805.31	820
Finite2 (16/16)	20 × 20	0.353	329	**344.48**	360	0.726	287	331.45	350	0.642	294	327.90	353
	25 × 25	0.536	513	536.23	549	1.066	494	530.95	547	0.656	507	**542.91**	559
	30 × 30	0.856	748	**771.18**	796	1.783	688	760.92	782	1.667	701	751.80	791

Table A.2: Numerical tests of the maximum-cover heuristic, Algorithm A.4, initialized based on Algorithms A.1, A.2, and A.3.

When comparing Table A.1 with Table A.2, few patterns emerge. First, the heuristic algorithm always generates valid tilings if (any of) the stochastic tile sets are used. For aperiodic and periodic tile sets, Gurobi required considerably longer time to reach the solutions of a similar quality, but usually surpassed the developed algorithms in the time limit of 300 s. In the case of Algorithm A.1, it can be seen that the resulting covers are very competitive to the outputs of (A.19), and also obtained in much shorter times.

A.5.3 Periodic tile packing problem

As the second numerical example, we consider the periodic tile packing problem investigated in computer graphics applications [120]. Considering a complete edge tile set, the authors searched for a periodic square valid tiling with each tile from the tile set used exactly once. Clearly, such tilings not only contain the entire (textural) information stored in individual tiles but also maintain compatibility with the traditional periodic arrangement.

While Lagae and Dutré [120] proposed a backtracking-based algorithm to generate periodic packings, we rely here on a solution to the decision program (A.10) supplemented with the packing (A.34) and fixed periodicity (A.31) constraints. The resulting core times spent in the search for a single feasible solution (Table A.3) clearly illustrate higher effectiveness of our

method.

Tile set	Time (A.10,A.31,A.34)	Time Lagae and Dutré [120]
Stochastic edge (16/2)	< 1 sec.	< 1 sec.
Stochastic edge (81/3)	< 1 sec.	< 1 sec.
Stochastic edge (256/4)	9 sec.	140 days
Stochastic edge (625/5)	4 days	-

Table A.3: Periodic tile packing problem: comparison of core time needed to find a single feasible solution by integer programming (second column) and by the backtracking method (third column) proposed in Lagae and Dutré [120] to find a feasible solution.

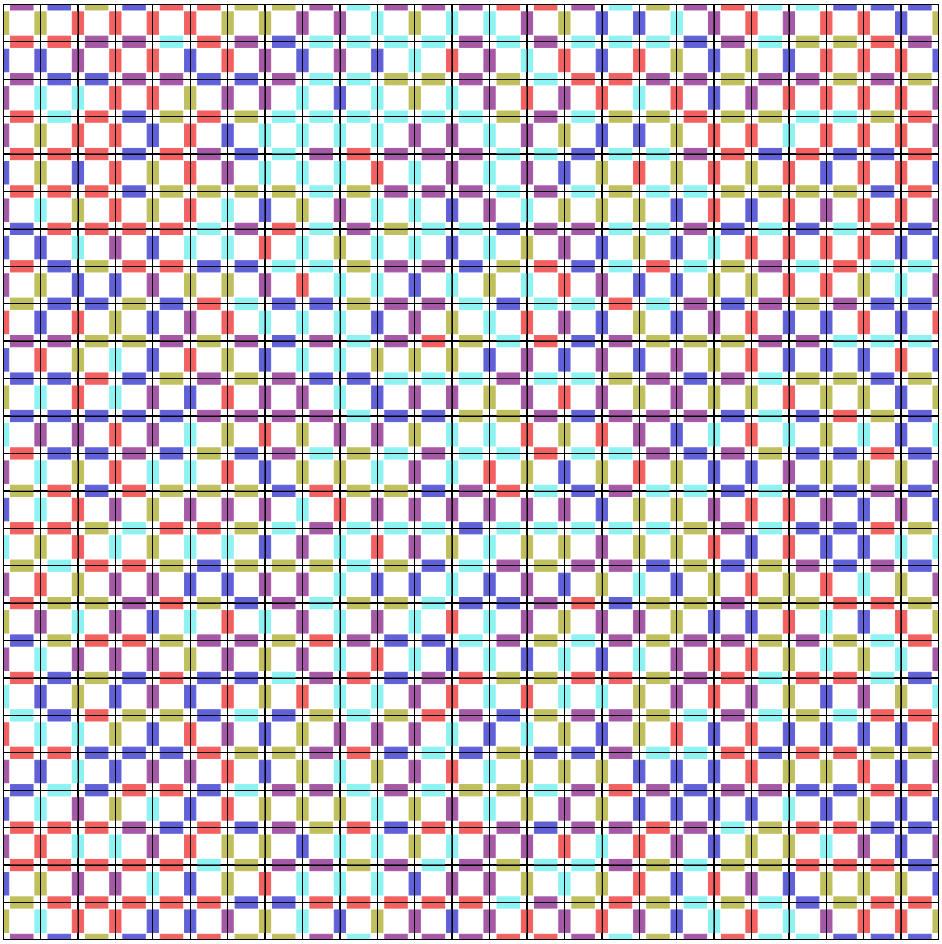


Figure A.11: Periodic packing of a complete set of 625 tiles over 5 colors.

A.5.4 Unusable tile in the Knuth tile set

One of the oldest aperiodic tile sets, containing 92 tiles over 26 colors, is due to Knuth [103, Exercise 5 in Section 2.3.4.3]. Generating valid tilings from the Knuth tile set using the decision program (A.10) together with the tile-based boundary conditions, recall Section

A.3.5.1, led to an unexpected observation that enforced placement of the βUS tile makes the program (A.10) infeasible under certain circumstances.

After a careful investigation, it indeed turned out that there does not exist any 2×2 valid tiling with the βUS tile placed at $(2, 2)$. Moreover, there also does not exist any 4×3 valid tiling with the βUS tile placed at $(3, 1)$. Thus, using the maximum-cover optimization variant (A.19) and the βUS tile enforced at the respective coordinate, there exist exactly 31 optimal solutions with the objective function equal to 3, and 498 optimal solutions with the objective function equal to 11, respectively.

Consequently, the βUS tile can appear only in the strip of at most 3 consecutive infinite columns, and does not allow for simply-connected valid tilings of the infinite plane. In private communication, Knuth confirmed the issue, and discovered another 5 tiles that are *unnecessary* but usable in infinite valid tilings, allowing for a possible reduction of the tile set to 86 tiles. For more information, we refer the interested reader to Knuth's discussion about the reduced tile set [105, Exercise 221 in Section 7.2.2.1].

A.5.5 Periodicity of the Lagae corner tile set

Analogously to the Wang tiles, with the connectivity information stored in the edges, Lagae and Dutré [119] introduced the *corner tiles* with colored corners. As Wang [204] noted in 1975, these formalisms are interchangeable if the (infinite) domino problem is considered, because every set of Wang tiles can be represented by sets of corner tiles with greater or equal cardinality [121]. However, the corner tiles avoid the so-called corner problem of Wang tiles in computer graphics [119], motivating Lagae *et al.* [121] to develop conversion methods for transforming Wang tiles to the corner tiles, and vice versa.

A direct product of these conversion methods are aperiodic tile sets of the corner tiles [121]. Among other well-settled approaches, two of the methods—called horizontal, and vertical translations—were used to convert the Ammann set of 16 Wang tiles over 6 colors [79] to the set of 44 corner tiles over 6 colors, and the resulting isomorphic corner tile sets were claimed aperiodic [121]. In 2016, Nurmi [145] noticed that, in this set, 14 tiles are unusable in infinite valid tilings, and reduced the corner tile set to 30 tiles over 6 colors. Quite surprisingly, neither Lagae *et. al.* nor Nurmi have recognized that the tile set forms a torus, and is therefore periodic! To this goal, we show the construction of the smallest periodic pattern next.

Instead of developing a new formulation for another type of tiles, we first notice that the corner tiles are actually a subset of the Wang tiles, and therefore every set of corner tiles is represented equivalently by a set of Wang tiles of the same cardinality, see Appendix A.A. For these tiles, we solve the rectangular tiling formulation (A.16) with periodic boundary conditions (A.32) and an objective function to find the smallest tiling (A.33). As its output, we receive the optimal value of 6 and 12 optimal periodic rectangular tilings of the size 2×3. Not surprisingly, all these solutions follow from only two periodic patterns shown in Fig. A.12 by translations over the infinite plane.

Having revealed the smallest periodic solutions, it remains to be shown why the Lagae conversion methods failed. To this goal, Lagae *et al.* [121] mentioned that their methods lack bijectiveness in general but assumed it is not the case here. Therefore, we believe it is useful to state the conditions under which the methods are bijective, and show that they are not satisfied for the Ammann tile set.

Lemma A.7. *The horizontal translation method of Lagae* et al. *[121] is bijective iff the dual*

2	1	1	2
3	0	0	3
3	0	0	3
5	1	1	5
5	1	1	5
2	1	1	2

(a)

2	1	1	2
3	0	0	3
3	0	0	3
4	0	0	4
4	0	0	4
2	1	1	2

(b)

Figure A.12: Rectangular periodic valid tilings. Translating a 2×3 rectangle over the infinite valid tiling generated from (a) or (b) leads to 6 different periodic patterns of the same size. Consequently, the tile set allows for 12 periodic rectangles of the size 2×3.

transducer graph $G_{\mathrm{T,v}}$ of the input tile set $\mathcal{T}$ does not contain any parallel arcs.

Proof. The horizontal translation method is formally a mapping $\mathcal{T} \times \mathcal{T} \mapsto \mathcal{T}_{\mathrm{corner}}$ that generates $\forall (p, q) \in \mathcal{T}^2 : c_p^{\mathrm{e}} = c_q^{\mathrm{w}}$ a corner tile $(c_p^{\mathrm{n}}, c_p^{\mathrm{s}}, c_q^{\mathrm{s}}, c_q^{\mathrm{n}})$. To be bijective, the cardinality of the output needs to be equal to the cardinality of the input, and the mapping has to produce unique output for each input. Consequently, all the tiles $p \in \mathcal{T}$ in the original tile set must be uniquely determined by c_p^{n} and c_p^{s}, as the color codes of the vertical edges of $\mathcal{T}$ are avoided in the construction of $\mathcal{T}_{\mathrm{corner}}$.

Let us now consider that the dual transducer graph contains a parallel arc connecting the state c^{n} with c^{s}. Then, there may exist two tiles colored by $(c^{\mathrm{n}}, c_p^{\mathrm{w}}, c^{\mathrm{s}}, c_p^{\mathrm{e}})$ and $(c^{\mathrm{n}}, c_q^{\mathrm{w}}, c^{\mathrm{s}}, c_q^{\mathrm{e}})$ that are indistinguishable in $\mathcal{T}_{\mathrm{corner}}$, which contradicts the bijection. For the other option, if the transducer graph does not contain any parallel arcs, then each c_q^{n}, c_q^{s} identifies with a single arc labeled by $c_q^{\mathrm{w}}|c_q^{\mathrm{e}}$, i.e., with a single tile, which completes the proof. □

Similarly to Proposition A.7, we also formulate the conditions for bijection of the vertical translation method:

Lemma A.8. *The vertical translation method of Lagae* et al. *[121] is bijective iff the transducer graph $G_{\mathrm{T,h}}$ of the input tile set $\mathcal{T}$ does not contain any parallel arcs.*

Proof. Proof as in Proposition A.7 with the tile set rotated by 90 degrees. □

Remark A.9. If the input tile set is the Wang tile representation of the corner tile set, then both the horizontal and vertical translations on this set are bijective.

For the Ammann tile set, we obtain the transducer graph $G_{\mathrm{T,h}} = G_{\mathrm{T,v}}$ shown in Fig. A.13. Clearly, there exist parallel arcs $1 \to 0$. Moreover, using the same approach, we can show that the horizontal translation method also fails for the Robinson tile set of 24 tiles over 24 colors [79], contrary to the claims in [121], and the corresponding corner tile set is also periodic.

A.6 Conclusions

In this contribution, we investigated methods generating bounded Wang tilings for arbitrary tile sets. To this goal, we developed an $\mathcal{NP}$-complete binary linear programming formulation (A.10), as well as its $\mathcal{NP}$-hard optimization variants relaxing some of the initial assumptions: tilability of the entire rectangular domain leading to the maximum rectangular tiling formulation (A.16), simple-connectedness to the maximum-cover program (A.19), and tiling validity

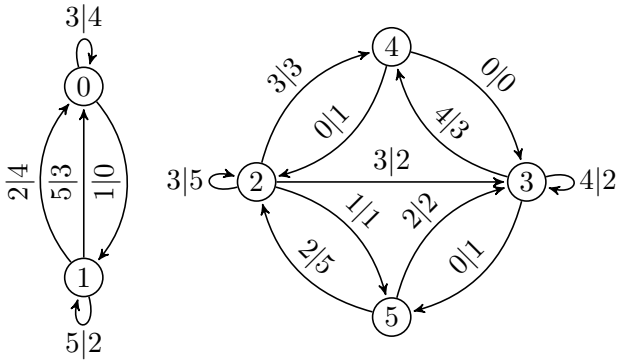


Figure A.13: Transducer graph of the Ammann set of 16 Wang tiles over 6 colors.

to the maximum constraint satisfaction problem (A.22). In addition, we supplemented these formulations enabling for a control over individual tiles and their colors, included a tile-packing constraint to enable generation of periodic tile packings, and introduced the variable-sized periodic constraints to facilitate a computation of the smallest periodic patterns.

Motivated by the $\mathcal{NP}$-hardness of the optimization formulations, we developed simple yet efficient heuristic algorithms for the maximum-cover variant (A.19). These algorithms rely on the fact that generating the maximum row cover is equivalent to finding the shortest path in a directed acyclic graph. Moreover, well-chosen costs of the graph edges also maintain color matches with the neighboring rows. Thus, in the simplest case, a heuristic solution follows from a sequential generation of row-cover tilings. Moreover, with simple modifications to the row processing order, we showed how to provide a $1/2$ approximation factor for general tile sets and a $2/3$ guarantee for tile sets whose transducer graphs are cyclic.

We illustrated the effectiveness of these methods on a collection of 11 tile sets. Generating tilings of the respective sizes 20×20, 25×25, and 30×30 revealed that the decision program (A.10) is the most efficient for our test problems. However, when a time limit is imposed or if the tile set does not allow for a valid tiling of the entire domain, then the maximum cover (A.19) and maximum adjacency constraint satisfaction problems (A.22) appear to be similarly efficient. The remaining formulation—maximum rectangular tiling (A.16)—exhibits the worst performance as it is $\mathcal{NP}$-hard to find any rectangular tilings except for a single row or column.

The usefulness of integer programming extensions was demonstrated by means of three illustrative problems: showing a better solution efficiency to the tile packing problem than the backtracking approach of Lagae and Dutré [120], revealing an unusable tile in the Knuth [103] tile set, and proving that the Lagae *et al.* [121] tile set of corner tiles lacks aperiodicity. For the latter case, we also included an explanation why the tile set construction method failed.

Furthermore, we also tested the performance of developed heuristic algorithms against the outputs generated by Gurobi optimizer solving the problem (A.19) for 300 s. The testing revealed that the simplest setup of Algorithm A.4 initialized with the cover generated by Algorithm A.1 produces the best results that are obtained faster yet in most cases competitive with the output of Gurobi. Somewhat surprisingly, the variants with guaranteed lower bounds exhibited slightly worse performance on average.

Having summarized our contributions, we believe that this chapter has not only founded

new methods that can possibly be applied in materials engineering, but also a simple and quite extensible framework to verify theoretical results in Wang tilings.

A.A Corner tiles represented as Wang tiles

Each corner tile in the corner tile set is defined by a quadruple of color codes $(c_k^{\mathrm{nw}}, c_k^{\mathrm{sw}}, c_k^{\mathrm{se}}, c_k^{\mathrm{ne}})$, with c_k^{nw}, c_k^{sw}, c_k^{se}, and c_k^{ne} denoting the colors of the northwest, southwest, southeast, and northeast corner of the k-th tile. Similarly to Wang tiles, corner tiles are assembled such that the color codes at the adjoining corners match.

This is, however, also maintained if we denote their edges by labels in the form of tuples of the corner codes, $(c_k^{\mathrm{nw}}, c_k^{\mathrm{ne}})$, $(c_k^{\mathrm{nw}}, c_k^{\mathrm{sw}})$, $(c_k^{\mathrm{sw}}, c_k^{\mathrm{se}})$, and $(c_k^{\mathrm{ne}}, c_k^{\mathrm{se}})$, each of which denotes a single edge label of the north, west, south, and east edge, respectively. Consequently, we can compute a unique color codes as

$$c_k^{\mathrm{n}} = c_k^{\mathrm{nw}} + c_k^{\mathrm{ne}} n_{\mathrm{vc}}, \tag{A.38a}$$

$$c_k^{\mathrm{w}} = c_k^{\mathrm{nw}} + c_k^{\mathrm{sw}} n_{\mathrm{vc}}, \tag{A.38b}$$

$$c_k^{\mathrm{s}} = c_k^{\mathrm{sw}} + c_k^{\mathrm{se}} n_{\mathrm{vc}}, \tag{A.38c}$$

$$c_k^{\mathrm{e}} = c_k^{\mathrm{ne}} + c_k^{\mathrm{se}} n_{\mathrm{vc}}, \tag{A.38d}$$

where n_{vc} stands for the number of colors used in the corner tile set. Graphical illustration of the corner-edge tile equivalence is shown in Fig. A.14.

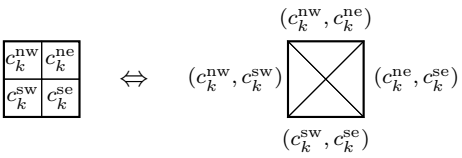


Figure A.14: A corner tile expressed using the edge formalism.

Bibliography

[1] N. Aage, E. Andreassen, B. S. Lazarov, and O. Sigmund, Giga-voxel computational morphogenesis for structural design, *Nature*, 550(7674):84–86, 2017, DOI: 10.1038/nature23911.

[2] W. Achtziger and M. Kočvara, On the maximization of the fundamental eigenvalue in topology optimization, *Structural and Multidisciplinary Optimization*, 34(3):181–195, 2007, ISSN 1615147X, DOI: 10.1007/s00158-007-0117-3.

[3] W. Achtziger and M. Kočvara, Structural topology optimization with eigenvalues, *SIAM Journal on Optimization*, 18(4):1129–1164, 2008, DOI: 10.1137/060651446.

[4] W. Achtziger, M. P. Bendsøe, A. Ben-Tal, and J. Zowe, Equivalent displacement based formulations for maximum strength truss topology design, *IMPACT of Computing in Science and Engineering*, 4(4):315–345, 1992, DOI: 10.1016/0899-8248(92)90005-s.

[5] B. Addis, *Building with Reclaimed Components and Materials*. Routledge, 2012, DOI: 10.4324/9781849770637.

[6] J. Alexandersen and B. S. Lazarov, Topology optimisation of manufacturable microstructural details without length scale separation using a spectral coarse basis preconditioner, *Computer Methods in Applied Mechanics and Engineering*, 290:156–182, 2015, DOI: 10.1016/j.cma.2015.02.028.

[7] G. Allaire, P. Geoffroy-Donders, and O. Pantz, Topology optimization of modulated and oriented periodic microstructures by the homogenization method, *Computers & Mathematics with Applications*, 78(7):2197–2229, 2019, DOI: 10.1016/j.camwa.2018.08.007.

[8] H. An and H. Huang, Topology and sizing optimization for frame structures with a two-level approximation method, *AIAA Journal*, 55(3):1044–1057, 2017, DOI: 10.2514/1.j055020.

[9] 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*, 43(1):1–16, 2010, DOI: 10.1007/s00158-010-0594-7.

[10] M. Anjos, Y. Emine, A. Lodi, and Z. Sun, A note on the Lasserre hierarchy for different formulations of the maximum independent set problem, *Operations Research Letters*, 49(1):30–34, 2021, DOI: 10.1016/j.orl.2020.10.009.

[11] M. F. Anjos and J. B. Lasserre, editors, *Handbook on semidefinite, conic and polynomial optimization*. Springer US, 2012, DOI: 10.1007/978-1-4614-0769-0.

[12] P. Antolin, A. Buffa, E. Cohen, J. F. Dannenhoffer, G. Elber, S. Elgeti, R. Haimes, and R. Riesenfeld, Optimizing micro-tiles in micro-structures as a design paradigm, *Computer-Aided Design*, 115:23–33, 2019, DOI: 10.1016/j.cad.2019.05.020.

[13] M. Baandrup, O. Sigmund, H. Polk, and N. Aage, Closing the gap towards super-long suspension bridges using computational morphogenesis, *Nature Communications*, 11(1), 2020, DOI: 10.1038/s41467-020-16599-6.

[14] V. Balabanov and R. T. Haftka, Topology optimization of transport wing internal structure, *Journal of Aircraft*, 33(1):232–233, 1996, ISSN 0021-8669, DOI: 10.2514/3.46926.

[15] A. Ben-Tal and A. Nemirovski, Robust truss topology design via semidefinite programming, *SIAM Journal on Optimization*, 7(4):991–1016, 1997, ISSN 1052-6234, DOI: 10.1137/S1052623495291951.

[16] A. Ben-Tal and A. Nemirovski, Structural design, In H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors, *Handbook of semidefinite programming*, chapter 15, pages 443–467. Springer, Boston, MA, 2000, DOI: 10.1007/978-1-4615-4381-7_15.

[17] A. Ben-Tal and A. Nemirovski, *Lectures on modern convex optimization*. Society for Industrial and Applied Mathematics, 2001, DOI: 10.1137/1.9780898718829.

[18] A. Ben-Tal, M. Kočvara, A. Nemirovski, and J. Zowe, Free material design via semidefinite programming: The multiload case with contact conditions, *SIAM Journal on Optimization*, 9(4):813–832, 1999, ISSN 1052-6234, DOI: 10.1137/S1052623497327994.

[19] A. Ben-Tal, F. Jarre, M. Kočvara, A. Nemirovski, and J. Zowe, Optimal design of trusses under a nonconvex global buckling constraint, *Optimization and Engineering*, 1 (2):189–213, 2000, DOI: 10.1023/A:1010091831812.

[20] M. P. Bendsøe and N. Kikuchi, Generating optimal topologies in structural design using a homogenization method, *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, 1988, DOI: 10.1016/0045-7825(88)90086-2.

[21] M. P. Bendsøe and O. Sigmund, Material interpolation schemes in topology optimization, *Archive of Applied Mechanics (Ingenieur Archiv)*, 69(9-10):635–654, 1999, DOI: 10.1007/s004190050248.

[22] M. P. Bendsøe and O. Sigmund, *Topology optimization: Theory, methods, and applications*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, ISBN 978-3-642-07698-5, DOI: 10.1007/978-3-662-05086-6.

[23] M. P. Bendsøe, A. Ben-Tal, and R. Haftka, New displacement-based methods for optimal truss topology design, In *32nd Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, 1991, DOI: 10.2514/6.1991-1215.

[24] M. P. Bendsøe, A. Ben-Tal, and J. Zowe, Optimization methods for truss geometry and topology design, *Structural and Multidisciplinary Optimization*, 7(3):141–159, 1994, ISSN 0934-4373, DOI: 10.1007/bf01742459.

[25] R. Berger, *The Undecidability of the Domino Problem*, PhD thesis, Harvard University, 1964.

[26] R. Berger, The undecidability of the domino problem, *Memoirs of the American Mathematical Society*, 0(66), 1966, DOI: 10.1090/memo/0066.

[27] V. Birman and G. A. Kardomateas, Review of current trends in research and applications of sandwich structures, *Composites Part B: Engineering*, 142:221–240, 2018, ISSN 13598368, DOI: 10.1016/j.compositesb.2018.01.027.

[28] C. D. Bisbos and P. M. Pardalos, Second-order cone and semidefinite representations of material failure criteria, *Journal of Optimization Theory and Applications*, 134(2): 275–301, 2007, ISSN 0022-3239, DOI: 10.1007/s10957-007-9243-8.

[29] J. P. Blasques, Multi-material topology optimization of laminated composite beams with eigenfrequency constraints, *Composite Structures*, 111:45–55, 2014, ISSN 02638223, DOI: 10.1016/j.compstruct.2013.12.021.

[30] J. F. Bourgat, Numerical experiments of the homogenization method, In *Lecture Notes in Mathematics*, pages 330–356. Springer Berlin Heidelberg, 1979, DOI: 10.1007/bfb0063630.

[31] D. Brissaud and P. Zwolinski, The scientific challenges for a sustainable consumption and production scenario: The circular reuse of materials for the upgrading and repurposing of components, *Procedia CIRP*, 61:663–666, 2017, DOI: 10.1016/j.procir.2016.11.148.

[32] J. T. Cantrell, S. Rohde, D. Damiani, R. Gurnani, L. DiSandro, J. Anton, A. Young, A. Jerez, D. Steinbach, C. Kroese, and P. G. Ifju, Experimental characterization of the mechanical properties of 3D-printed ABS and polycarbonate parts, *Rapid Prototyping Journal*, 23(4):811–824, 2017, ISSN 1355-2546, DOI: 10.1108/RPJ-03-2016-0042.

[33] A. Cerveira, F. Bastos, and V. Real, Semidefinite relaxations and Lagrangean duality in truss topology design problem, *International Journal of Mathematics and Statistics*^{TM}, 9:12–25, 2011, http://www.ceser.in/ceserp/index.php/ijms/article/view/2759.

[34] K. Challis, P. Burchill, A. P. Mouritz, and E. Gellert, Review of advanced composite structures for naval ships and submarines, *Composite Structures*, 53(1):21–42, 2001, DOI: 10.1016/S0263-8223(00)00175-6.

[35] C.-M. Chan, D. E. Grierson, and A. N. Sherbourne, Automatic optimal design of tall steel building frameworks, *Journal of Structural Engineering*, 121(5):838–847, 1995, DOI: 10.1061/(asce)0733-9445(1995)121:5(838).

[36] A. Clausen, N. Aage, and O. Sigmund, Topology optimization of coated structures and material interface problems, *Computer Methods in Applied Mechanics and Engineering*, 290:524–541, 2015, ISSN 00457825, DOI: 10.1016/j.cma.2015.02.011.

[37] A. Clausen, N. Aage, and O. Sigmund, Exploiting additive manufacturing infill in topology optimization for improved buckling load, *Engineering*, 2(2):250–257, 2016, ISSN 20958099, DOI: 10.1016/J.ENG.2016.02.006.

[38] A. Clausen, E. Andreassen, and O. Sigmund, Topology optimization of 3D shell structures with porous infill, *Acta Mechanica Sinica*, 33(4):778–791, 2017, ISSN 0567-7718, DOI: 10.1007/s10409-017-0679-2.

[39] P. G. Coelho and H. C. Rodrigues, Hierarchical topology optimization addressing material design constraints and application to sandwich-type structures, *Structural and Multidisciplinary Optimization*, 52(1):91–104, 2015, ISSN 1615-147X, DOI: 10.1007/s00158-014-1220-x.

[40] P. G. Coelho, P. R. Fernandes, J. M. Guedes, and H. C. Rodrigues, A hierarchical model for concurrent material and topology optimisation of three-dimensional structures, *Structural and Multidisciplinary Optimization*, 35(2):107–115, 2007, DOI: 10.1007/s00158-007-0141-3.

[41] M. F. Cohen, J. Shade, S. Hiller, and O. Deussen, Wang tiles for image and texture generation, *ACM Transactions on Graphics*, 22(3):287–294, 2003, DOI: 10.1145/882262.882265.

[42] J. Conway and J. Lagarias, Tiling with polyominoes and combinatorial group theory, *Journal of Combinatorial Theory, Series A*, 53(2):183–208, 1990, ISSN 00973165, DOI: 10.1016/0097-3165(90)90057-4.

[43] C. Coulais, E. Teomy, K. de Reus, Y. Shokef, and M. van Hecke, Combinatorial design of textured mechanical metamaterials, *Nature*, 535(7613):529–532, 2016, DOI: 10.1038/nature18960.

[44] K. Čulík, An aperiodic set of 13 Wang tiles, *Discrete Mathematics*, 160(1-3):245–251, 1996, DOI: 10.1016/s0012-365x(96)00118-5.

[45] R. E. Curto and L. A. Fialkow, Solution of the truncated complex moment problem for flat data, *Memoirs of the American Mathematical Society*, 119(568), 1996, DOI: 10.1090/memo/0568.

[46] M. Davis, *Computability & Unsolvability*, Dover Books on Computer Science Series. Dover Publications, 1958, ISBN 9780486614717.

[47] S. Daynes, S. Feih, W. F. Lu, and J. Wei, Optimisation of functionally graded lattice structures using isostatic lines, *Materials & Design*, 127:215–223, 2017, ISSN 02641275, DOI: 10.1016/j.matdes.2017.04.082.

[48] E. de Klerk, C. Roos, and T. Terlaky, Semi-definite problems in truss topology optimization, 1995, https://pdfs.semanticscholar.org/dd6b/8a14a816adba3aa31c6c8553ef51f2219c6f.pdf.

[49] E. D. Demaine and M. L. Demaine, Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity, *Graphs and Combinatorics*, 23:195–208, 2007, DOI: 10.1007/s00373-007-0713-4.

[50] A. Derouet-Jourdan, S. Kaji, and Y. Mizoguchi, A linear algorithm for brick Wang tiling, *Japan Journal on Industrial and Applied Mathematics*, 36(3):749–761, 2019, DOI: 10.1007/s13160-019-00369-z.

[51] A. R. Díaaz and N. Kikuchi, Solutions to shape and topology eigenvalue optimization problems using a homogenization method, *International Journal for Numerical Methods in Engineering*, 35(7):1487–1502, 1992, ISSN 0029-5981, DOI: 10.1002/nme.1620350707.

[52] A. Díaz and O. Sigmund, Checkerboard patterns in layout optimization, *Structural Optimization*, 10(1):40–45, 1995, DOI: 10.1007/bf01743693.

[53] W. S. Dorn, Automatic design of optimal structures, *Journal de Mécanique*, 3:25–52, 1964.

[54] M. Doškář and J. Novák, A jigsaw puzzle framework for homogenization of high porosity foams, *Computers & Structures*, 166:33–41, 2016, DOI: 10.1016/j.compstruc.2016.01.003.

[55] M. Doškář and M. Tyburec, Modular topology optimization, https://gitlab.com/MartinDoskar/modular-topology-optimization, 2021.

[56] M. Doškář, J. Novák, and J. Zeman, Aperiodic compression and reconstruction of real-world material systems based on Wang tiles, *Physical Review E*, 90(6):062118, 2014, DOI: 10.1103/PhysRevE.90.062118.

[57] M. Doškář, J. Zeman, D. Jarušková, and J. Novák, Wang tiling aided statistical determination of the Representative Volume Element size of random heterogeneous materials, *European Journal of Mechanics – A/Solids*, 70:280–295, 2018, DOI: 10.1016/j.euromechsol.2017.12.002.

[58] M. Doškář, J. Zeman, D. Rypl, and J. Novák, Level-set based design of Wang tiles for modelling complex microstructures, *Computer-Aided Design*, 123:102827, 2020, DOI: 10.1016/j.cad.2020.102827.

[59] M. Doškář, J. Zeman, P. Krysl, and J. Novák, Microstructure-informed reduced modes synthesized with Wang tiles and the generalized finite element method, *Computational Mechanics*, 68(2):233–253, 2021, DOI: 10.1007/s00466-021-02028-y.

[60] E. N. Dvorkin and K.-J. Bathe, A continuum mechanics based four-node shell element for general non-linear analysis, *Engineering Computations*, 1(1):77–88, 1984, ISSN 0264-4401, DOI: 10.1108/eb023562.

[61] A. Ferrer, J. Cante, J. Hernández, and J. Oliver, Two-scale topology optimization in computational material design: An integrated approach, *Internation Journal of Numerical Methods in Engineering*, 114(3):232–254, 2018, DOI: 10.1002/nme.5742.

[62] J. Fiala, M. Kočvara, and M. Stingl, PENLAB: A MATLAB solver for nonlinear semidefinite optimization, 2013, arXiv: 1311.5240.

[63] M. Franchetti and C. Kress, An economic analysis comparing the cost feasibility of replacing injection molding processes with emerging additive manufacturing techniques, *Internatial Journal of Advanced Manufacturing*, 88(9-12):2573–2579, 2016, DOI: 10.1007/s00170-016-8968-7.

[64] H. Fredricson, Topology optimization of frame structures—Joint penalty and material selection, *Structural and Multidisciplinary Optimization*, 30(3):193–200, 2005, DOI: 10.1007/s00158-005-0515-3.

[65] H. Fredricson, T. Johansen, A. Klarbring, and J. Petersson, Topology optimization of frame structures with flexible joints, *Structural and Multidisciplinary Optimization*, 25 (3):199–214, 2003, DOI: 10.1007/s00158-003-0281-z.

[66] K. Friedrich and A. A. Almajid, Manufacturing aspects of advanced polymer composites for automotive applications, *Applied Composite Materials*, 20(2):107–128, 2013, ISSN 0929189X, DOI: 10.1007/s10443-012-9258-7.

[67] J. Gallier, *Geometric Methods and Applications*. Springer New York, 2011, DOI: 10.1007/978-1-4419-9961-0.

[68] G. Gan, C. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications*. Society for Industrial and Applied Mathematics, second edition, 2020, DOI: 10.1137/1.9781611976335.

[69] J. Gao, Z. Luo, H. Li, and L. Gao, Topology optimization for multiscale design of porous composites with multi-domain microstructures, *Computer Methods in Applied Mechanics and Engineering*, 344:451–476, 2019, DOI: 10.1016/j.cma.2018.10.017.

[70] J. Gao, Z. Luo, H. Li, P. Li, and L. Gao, Dynamic multiscale topology optimization for multi-regional micro-structured cellular composites, *Composite Structures*, 211:401–417, 2019, DOI: 10.1016/j.compstruct.2018.12.031.

[71] E. Garner, H. M. Kolken, C. C. Wang, A. A. Zadpoor, and J. Wu, Compatibility in microstructural optimization for additive manufacturing, *Additive Manufacturing*, 26: 65–75, 2019, DOI: 10.1016/j.addma.2018.12.007.

[72] P. Geoffroy-Donders, G. Allaire, and O. Pantz, 3-d topology optimization of modulated and oriented periodic microstructures by the homogenization method, *Journal of Computational Physics*, 401:108994, 2020, DOI: 10.1016/j.jcp.2019.108994.

[73] H. Ghiasi, D. Pasini, and L. Lessard, Optimum stacking sequence design of composite materials Part I: Constant stiffness design, *Composite Structures*, 90(1):1–11, 2009, ISSN 02638223, DOI: 10.1016/j.compstruct.2009.01.006.

[74] H. Ghiasi, K. Fayazbakhsh, D. Pasini, and L. Lessard, Optimum stacking sequence design of composite materials Part II: Variable stiffness design, *Composite Structures*, 93(1):1–13, 2010, ISSN 02638223, DOI: 10.1016/j.compstruct.2010.06.001.

[75] J. P. Groen and O. Sigmund, Homogenization-based topology optimization for high-resolution manufacturable microstructures, *International Journal for Numerical Methods in Engineering*, 113(8):1148–1163, 2018, DOI: 10.1002/nme.5575.

[76] J. P. Groen, J. Wu, and O. Sigmund, Homogenization-based stiffness optimization and projection of 2D coated structures with orthotropic infill, *Computer Methods in Applied Mechanics and Engineering*, 349:722–742, 2019, DOI: 10.1016/j.cma.2019.02.031.

[77] J. P. Groen, F. C. Stutz, N. Aage, J. A. Bærentzen, and O. Sigmund, De-homogenization of optimal multi-scale 3D topologies, *Computer Methods in Applied Mechanics and Engineering*, 364:112979, 2020, DOI: 10.1016/j.cma.2020.112979.

[78] A. A. Groenwold and L. F. P. Etman, A simple heuristic for gray-scale suppression in optimality criterion-based topology optimization, *Structural and Multidisciplinary Optimization*, 39(2):217–225, 2009, DOI: 10.1007/s00158-008-0337-1.

[79] B. Grünbaum and G. C. Shephard, *Tilings and patterns*. Dover Publications, Inc, Mineola, New York, 2016.

[80] G. X. Gu, C.-T. Chen, D. J. Richmond, and M. J. Buehler, Bioinspired hierarchical composite design using machine learning: Simulation, additive manufacturing, and experiment, *Materials Horizons*, 5(5):939–945, 2018, DOI: 10.1039/C8MH00653A.

[81] G. Gurtner and M. Durand, Stiffest elastic networks, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 470:20130611, 2014, DOI: 10.1098/rspa.2013.0611.

[82] S. Habibi, A. Kavand, M. Kočvara, and M. Stingl, Barrier and penalty methods for low-rank semidefinite programming with application to truss topology design, 2021, arXiv: 2105.08529.

[83] M. Heidari, R. Cogill, P. Allaire, and P. Sheth, Optimization of peak power in vibrating structures via semidefinite programming, In *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, 2009, DOI: 10.2514/6.2009-2181.

[84] M. Helou and S. Kara, Design, analysis and manufacturing of lattice structures: An overview, *International Journal of Computer Integrated Manufacturing*, 31(3):243–261, 2018, ISSN 13623052, DOI: 10.1080/0951192X.2017.1407456.

[85] D. Henrion and J.-B. Lasserre, Convergent relaxations of polynomial matrix inequalities and static output feedback, *IEEE Transactions on Automatic Control*, 51(2):192–202, 2006, DOI: 10.1109/tac.2005.863494.

[86] S. Hiller, O. Deussen, and A. Keller, Tiled blue noise samples, In *Proceedings of the Vision Modeling and Visualization Conference 2001*, VMV '01, pages 265–272, Stuttgart, Germany, 2001. Aka GmbH, ISBN 3-89838-028-9.

[87] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1992.

[88] J. Hu, M. Li, X. Yang, and S. Gao, Cellular structure design based on free material optimization under connectivity control, *Computer-Aided Design*, page 102854, 2020, DOI: 10.1016/j.cad.2020.102854.

[89] X. Huang and Y. M. Xie, Optimal design of periodic structures using evolutionary topology optimization, *Structural and Multidisciplinary Optimization*, 36(6):597–606, 2008, DOI: 10.1007/s00158-007-0196-1.

[90] A. Ion, L. Wall, R. Kovacs, and P. Baudisch, Digital mechanical metamaterials, In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 977–988. ACM, 2017, DOI: 10.1145/3025453.3025624.

[91] E. Irving, *Polymer composites in the aerospace industry*. Woodhead Publishing, Kidlington, UK, 2015, ISBN 978-0-85709-523-7, DOI: 10.1016/C2013-0-16303-9.

[92] E. Jeandel and M. Rao, An aperiodic set of 11 Wang tiles, *Advances in Combinatorics*, (2021:1):1–37, 2021, ISSN 2517-5599, DOI: 10.19086/aic.18614.

[93] M. Jílek, M. Somr, M. Kulich, J. Zeman, and L. Přeučil, Towards a passive self-assembling macroscale multi-robot system, *IEEE Robotics and Automation Letters*, 6 (4):7293–7300, 2021, DOI: 10.1109/LRA.2021.3096748.

[94] A. S. Kahr, E. F. Moore, and H. Wang, Entscheidungsproblem reduced to the ∀∃∀ case, *Proceedings of the National Academy of Sciences*, 48(3):365–377, 1962, DOI: 10.1073/pnas.48.3.365.

[95] Y. Kanno, *Nonsmooth mechanics and convex optimization*. CRC Press, Boca Raton, FL, 2011, ISBN 9781138072787.

[96] Y. Kanno, Mixed-integer second-order cone programming for global optimization of compliance of frame structure with discrete design variables, *Structural and Multidisciplinary Optimization*, 54(2):301–316, 2016, DOI: 10.1007/s00158-016-1406-5.

[97] J. Kari, Reversibility of 2D cellular automata is undecidable, *Physica D: Nonlinear Phenomena*, 45(1-3):379–385, 1990, DOI: 10.1016/0167-2789(90)90195-u.

[98] J. Kari, A small aperiodic set of Wang tiles, *Discrete Mathematics*, 160(1-3):259–264, 1996, DOI: 10.1016/0012-365x(95)00120-l.

[99] J. Kari and P. Papasoglu, Deterministic aperiodic tile sets, *Geometric And Functional Analysis*, 9(2):353–369, 1999, DOI: 10.1007/s000390050090.

[100] M. R. Khan, Optimality criterion techniques applied to frames having general cross-sectional relationships, *AIAA Journal*, 22(5):669–676, 1984, DOI: 10.2514/3.8653.

[101] S. Kim, M. Kojima, M. Mevissen, and M. Yamashita, Exploiting sparsity in linear and nonlinear matrix inequalities via positive semidefinite matrix completion, *Mathematical Programming*, 129(1):33–68, 2011, ISSN 0025-5610, DOI: 10.1007/s10107-010-0402-6.

[102] Y. Y. Kim and T. S. Kim, Topology optimization of beam cross sections, *International Journal of Solids and Structures*, 37(3):477–493, 2000, ISSN 00207683, DOI: 10.1016/S0020-7683(99)00015-3.

[103] D. E. Knuth, *The Art of Computer Programming, Volume 1: Fundamental Algorithms*. Addison-Wesley Educational Publishers Inc, 1968, ISBN 9780201038019.

[104] D. E. Knuth, Two notes on notation, *The American Mathematical Monthly*, 99(5): 403–422, 1992, DOI: 10.1080/00029890.1992.11995869.

[105] D. E. Knuth, *The Art of Computer Programming, Volume 4B, Fascicle 5: The: Mathematical Preliminaries Redux; Backtracking; Dancing Links*. Addison-Wesley Professional, 2018, ISBN 978-0134671796.

[106] M. Kočvara, On the modelling and solving of the truss design problem with global stability constraints, *Structural and Multidisciplinary Optimization*, 23(3):189–203, 2002, ISSN 1615147X, DOI: 10.1007/s00158-002-0177-3.

[107] M. Kočvara, Decomposition of arrow type positive semidefinite matrices with application to topology optimization, *Mathematical Programming*, 190(1-2):105–134, 2020, DOI: 10.1007/s10107-020-01526-w.

[108] M. Kočvara and J. V. Outrata, Effective reformulations of the truss topology design problem, *Optimization and Engineering*, 7(2):201–219, 2006, DOI: 10.1007/s11081-006-6839-z.

[109] M. Kočvara and M. Stingl, PENNON: A code for convex nonlinear and semidefinite programming, *Optimization Methods and Software*, 18(3):317–333, 2003, DOI: 10.1080/1055678031000098773.

[110] M. Kočvara, M. Stingl, and J. Zowe, Free material optimization: Recent progress, *A Journal of Mathematical Programming and Operations Research*, 57(1):79–100, 2008, DOI: 10.1080/02331930701778908.

[111] R. V. Kohn and G. Strang, Optimal design and relaxation of variational problems, I, *Communications on Pure and Applied Mathematics*, 39(1):113–137, 1986, DOI: 10.1002/cpa.3160390107.

[112] M. Kojima, Sums of squares relaxations of polynomial semidefinite programs, Technical Report B-397, Department of Mathematical and Computing Science, Tokyo Institute of Technology, Japan, 2003.

[113] M. Kojima and M. Muramatsu, An extension of sums of squares relaxations to polynomial optimization problems over symmetric cones, *Mathematical Programming*, 110(2): 315–336, 2006, DOI: 10.1007/s10107-006-0004-5.

[114] S. Z. Kovalsky, D. Glasner, and R. Basri, A global approach for solving edge-matching puzzles, *SIAM Journal on Imaging Sciences*, 8(2):916–938, 2015, ISSN 1936-4954, DOI: 10.1137/140987869.

[115] P. Kovanic, On the pseudoinverse of a sum of symmetric matrices with applications to estimation, *Kybernetika*, 15(5):341–348, 1979.

[116] T. Kumar and K. Suresh, A density-and-strain-based K-clustering approach to microstructural topology optimization, *Structural and Multidisciplinary Optimization*, 61 (4):1399–1415, 2019, DOI: 10.1007/s00158-019-02422-4.

[117] S. Labbé, A self-similar aperiodic set of 19 Wang tiles, *Geometriae Dedicata*, 201(1): 81–109, 2019, DOI: 10.1007/s10711-018-0384-8.

[118] A. Lagae and P. Dutré, A procedural object distribution function, *ACM Transactions on Graphics*, 24(4):1442–1461, 2005, DOI: 10.1145/1095878.1095888.

[119] A. Lagae and P. Dutré, An alternative for Wang tiles: Colored edges versus colored corners, *ACM Transactions on Graphics*, 25:1442–1459, 2006, DOI: 10.1145/1183287.1183296.

[120] A. Lagae and P. Dutré, The tile packing problem, *Geombinatorics*, 17(1):8–18, 2007.

[121] A. Lagae, J. Kari, and P. Dutré, Aperiodic sets of square tiles with colored corners, 2006, http://graphics.cs.kuleuven.be/publications/LKD06ASSTCC/LKD06ASSTCC.pdf.

[122] J. B. Lasserre, *An introduction to polynomial and semi-algebraic optimization*. Cambridge University Press, 2015, DOI: 10.1017/cbo9781107447226.

[123] A. C. Lemonge, H. J. Barbosa, A. L. Coutinho, and C. C. Borges, Multiple cardinality constraints and automatic member grouping in the optimal design of steel framed structures, *Engineering Structures*, 33(2):433–444, 2011, DOI: 10.1016/j.engstruct.2010.10.026.

[124] H. Lewis and C. Papimitriou, *Elements of the theory of computation*, Prentice-Hall Software Series. Pearson Education Canada, 1981, ISBN 9780132734172.

[125] H. R. Lewis, *Complexity of solvable cases of the decision problem for the predicate calculus*, pages 35–47. IEEE, 1978, DOI: 10.1109/SFCS.1978.9.

[126] H. Li, Z. Luo, L. Gao, and Q. Qin, Topology optimization for concurrent design of structures with multi-patch microstructures by level sets, *Computer Methods in Applied Mechanics and Engineering*, 331:536–561, 2018, DOI: 10.1016/j.cma.2017.11.033.

[127] C. Liu, Z. Du, W. Zhang, Y. Zhu, and X. Guo, Additive manufacturing-oriented design of graded lattice structures through explicit topology optimization, *Journal of Applied Mechanics*, 84(8):081008, 2017, DOI: 10.1115/1.4036941.

[128] L. Liu, J. Yan, and G. Cheng, Optimum structure with homogeneous optimum truss-like material, *Computers & Structures*, 86(13–14):1417–1425, 2008, DOI: 10.1016/j.compstruc.2007.04.030.

[129] M. Livesu, S. Ellero, J. Martínez, S. Lefebvre, and M. Attene, From 3D models to 3D prints: An overview of the processing pipeline, *Computer Graphics Forum*, 36(2):537–564, 2017, ISSN 14678659, DOI: 10.1111/cgf.13147.

[130] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, Applications of second-order cone programming, *Linear Algebra and its Applications*, 284(1-3):193–228, 1998, DOI: 10.1016/s0024-3795(98)10032-0.

[131] J. Löfberg, YALMIP: A toolbox for modeling and optimization in MATLAB, In *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[132] Y. Lu and L. Tong, Concurrent topology optimization of cellular structures and anisotropic materials, *Computers & Structures*, 255:106624, 2021, DOI: 10.1016/j.compstruc.2021.106624.

[133] J. Luo and H. C. Gea, A systematic topology optimization approach for optimal stiffener design, *Structural Optimization*, 16(4):280–288, 1998, ISSN 0934-4373, DOI: 10.1007/BF01271435.

[134] Z. D. Ma, N. Kikuchi, and I. Hagiwara, Structural topology and shape optimization for a frequency response problem, *Computational Mechanics*, 13(3):157–174, 1993, ISSN 0178-7675, DOI: 10.1007/BF00370133.

[135] K. Maute and M. Allen, Conceptual design of aeroelastic structures by topology optimization, *Structural and Multidisciplinary Optimization*, 27(1-2):27–42, 2004, ISSN 1615-147X, DOI: 10.1007/s00158-003-0362-z.

[136] A. Michell, The limits of economy of material in frame-structures, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 8(47):589–597, 1904, DOI: 10.1080/14786440409463229.

[137] J. H. Mikkola and O. Gassmann, Managing modularity of product architectures: Toward an integrated theory, *IEEE Trans. Eng. Manage.*, 50(2):204–218, 2003, DOI: 10.1109/TEM.2003.810826.

[138] MOSEK ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 8.1.*, 2017, http://docs.mosek.com/8.1/toolbox/index.html.

[139] S. Mozes, Tilings, substitution systems and dynamical systems generated by them, *Journal d'Analyse Mathématique*, 53(1):139–186, 1989, ISSN 0021-7670, DOI: 10.1007/BF02793412.

[140] K. Murota, Y. Kanno, M. Kojima, and S. Kojima, A numerical algorithm for block-diagonal decomposition of matrix $*$-algebras with application to semidefinite programming, *Japan Journal of Industrial and Applied Mathematics*, 27(1):125–160, 2010, DOI: 10.1007/s13160-010-0006-9.

[141] V. Nežerka, M. Somr, T. Janda, J. Vorel, M. Doškář, J. Antoš, J. Zeman, and J. Novák, A jigsaw puzzle metamaterial concept, *Composite Structures*, 202:1275–1279, 2018, DOI: 10.1016/j.compstruct.2018.06.015.

[142] H.-D. Nguyen, G.-W. Jang, D.-M. Kim, and Y. Y. Kim, Finite prism method based topology optimization of beam cross section for buckling load maximization, *Structural and Multidisciplinary Optimization*, 57(1):55–70, 2018, ISSN 1615-147X, DOI: 10.1007/s00158-017-1860-8.

[143] J. Novák, A. Kučerová, and J. Zeman, Compressing random microstructures via stochastic Wang tilings, *Physical Review E*, 86(4):4–7, 2012, DOI: 10.1103/PhysRevE.86.040104.

[144] J. Novák, A. Kučerová, and J. Zeman, Microstructural enrichment functions based on stochastic Wang tilings, *Modelling and Simulation in Materials Science and Engineering*, 21(2):025014, 2013, DOI: 10.1088/0965-0393/21/2/025014.

[145] T. Nurmi, From checkerboard to cloverfield: Using Wang tiles in seamless non-periodic patterns, In *Bridges Finland Conference Proceedings*, 2016.

[146] M. Ohsaki, K. Fujisawa, N. Katoh, and Y. Kanno, Semi-definite programming for topology optimization of trusses under multiple eigenvalue constraints, *Computer Methods in Applied Mechanics and Engineering*, 180(1-2):203–217, 1999, ISSN 00457825, DOI: 10.1016/S0045-7825(99)00056-0.

[147] N. Olhoff and J. E. Taylor, On structural optimization, *Journal of Applied Mechanics*, 50(4b):1139, 1983, ISSN 00218936, DOI: 10.1115/1.3167196.

[148] G. Oliveri and J. T. B. Overvelde, Inverse design of mechanical metamaterials that undergo buckling, *Advanced Functional Materials*, 30(12):1909033, 2020, DOI: 10.1002/adfm.201909033.

[149] E. A. Olivetti and J. M. Cullen, Toward a sustainable materials system, *Science*, 360 (6396):1396–1398, 2018, DOI: 10.1126/science.aat6821.

[150] N. Ollinger, Two-by-two substitution systems and the undecidability of the domino problem, In *Logic and Theory of Algorithms*, pages 476–485. Springer Berlin Heidelberg, 2008, DOI: 10.1007/978-3-540-69407-6_51.

[151] M. M. Opgenoord and K. E. Willcox, Aeroelastic tailoring using additively manufactured lattice structures, In *2018 Multidisciplinary Analysis and Optimization Conference*, Reston, Virginia, 2018. American Institute of Aeronautics and Astronautics, ISBN 978-1-62410-550-0, DOI: 10.2514/6.2018-4055.

[152] O. Pantz and K. Trabelsi, A post-treatment of the homogenization method for shape optimization, *SIAM Journal on Control and Optimization*, 47(3):1380–1398, 2008, DOI: 10.1137/070688900.

[153] S. Pauliuk, N. Heeren, P. Berrill, T. Fishman, A. Nistad, Q. Tu, P. Wolfram, and E. G. Hertwich, Global scenarios of resource and emission savings from material efficiency in residential buildings and cars, *Nature Communications*, 12(1), 2021, DOI: 10.1038/s41467-021-25300-4.

[154] B. Poizat, Une théorie finiement axiomatisable et superstable, *Groupe d'étude des théories stables*, 3(1):1–9, 1980.

[155] G. Prathap, The poor bending response of the four-node plane stress quadrilateral, *International Journal for Numerical Methods in Engineering*, 21(5):825–835, 1985, DOI: 10.1002/nme.1620210505.

[156] Z. A. Provchy, A. N. Palazotto, and P. J. Flater, Topology optimization for projectile design, *Journal of Dynamic Behavior of Materials*, 4(1):129–137, 2018, ISSN 2199-7446, DOI: 10.1007/s40870-018-0143-9.

[157] Z. Qiu, Q. Li, S. Liu, and R. Xu, Clustering-based concurrent topology optimization with macrostructure, components, and materials, *Structural and Multidisciplinary Optimization*, 2020, DOI: 10.1007/s00158-020-02755-5.

[158] C. Radin, Low temperature and the origin of crystalline symmetry, *International Journal of Modern Physics B*, 1(05n06):1157–1191, 1987, ISSN 0217-9792, DOI: 10.1142/S0217979287001675.

[159] M. Rais-Rohani and J. Lokits, Reinforcement layout and sizing optimization of composite submarine sail structures, *Structural and Multidisciplinary Optimization*, 34(1):75–90, 2007, ISSN 1615-147X, DOI: 10.1007/s00158-006-0066-2.

[160] R. M. Robinson, Seven polygons which permit only nonperiodic tilings of the plane, *Notices of American Mathematical Society*, (14):835, 1967.

[161] R. M. Robinson, Undecidability and nonperiodicity for tilings of the plane, *Inventiones Mathematicae*, 12(3):177–209, 1971, DOI: 10.1007/bf01418780.

[162] R. M. Robinson, Undecidable tiling problems in the hyperbolic plane, *Inventiones Mathematicae*, 44(3):259–264, 1978, DOI: 10.1007/bf01403163.

[163] H. Rodrigues, J. M. Guedes, and M. P. Bendsøe, Hierarchical optimization of material and structure, *Structural and Multidisciplinary Optimization*, 24(1):1–10, 2002, DOI: 10.1007/s00158-002-0209-z.

[164] G. I. N. Rozvany, *Structural design via optimality criteria*. Springer Netherlands, 1989, DOI: 10.1007/978-94-009-1161-1.

[165] G. I. N. Rozvany, T. Sokół, and V. Pomezanski, Fundamentals of exact multi-load topology optimization – stress-based least-volume trusses (generalized Michell structures) – Part I: Plastic design, *Structural and Multidisciplinary Optimization*, 50(6):1051–1078, 2014, DOI: 10.1007/s00158-014-1118-7.

[166] C. R. Rui Yu and L. Agapito, Solving jigsaw puzzles with linear programming, In E. R. H. Richard C. Wilson and W. A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 139.1–139.12. BMVA Press, 2016, ISBN 1-901725-59-6, DOI: 10.5244/C.30.139.

[167] M. Saka, Optimum design of rigidly jointed frames, *Computers & Structures*, 11(5): 411–419, 1980, DOI: 10.1016/0045-7949(80)90107-8.

[168] F. Salassa, W. Vancroonenburg, T. Wauters, F. Della Croce, and G. V. Berghe, MILP and Max-Clique based heuristics for the Eternity II puzzle, 2017, arXiv: 1709.00252.

[169] C. Schumacher, B. Bickel, J. Rys, S. Marschner, C. Daraio, and M. Gross, Microstructures to control elasticity in 3D printing, *ACM Transactions on Graphics*, 34(4), 2015, DOI: 10.1145/2766926.

[170] F. Schury, M. Stingl, and F. Wein, Efficient two-scale optimization of manufacturable graded structures, *SIAM Journal on Scientific Computing*, 34(6):B711–B733, 2012, DOI: 10.1137/110850335.

[171] N. C. Seeman, C. Mao, T. H. LaBean, and J. H. Reif, Logical computation using algorithmic self-assembly of DNA triple-crossover molecules, *Nature*, 407(6803):493–496, 2000, ISSN 00280836, DOI: 10.1038/35035038.

[172] M. Senechal, *Quasicrystals and Geometry*. Cambridge University Press, 1996, ISBN 9780521575416.

[173] J. E. Sharpe, Computer tools for integrated conceptual design, *Design Studies*, 16(4): 471–488, 1995, ISSN 0142694X, DOI: 10.1016/0142-694X(95)00021-I.

[174] K. Shea, J. Cagan, and S. J. Fenves, A shape annealing approach to optimal truss design with dynamic grouping of members, *Journal of Mechanical Design*, 119(3):388–394, 1997, DOI: 10.1115/1.2826360.

[175] O. Sigmund, Materials with prescribed constitutive parameters: An inverse homogenization problem, *International Journal of Solids and Structures*, 31(17):2313–2329, 1994, DOI: 10.1016/0020-7683(94)90154-6.

[176] O. Sigmund, Tailoring materials with prescribed elastic properties, *Mechanics of Materials*, 20(4):351–368, 1995, DOI: 10.1016/0167-6636(94)00069-7.

[177] O. Sigmund, On the design of compliant mechanisms using topology optimization, *Mechanics of Structures and Machines*, 25(4):493–524, 1997, DOI: 10.1080/08905459708945415.

[178] O. Sigmund, Morphology-based black and white filters for topology optimization, *Structural and Multidisciplinary Optimization*, 33(4-5):401–424, 2007, ISSN 1615-147X, 1615-1488, DOI: 10.1007/s00158-006-0087-x.

[179] R. Sivapuram, P. D. Dunning, and H. A. Kim, Simultaneous material and structural optimization by multiscale topology optimization, *Structural and Multidisciplinary Optimization*, 54(5):1267–1281, 2016, DOI: 10.1007/s00158-016-1519-x.

[180] H. Spencer, *The Principles of Biology*, Spencer, Herbert: A system of synthetic philosophy. Williams and Norgate, 1864.

[181] J. Stam, *Aperiodic texture mapping*. European Research Consortium for Informatics and Mathematics, 1997.

[182] B. K. Stanford and P. D. Dunning, Optimal topology of aircraft rib and spar structures under aeroelastic loads, *Journal of Aircraft*, 52(4):1298–1311, 2015, ISSN 0021-8669, DOI: 10.2514/1.C032913.

[183] L. L. Stromberg, A. Beghini, W. F. Baker, and G. H. Paulino, Application of layout and topology optimization using pattern gradation for the conceptual design of buildings, *Structural and Multidisciplinary Optimization*, 43(2):165–180, 2010, DOI: 10.1007/s00158-010-0563-1.

[184] K. Svanberg, The method of moving asymptotes—a new method for structural optimization, *International Journal for Numerical Methods in Engineering*, 24(2):359–373, 1987, DOI: 10.1002/nme.1620240207.

[185] G. Sved, The minimum weight of certain redundant structures, *Australian Journal of Applied Science*, 5:1–9, 1954.

[186] V. Toğan and A. T. Daloğlu, An improved genetic algorithm with initial population strategy and self-adaptive member grouping, *Computers & Structures*, 86(11–12):1204–1218, 2008, DOI: 10.1016/j.compstruc.2007.11.006.

[187] A. Tugilimana, A. P. Thrall, and R. F. Coelho, Conceptual design of modular bridges including layout optimization and component reusability, *Journal of Bridge Engineering*, 22(11):04017094, 2017, DOI: 10.1061/(asce)be.1943-5592.0001138.

[188] A. Tugilimana, A. P. Thrall, B. Descamps, and R. F. Coelho, Spatial orientation and topology optimization of modular trusses, *Structural and Multidisciplinary Optimization*, 55(2):459–476, 2017, DOI: 10.1007/s00158-016-1501-7.

[189] A. Tugilimana, R. F. Coelho, and A. P. Thrall, An integrated design methodology for modular trusses including dynamic grouping, module spatial orientation, and topology optimization, *Structural and Multidisciplinary Optimization*, 60(2):613–638, 2019, DOI: 10.1007/s00158-019-02230-w.

[190] A. M. Turing, On computable numbers, with an application to the Entscheidungsproblem, *Proceedings of the London Mathematical Society*, s2–42:230–265, 1937, DOI: 10.1112/plms/s2-42.1.230.

[191] M. Tyburec, J. Zeman, J. Novák, M. Lepš, T. Plachý, and R. Poul, Designing modular 3D printed reinforcement of wound composite hollow beams with semidefinite programming, *Materials & Design*, 183:108131, 2019, DOI: 10.1016/j.matdes.2019.108131.

[192] M. Tyburec, J. Zeman, M. Doškář, M. Kružík, and M. Lepš, Modular-topology optimization with Wang tilings: an application to truss structures, *Structural and Multidisciplinary Optimization*, 63(3):1099–1117, 2020, DOI: 10.1007/s00158-020-02744-8.

[193] M. Tyburec, J. Zeman, M. Doškář, M. Kružík, and M. Lepš, Source codes for preprint Modular-topology optimization with Wang tilings: An application to truss structures, 2020, DOI: 10.5281/zenodo.3835555.

[194] M. Tyburec, J. Zeman, M. Kružík, and D. Henrion, On optimum design of frame structures, *Acta Polytech. CTU Proc.*, 26:117–125, 2020, DOI: 10.14311/app.2020.26.0117.

[195] M. Tyburec, J. Zeman, M. Kružík, and D. Henrion, Source codes for preprint Global optimality in minimum compliance topology optimization of frames and shells by moment-sum-of-squares hierarchy, 2020, DOI: 10.5281/zenodo.4048828.

[196] M. Tyburec, J. Zeman, M. Kružík, and D. Henrion, Global optimality in minimum compliance topology optimization of frames and shells by moment-sum-of-squares hierarchy, *Structural and Multidisciplinary Optimization*, 64(4):1963–1981, 2021, DOI: 10.1007/s00158-021-02957-5.

[197] L. Vandenberghe and S. Boyd, Semidefinite programming, *SIAM Review*, 38(1):49–95, 1996, ISSN 0036-1445, DOI: 10.1137/1038003.

[198] V. V. Vasiliev, V. A. Barynin, and A. F. Razin, Anisogrid composite lattice structures – Development and aerospace applications, *Composite Structures*, 94(3):1117–1127, 2012, ISSN 02638223, DOI: 10.1016/j.compstruct.2011.10.023.

[199] J. R. Vinson, Structural optimization to obtain minimum weight sandwich panels, In *Plate and Panel Structures of Isotropic, Composite and Piezoelectric Materials, Including Sandwich Construction*, volume 120 of *Solid Mechanics and Its Applications*, chapter 17, pages 345–377. Springer, Dordrecht, 2005, DOI: 10.1007/1-4020-3111-4_17.

[200] A.-J. Wang and D. McDowell, Optimization of a metal honeycomb sandwich beam-bar subjected to torsion and bending, *International Journal of Solids and Structures*, 40(9): 2085–2099, 2003, ISSN 00207683, DOI: 10.1016/S0020-7683(03)00033-7.

[201] H. Wang, Proving theorems by pattern recognition–II, *Bell System Technical Journal*, 40(1):1–41, 1961, DOI: 10.1002/j.1538-7305.1961.tb03975.x.

[202] H. Wang, Dominoes and the AEA case of the decision problem, *Symposium on the Mathematical Theory of Automata*, pages 23–55, 1963.

[203] H. Wang, Games, logic and computers, In *Computation, Logic, Philosophy*, pages 195–217. Springer Netherlands, 1965, DOI: 10.1007/978-94-009-2356-0_10.

[204] H. Wang, Notes on a class of tiling problems, *Fundamenta Mathematicae*, 82(4):295–305, 1975.

[205] L. Wang, P. K. Basu, and J. P. Leiva, Automobile body reinforcement by finite element optimization, *Finite Elements in Analysis and Design*, 40(8):879–893, 2004, ISSN 0168874X, DOI: 10.1016/S0168-874X(03)00118-5.

[206] Q. Wang and J. S. Arora, Alternative formulations for structural optimization: An evaluation using frames, *Journal of Structural Engineering*, 132(12):1880–1889, 2006, DOI: 10.1061/(asce)0733-9445(2006)132:12(1880).

[207] Y. Wang, L. Zhang, S. Daynes, H. Zhang, S. Feih, and M. Y. Wang, Design of graded lattice structure with optimized mesostructures for additive manufacturing, *Materials & Design*, 142:114–123, 2018, ISSN 02641275, DOI: 10.1016/j.matdes.2018.01.011.

[208] E. Winfree, Algorithmic self-assembly of DNA: Theoretical motivations and 2D assembly experiments, *Journal of Biomolecular Structure and Dynamics*, 17(SUPPL. 1):263–270, 2000, DOI: 10.1080/07391102.2000.10506630.

[209] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman, Design and self-assembly of two-dimensional DNA crystals, *Nature*, 394(6693):539–544, 1998, DOI: 10.1038/28998.

[210] J. Wu, A. Clausen, and O. Sigmund, Minimum compliance topology optimization of shell-infill composites for additive manufacturing, *Computer Methods in Applied Mechanics and Engineering*, 326:358–375, 2017, ISSN 00457825, DOI: 10.1016/j.cma.2017.08.018.

[211] J. Wu, N. Aage, R. Westermann, and O. Sigmund, Infill optimization for additive manufacturing–approaching bone-like porous structures, *IEEE Transactions on Visualization and Computer Graphics*, 24(2):1127–1140, 2018, ISSN 1077-2626, DOI: 10.1109/TVCG.2017.2655523.

[212] J. Wu, O. Sigmund, and J. P. Groen, Topology optimization of multi-scale structures: A review, *Structural and Multidisciplinary Optimization*, 63(3):1455–1480, 2021, DOI: 10.1007/s00158-021-02881-8.

[213] K. Wu, O. Sigmund, and J. Du, Design of metamaterial mechanisms using robust topology optimization and variable linking scheme, *Structural and Multidisciplinary Optimization*, 63(4):1975–1988, 2021, DOI: 10.1007/s00158-020-02791-1.

[214] L. Wu, L. Liu, Y. Wang, Z. Zhai, H. Zhuang, D. Krishnaraju, Q. Wang, and H. Jiang, A machine learning-based method to design modular metamaterials, *Extreme Mechanics Letters*, 36:100657, 2020, DOI: 10.1016/j.eml.2020.100657.

[215] L. Xu and G. Cheng, Two-scale concurrent topology optimization with multiple micro materials based on principal stress orientation, *Structural and Multidisciplinary Optimization*, 57(5):2093–2107, 2018, DOI: 10.1007/s00158-018-1916-4.

[216] M. Xu and Z. Qiu, Free vibration analysis and optimization of composite lattice truss core sandwich beams with interval parameters, *Composite Structures*, 106:85–95, 2013, ISSN 02638223, DOI: 10.1016/j.compstruct.2013.05.048.

[217] S. Yamada and Y. Kanno, Relaxation approach to topology optimization of frame structure under frequency constraint, *Structural and Multidisciplinary Optimization*, 53 (4):731–744, 2015, DOI: 10.1007/s00158-015-1353-6.

[218] J. Yan, Q. Sui, Z. Fan, Z. Duan, and T. Yu, Clustering-based multiscale topology optimization of thermo-elastic lattice structures, *Computational Mechanics*, 66(4): 979–1002, 2020, DOI: 10.1007/s00466-020-01892-4.

[219] W. Yang, Q. Liu, Z. Gao, Z. Yue, and B. Xu, Theoretical search for heterogeneously architected 2D structures, *Proceedings of the National Academy of Sciences*, 115(31): E7245–E7254, 2018, DOI: 10.1073/pnas.1806769115.

[220] X. Yang and M. Li, Free isotropic material optimization via second order cone programming, *Computer-Aided Design*, 115:52–63, 2019, DOI: 10.1016/j.cad.2019.05.002.

[221] H. Yasuda, P. R. Buskohl, A. Gillman, T. D. Murphey, S. Stepney, R. A. Vaia, and J. R. Raney, Mechanical computing, *Nature*, 598(7879):39–48, 2021, DOI: 10.1038/s41586-021-03623-y.

[222] Y. L. Young, M. R. Motley, R. Barber, E. J. Chae, and N. Garg, Adaptive composite marine propulsors and turbines: Progress and challenges, *Applied Mechanics Reviews*, 68(6):060803, 2016, ISSN 0003-6900, DOI: 10.1115/1.4034659.

[223] M. Zawidzki and Ł. Jankowski, Multiobjective optimization of modular structures: Weight versus geometric versatility in a Truss-Z system, *Computer-Aided Civil and Infrastructure Engineering*, 34(11):1026–1040, 2019, DOI: 10.1111/mice.12478.

[224] M. Zawidzki and K. Nishinari, Modular Truss-Z system for self-supporting skeletal free-form pedestrian networks, *Advances in Engineering Software*, 47(1):147–159, 2012, DOI: 10.1016/j.advengsoft.2011.12.012.

[225] J. Zeman and M. Šejnoha, From random microstructures to representative volume elements, *Modelling and Simulation in Materials Science and Engineering*, 15(4):S325–S335, 2007, DOI: 10.1088/0965-0393/15/4/S01.

[226] X. Zhang and Y. J. Kim, Efficient texture synthesis using strict Wang tiles, *Graphical Models*, 70(3):43–56, 2008, ISSN 15240703, DOI: 10.1016/j.gmod.2007.10.002.

[227] Y. Zhang, M. Xiao, H. Li, L. Gao, and S. Chu, Multiscale concurrent topology optimization for cellular structures with multiple microstructures based on ordered SIMP interpolation, *Computational Materials Science*, 155:74–91, 2018, DOI: 10.1016/j.commatsci.2018.08.030.

[228] J. Zhu, H. Zhou, C. Wang, L. Zhou, S. Yuan, and W. Zhang, A review of topology optimization for additive manufacturing: Status and challenges, *Chinese Journal of Aeronautics*, 34(1):91–110, 2021, DOI: 10.1016/j.cja.2020.09.020.

[229] Y. Zhu, S. Li, Z. Du, C. Liu, X. Guo, and W. Zhang, A novel asymptotic-analysis-based homogenisation approach towards fast design of infill graded microstructures, *Journal of the Mechanics and Physics of Solids*, 124:612–633, 2019, ISSN 00225096, DOI: 10.1016/j.jmps.2018.11.008.

[230] O. Zienkiewicz, R. Taylor, and J. Zhu, editors, *The finite element method: Its basis and fundamentals*. Butterworth-Heinemann, Oxford, seventh edition, 2013, ISBN 978-1-85617-633-0, DOI: 10.1016/c2009-0-24909-9.

[231] J. Zowe, M. Kočvara, and M. P. Bendsøe, Free material optimization via mathematical programming, *Mathematical Programming*, 79(1-3):445–466, 1997, DOI: 10.1007/BF02614328.