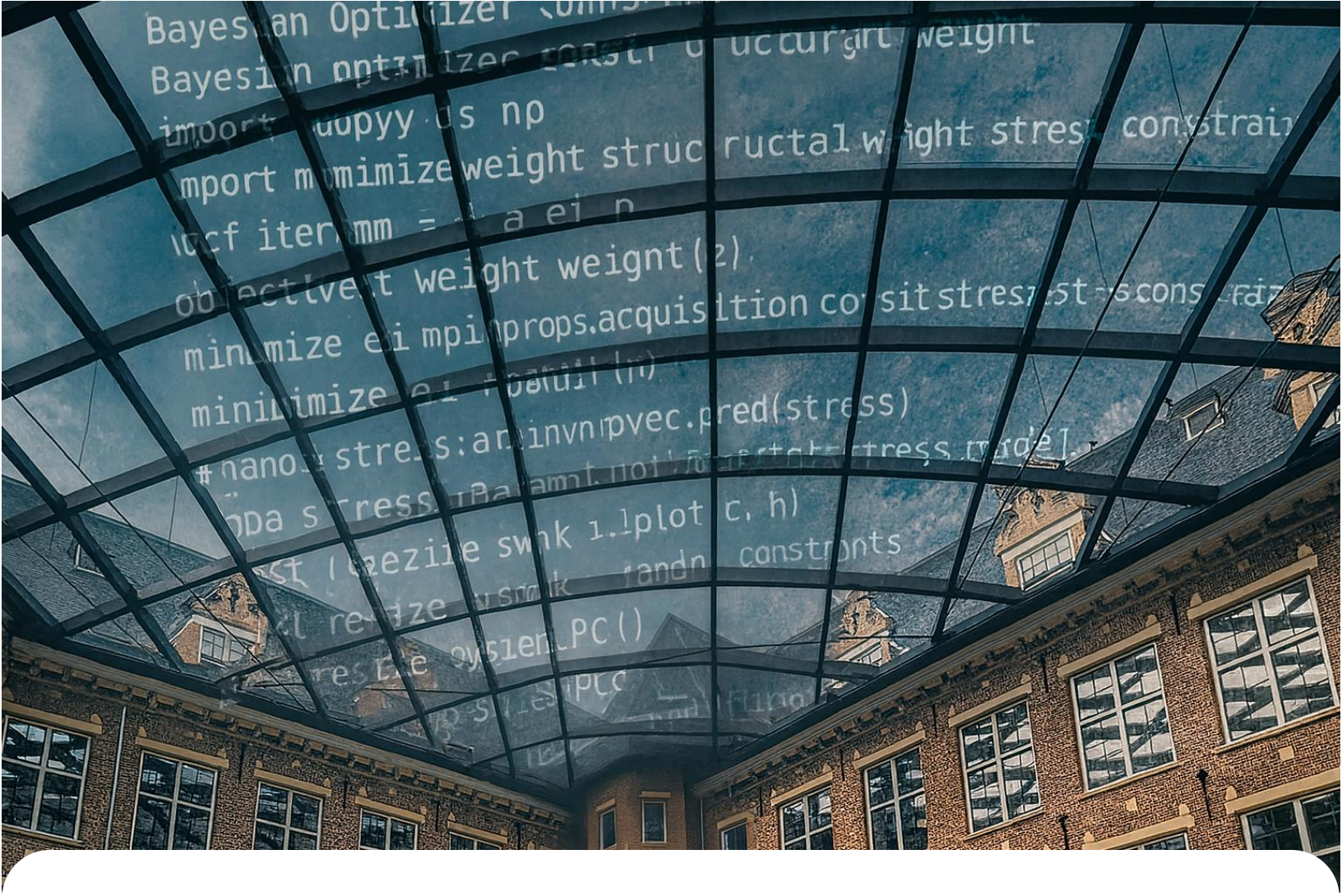


Bayesian optimization for weight minimization of gridshells in a complex design space

Prepared by:

Georgi D. Nikolov

Graduation Committee:

Dr.ir. F.P. (Frans) van der Meer	TU Delft
Dr. I. (Iuri) B.C.M. Rocha	TU Delft
Dr. F. (Florentia) Kavoura	TU Delft

Company Supervisor:

Ir. T.H. (Thijs) Schuiling Octatube Space Structures B.V.

(Page intentionally left blank)

Bayesian optimization for weight minimization of gridshells in a complex design space

by

Georgi D. Nikolov

In partial fulfilment of the requirements for the degree of

Master of Science in Civil Engineering

Track: Structural Engineering

At the Delft University of Technology,

to be defended publicly on Friday September 19th, 2025, at 15:30.

A digital version is available in the TU Delft repository.

(Page intentionally left blank)

Contact Information

Author

Name: Georgi D. Nikolov

Student number: 5857023

Faculty: Civil Engineering and Geosciences

Track: Structural Engineering

Chair & Main Supervisor

Name: Dr.ir. F.P. (Frans) van der Meer

Organization: Delft University of Technology

Faculty: Civil Engineering and Geosciences

Section: Applied Mechanics

Committee Member

Name: Dr. I. (Iuri) B.C.M. Rocha

Organization: Delft University of Technology

Faculty: Civil Engineering and Geosciences

Section: Applied Mechanics

Second Committee Member

Name: Dr. F. (Florentia) Kavoura

Organization: Delft University of Technology

Faculty: Civil Engineering and Geosciences

Section: Steel and Composite Structures

Company Supervisor

Name: Ir. T.H. (Thijs) Schuiling

Organization: Octatube Space Structures B.V.

(Page intentionally left blank)

Preface

This thesis is the result of my research at the intersection of machine learning, optimization algorithms, and structural engineering. It is the culmination of my period as a Structural Engineering student at the Delft University of Technology, Faculty of Civil Engineering and Geosciences. Besides being the result of a 6 month journey filled with challenges and personal growth, it is a reflection of my fascination with technology and how it can be applied to solve both scientific and practical problems.

The initial spark for this thesis came from a group project I undertook with Jos te Hennepe, Yu-Cih Lin, and Stefan Pol during the course Data Science and Artificial Intelligence for Engineers at TU Delft which focused on a 37-bar truss optimization under natural frequency constraints. That project showed me the potential of combining data-driven methods, particularly Bayesian optimization, with engineering design, and it served as the inspiration for the research direction I pursued in this thesis.

My enthusiasm to explore the uncharted waters of the application of Bayesian optimization for more complex geometries and structural constraints resonated with the structural engineers at Octatube. Having previously completed an internship on a completely different topic regarding the effect of non-rigid spacers on load-sharing in insulating glass units, showed our mutual curiosity about the potential of the research topic.

I am deeply grateful to my university supervisors Frans van der Meer and Iuri B.C.M. Rocha for their guidance, insightful remarks, and steady encouragement throughout this journey. Their feedback was essential in developing the optimization framework and challenged me to be precise in my assumptions and rigorous in my conclusions. I also wish to thank Florentia Kavoura for her kind support during the final stages of this period whose expertise in the steel and composite materials contributed to my understanding of these complex topics.

A special word of gratitude goes to my company supervisor, Thijs Schuiling, whose practical perspective and insightful questions were invaluable for translating this abstract topic for a broader audience. Our weekly progress meetings were instrumental for the completion of this thesis right from the very beginning of the project and his feedback always challenged me to zoom out and see the bigger picture when I was stuck in the details.

I am also grateful to my colleagues in the structural engineering team at Octatube for their interest and encouragement during the coffee breaks and for creating such a collaborative and stimulating working environment.

On a personal note, I want to thank my family and friends for their continuous encouragement during this journey. Their patience and support have been indispensable, especially during the most challenging phases of my research.

Georgi Nikolov
The Hague, September 2025

Abstract

This thesis investigates the application of Bayesian Optimization (BO) for the weight minimization of macrostructural systems, focusing on a cantilever beam, a truss, and two gridshells as case studies. Traditional structural optimization methods often struggle with high-dimensional, non-convex design spaces while being constrained by expensive function evaluations due to repeated finite element analyses. BO addresses these challenges through surrogate modelling with Gaussian Processes (GPs) and probabilistic acquisition functions that balance exploration and exploitation. The study integrates BO with the finite element package RFEM6, enabling automated optimization workflows subject to Eurocode-based strength, stiffness, and stability constraints.

Four case studies of increasing complexity are implemented:

(i) 1D cantilever beam with a varying size variable
(ii) 2D cantilever truss with varying shape and size variables
(iii) 3D 4×4 gridshell with varying size variables
(iv) 3D 9×9 gridshell with varying size variables

The achieved results via the constrained BO algorithm for the 4x4 gridshell show a *1.67*× lighter structure than the reference design and *2.54* × lighter structure than the reference design for the 9x9 gridshell confirming that BO can converge towards feasible and lightweight structural designs. The efficiency of the algorithm is further demonstrated by its ability to converge *18 times* faster to a design that is only 0.5% heavier than the reference design for the cantilever truss case study.

Furthermore, the 1D case demonstrated robustness and integration feasibility between the Python implementation and the RFEM6 software. The 2D truss highlighted the benefits of embedding structural knowledge in the sampling strategy and showed that using multiple GPs per member improved reliability compared to aggregated models. For high-dimensional 3D gridshells, the optimizer maintained feasibility but faced some scalability issues. Principal Component Analysis (PCA) is introduced to mitigate the "curse of dimensionality" by exploiting the underlying pattern of the cross-sections that depends on the internal forces while reducing computational cost. However, it is found that excessive dimensionality reduction degrade the solution quality, indicating a trade-off between efficiency and accuracy. Therefore, it has to be applied carefully to retain enough structural variance.

Two other key findings can be emphasized. First, increasing the number of surrogate models to approximate the structural constraints for each element in the system improves accuracy but increases the computational cost. Second, informative initialization and structural-domain knowledge can enhance the convergence rate.

The thesis concludes that Bayesian optimisation either with or without applying the PCA, is a viable and sample-efficient strategy for structural weight minimization under realistic structural constraints, capable of being integrated with industry-standard FEM software. Future work should explore the scalability of the BO framework in higher dimensional feature space, the implementation of multi-objective BO and apply it to case studies with broader structural typologies such as moment frames composed of different cross-section types.

Finally, a basic version of an interactive tool is developed that integrates the knowledge discussed in this thesis and that can be used by the structural engineers to explore various design options in the early design phase of a project.

Keywords: Bayesian Optimization, Surrogate Modelling, Structural Optimization, Gridshells, Finite Element Analysis (FEM), Principal Component Analysis (PCA)

Contents

List of Figures

List of Tables

Terminology

Bayesian optimization (BO) - sequential optimization technique for finding the minimum or maximum of expensive, black-box functions. It builds a probabilistic model (typically a Gaussian Process) of the objective function and uses this model to select promising points to evaluate, balancing exploration and exploitation to efficiently converge to the optimum.

Gaussian process (GP) - a Gaussian probability density over functions. In practice, this manifests as any arbitrary number of points in the function being jointly Gaussian.

Kernel function - defines the covariance (similarity) between any two input points (features) in a Gaussian process. The notation is $k(x,x')$, where $x,x' \in \mathbb{R}^D$ and D is the number of input dimensions.

Acquisition function – a strategy used in Bayesian optimization to determine the next point to evaluate by balancing exploration and exploitation of the design space based on the surrogate model's predictions. Examples include *Expected Improvement (EI)* and *Upper Confidence Bound (UCB)*.

Surrogate model - an approximate model used to mimic the behaviour of an expensive or complex function, enabling efficient optimization by providing predictions for the behaviour of this function.

Hyperparameter - a configuration variable whose value is set before the learning process begins and governs the behaviour or capacity of a machine learning algorithm, such as kernel length-scale.

Size optimization – the process of selecting the cross-sections of structural members to minimize an objective (e.g., total structural weight) while satisfying performance and safety constraints.

Shape optimization - The process of modifying the geometry of a structure to minimize an objective (e.g. total structural weight) while satisfying design constraints.

O(N) – Big O notation is a mathematical notation used to describe the upper bound of an algorithm's growth rate, expressing how the runtime or computational complexity scales with the size of the input.

API - (Application Programming Interface) is a defined set of rules and protocols that allows different software systems or components to communicate and exchange data with each other.

1. Introduction

1.1. Research Problem/Background

Nowadays, it is crucial for structural engineers and builders to implement strategies to minimize the use of new steel components in their projects. Steel ranks as the second most frequently used material in building construction after concrete (Chen et al., 2022). According to a 2024 report from the World Steel Association the building and infrastructure sectors accounted for 52% of global steel use in 2023 (WSA, 2024). Additionally, it is reported that on average 1.92 tonnes of CO2 are emitted for every tonne of steel produced which amounts to 7-9% of global and 5% of EU CO2 emissions (Somers J., 2021). Furthermore, it has been estimated by Allwood et al. (2010) that the demand for buildings materials is expected to double by 2050. Meanwhile, the IPCC report (Fischer et al., 2007) recommends that global carbon emissions need to be cut by at least 50% by 2050 in order to limit global warming to 1.5 °C compared to pre-industrial levels as stipulated in the Paris Agreement in 2015 (UNFCCC, 2015). Specifically, the European Union has set a target to reduce emissions by 55% by 2030 (Somers J., 2021). Similarly, the Netherlands has set the ambitious goal to be a net-zero country by 2050 in the Dutch Climate Act (Klimaatwet, 2019).

To address these issues, one strategy is to minimize the use of raw steel material early in the design stages of a project. In this thesis, the focus is on steel gridshell structures as they are considered as one of the most efficient structures. Driven by the demand for environmentally sustainable solutions, their relevance has grown, owing to their efficient use of materials (Dyvik et al., 2021) and capacity to span large distances (Grande et al., 2017a). They own this capacity to their shape because membrane forces (axial compression and tension) are mainly active, while bending moments are ideally zero (Adriaenssens et al., 2014). This enables the use of smaller cross-sections relative to other structures that fulfil the same purpose such as frames.

The optimization process of such structures is subject to several mechanical and fabrication requirements such as physical, environmental and economic constraints which often lead to single or multi-objective optimizations algorithms (Grande et al., 2017a). These techniques are often time-consuming and have high computational costs. Despite being powerful tools, optimization routines can pose significant challenges for designers. Small errors can only be identified and corrected after the process is finished, which can take a significant amount of time (e.g. hours or days).

1.2. Research Context

This thesis aims to use Bayesian optimization for optimizing the design of gridshells by efficiently navigating the complex design space to identify optimal configurations with minimal computational effort. The potential of this approach for high-dimensional non-convex multi-objective optimization problems has been suggested by Shende et al., (2021) and Mathern et al. (2020), even though they applied the procedure for different problems in the structural engineering field. Gridshell design involves numerous parameters, such as geometry, boundary conditions, load conditions and connection/joint stiffness design, which can create a vast and intricate design landscape that traditional optimization methods may struggle to explore effectively. Bayesian

optimization, however, excels in such scenarios by using a probabilistic model to predict and select promising design candidates that balance exploration and exploitation. By doing so, it requires fewer evaluations of the design space, which is particularly advantageous given the high computational cost of structural simulations via finite element modelling. Additionally, Bayesian optimization's ability to incorporate prior knowledge can further streamline the search for optimal gridshell designs. This characteristic is especially beneficial when dealing with constraints and complex structural performance criteria, enabling designers to efficiently converge on cost-efficient solutions that optimize material use and structural performance.

A basic version of a design tool can be developed based on this research that supports decision-making in the early stages of projects, particularly for complex structures such as gridshells. This industrial need forms the background of the present thesis and motivates the research questions on how Bayesian optimization can be applied to support feasible and efficient steel construction. Such a tool would bridge the gap between computational optimization, structural efficiency, and sustainability, contributing to a more responsible approach to steel construction.

1.3. Research Objectives

This research project aims to determine to what extent the Bayesian optimization framework can be applied to efficiently optimize the layout and cross-sections of truss and gridshell structures, made from steel elements, focusing on the minimization of the structural weight. The outcome must be a structurally sound and feasible structure. The developed workflow must have the possibility of being implemented into a structural engineer's practice by integrating it with industry-standard tools like RFEM6. Throughout the thesis "optimization framework" and "optimization algorithm" are used interchangeably.

The objectives of this research are the following:

1. Investigate how to streamline the data exchange between RFEM6 and a Python implementation of the Bayesian optimization framework. For this purpose, problems with lower complexity are used such as a cantilever beam and a cantilever truss case discussed in Chapter 5.
2. Implement the structural constraints in order to optimize for the shape, size of the 2D cantilever truss case, 4x4 and 9x9 gridshells and how are possible scalability issues addressed.
3. Investigate ways of improving the performance of the framework given the dimensions of the different test cases.

1.4. Research Questions

The main research question that this thesis aims to answer is the following:

"To what extent can Bayesian optimization be applied to efficiently optimize the shape and cross-sections of structures in terms of minimizing structural weight, while ensuring structural integrity and integration with industry-standard tools like RFEM6?"

To answer this research question effectively a set of sub-questions is defined. They are split into two categories that relate to the development of the Bayesian optimization framework and to the understanding of the gridshell as a structure and how to model it in RFEM6.

RQ1: How can the Bayesian optimization be used in the weight optimization of macrostructures?

1.1. What are the current state-of-the-art shape and size optimization algorithms for structural design given a set of spatial & structural constraints?
1.2. How can the Bayesian optimization framework be used to optimize the shape and size of structures (e.g. truss and gridshell)?
    1.2.1. What restrictions in the design space should be made to facilitate the convergence of the algorithm to a solution?
1.3. How can the key structural performance metrics be included effectively in the Bayesian framework?
    1.3.1. What surrogate models should be used to approximate the relationship between design variables (e.g. node coordinates & cross-sectional dimensions) and objectives (e.g. minimization of the structural weight)?
1.4. What will be an appropriate acquisition function that, in combination with the surrogate model, will allow for a balanced exploration versus exploitation approach to avoid local optimal solutions?
1.5. What should be the size of the initial training set?
1.6. Which hyperparameters are the most influential in the optimization algorithm based on a sensitivity analysis and how often do they need to be updated to obtain the optimal solution without significantly increasing the computational cost of the algorithm?
1.7. What should the stopping criteria of the Bayesian optimization framework be to arrive at useful and computationally efficient results?
1.8. What techniques can be used to reduce the dimensionality of the optimization problem in case the computational cost is too high?
1.9. How many finite element evaluations are needed and are the obtained optimal results consistent?
1.10. How do the results of the Bayesian optimization framework compare to other optimization algorithms from literature in terms of results, computational time and model evaluations?

RQ2: How can a gridshell be modelled in RFEM6 and what is the structural behaviour?

2.1. What is the structural behaviour of a gridshell under uniformly distributed load?
2.2. How can the geometry of the gridshell be generated to obtain the desired structural behaviour?

2.3. How can the structural model be created in a finite element software such as RFEM6?
    2.3.1. What is the governing load combination that should be considered in the analysis?
    2.3.2. What are the boundary conditions of the model?
    2.3.1. What cross-section types are suitable (e.g. standardized vs parametric, open vs closed)?
2.4. What are the key structural performance metrics that must be calculated to evaluate the structural feasibility in terms of strength, stiffness and stability and guide the optimization process?

1.5. Research Scope

To make the aim and objectives (outlined in the previous section) of this thesis feasible the research scope is defined within the following boundary conditions:

- Symmetric load cases and load combinations.
- Symmetric and asymmetric boundary conditions.
- The structural requirements are element based (e.g. stress, buckling).
- The detailed design of the connections is not considered.
- Connections are modelled as pinned.
- Geometrically linear static analysis is considered as the main analysis of obtaining the internal forces of members and resulting stresses.
- Global stability analysis is excluded from the BO framework, due to the nonlinear second-order (P- Δ) calculations that are necessary to be performed.

1.6. Research Methods & Software

For this research, a mixed-method research strategy is employed such as FEM modelling and expert elicitation. Integration of computational modelling and data analysis is used to explore the optimization of a truss and gridshell structures. The primary method involves using a Bayesian optimization framework to analyse and enhance design parameters, providing an efficient exploration and exploitation of the design space. The FEM software package RFEM6 is utilized for the structural analysis. It is integrated with Python as the base programming language to facilitate the development of the optimization algorithm. This combination allows for precise modelling and evaluation of structural performance under various scenarios with subsequent data analysis of the results.

1.7. Research Gap and Report Structure

As discussed by Shende et al. (2021) the potential of Bayesian optimization for higher dimensional structural design problems remains largely unexplored.

In Chapter 2 the literature has been analysed with the overview that mainly metaheuristic algorithms are used for those types of problems. In some cases, they have high computational cost (e.g. genetic algorithms), and the results in other cases (e.g. harmony search) are not as optimal as they have been found to be by the more accurate but even more time-consuming deterministic optimization algorithms.

This thesis aims to address the research gap concerning the application of probabilistic Bayesian optimization to higher-dimensional, non-convex shape and size optimization of structures. The study focuses on gridshell geometries, with the objective of minimizing structural weight while ensuring compliance with strength and stability constraints.

The report structure linking the research questions to the methods used to answer them is shown in Figure 1.

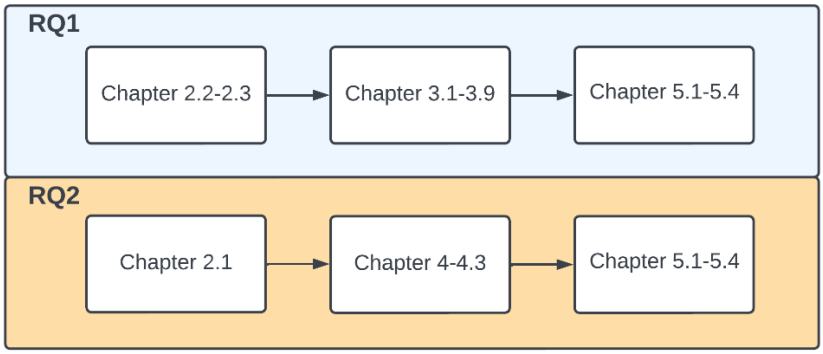


Figure 1: Structure of the thesis.

2. Literature review

In this chapter a literature study is presented covering the important topics relevant to the thesis questions and objectives outlined in the previous chapter. First, the mechanical behaviour and design considerations such as form-finding of gridshell structures is discussed to establish an understanding of how these structures work which will help in validating the RFEM6 models for the gridshell case studies discussed in Chapter 5. Next, to establish a reference frame, an overview of the most commonly used algorithms for shape and size optimization in the structural engineering field are presented. A detailed analysis of the pros and cons of each of the referenced papers outlined in Table 14 in Appendix A. Finally, some known and recent papers are discussed on the topic of Bayesian optimization and its successful application in different fields of engineering.

2.1. Structural principles of gridshell structures

Shells are thin single or double curved structures (Williams, 2014). They have been used for centuries in architecture and engineering, providing efficient, lightweight solutions for spanning large spaces. One historical example is the Roman Pantheon (Figure 2), with its vast concrete dome spanning almost 44m. In the 20th century, the development of reinforced concrete and advanced computational methods enabled the construction of even more complex shell structures, such as Pier Luigi Nervi's reinforced concrete dome Palazzetto dello Sport (Figure 3) and Felix Candela's thin concrete shells such as the hypar shell L'Oceanogràfic in Valencia, Spain, (Figure 4). A good example for the efficiency of shell structures is the elliptic paraboloid built by Ove Arup as the new concrete roof for the Smithfield Poultry Market in London (Figure 5). The structure spans almost 70m while the thickness of the concrete is mainly 7cm in the middle part and is increased to 15cm towards the edges due to the bending moments (Ahm & Perry, 1965).

Figure 2: The Pantheon, Rome, Italy. Bult in 126 AD. (*Source: Ravisetti, 2023*).

Figure 3: Pier Luigi Nervi - Palazzetto dello Sport, Rome, Italy. Built in 1957. (*Source: Structurae*).

Figure 4:Felix Candela - L'Oceanogràfic, City of Arts and Sciences, Valencia, Spain. Built in 2003. (*Source:Lázaro. 2023*).

Figure 5: Ove Arup - Concrete shell roof for the Smithfield Poultry Market, London, UK. (*Source: Julian Harrap Architects*).

To achieve this slenderness the load transfer mechanism of shell structures is fundamentally different from that of beam or frame systems. Rather than resisting loads primarily through bending, shells rely on in-plane forces or "membrane action" to efficiently distribute loads (Borgart, 2024b*)*. This allows shells to achieve remarkable structural efficiency, as forces flow smoothly through their curved geometry toward supports. The shell behaviour is compared to the plate behaviour in Figure 6.

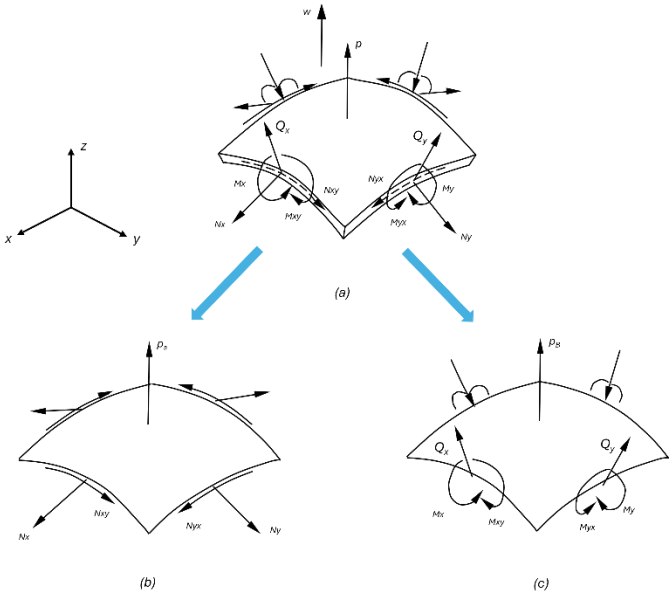


Figure 6: Shell behaviour. (a) Internal forces of a curved (shell) element; (b) Membrane action; (c) Plate behaviour. (*Source: Borgart, 2024b*).

A key characteristic to consider in the design of shell structures is their deformation. There are two types: extensional and inextensional deformation (Figure 7). In the former when a load is applied the middle part of the shell is stretching to accommodate the bending deformation. The load is mainly carried by membrane forces and the structure remains stiff. The case of inextensional deformation refers to the ability of shells to change shape without significant stretching or

compression of their surface. Here the load is carried by bending forces and there are large displacements. Therefore, thin shells have to be designed such that they do not undergo inextensional deformation when a certain load is applied (e.g. snow) because then the bending stresses are very large. However, when there is imposed displacement (e.g. foundation settlement) it is good to allow inextensional deformation to occur since it gives lower stresses in the shell (P.C.J. Hoogenboom, 2024).

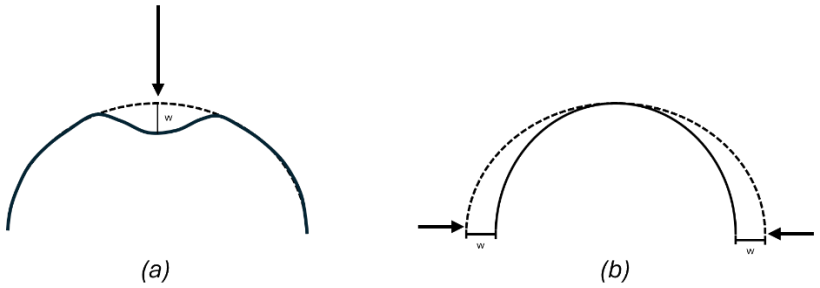


Figure 7: Shell deformations. (a) extensional deformation; (b) inextensional deformation.

The design of shell structures often involves form-finding, a process that determines the optimal shape of a structure based on a given load, force flow or stress field and boundary conditions. In traditional construction, form-finding is achieved through physical models using hanging chains or soap films, as seen in the work of Antoni Gaudí (Figure 8) and Frei Otto (Figure 9).Today, computational tools such as Grasshopper's Kangaroo physics engine (Castro & Song, 2024), which uses the dynamic relaxation method, enable engineers to simulate and refine shell geometries digitally (Figure 10). This method is used to form find the gridshell geometries discussed in subchapters 5.3 & 5.4.

Figure 8: Antoni Gaudi - Hanging Chain Model for La Sagrada Familia, Barcelona, Spain. (*Source: Zexin & Mei, 2017*).

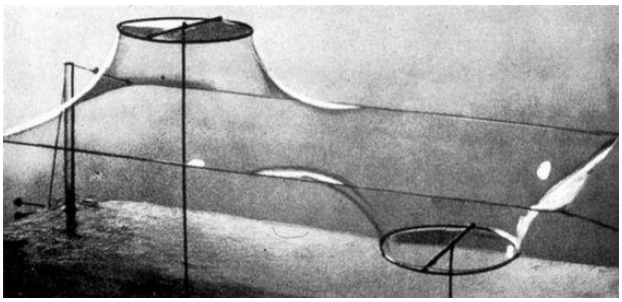


Figure 9: Frei Otto - Soap film model for the Munich Olympic Stadium, Munich, Germany. (*Source: Zexin & Mei, 2017*).

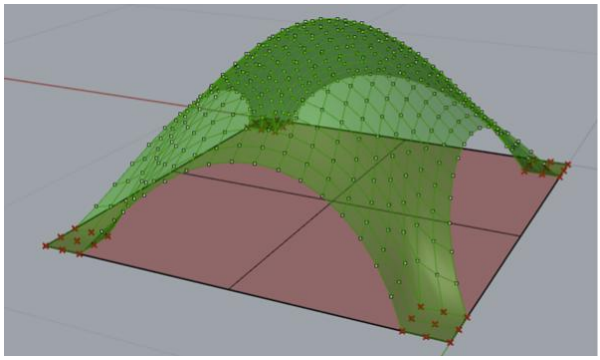


Figure 10: Concrete shell form found using Kangaroo/Grasshopper. (*Source: Tamplin, R & Iuorio, Ornella., 2018*).

Techniques for designing 2D curves and 3D surfaces can be categorized as either physics-based or mathematics-based. A summary is provided in Figure 11. By using these methods, designers can create shell structures that are not only aesthetically pleasing but also structurally optimized to withstand applied loads efficiently.

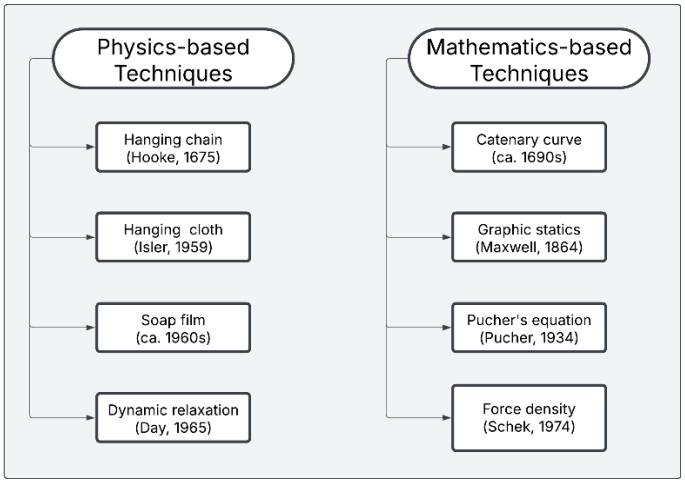


Figure 11: Classification of form-finding methods. (*Source: Chiang, 2022*).

This thesis focuses on gridshells, which are a subset of shells. Both share a fundamental structural principle: they rely on their curved geometry to efficiently transfer loads primarily through membrane action, minimizing bending stresses. However, while shells are continuous, thin-surfaced structures made from mainly concrete, gridshells are composed of a network of interconnected linear elements (often timber, steel, or composite materials) that form a flexible lattice capable of assuming a shell-like shape. They can be single layered or double layered (Figure 12). This distinction allows gridshells to be lighter and more adaptable, as they can be assembled flat and then deformed into their final shape, a technique seen in projects like the Mannheim Multihalle by Frei Otto (Figure 13).

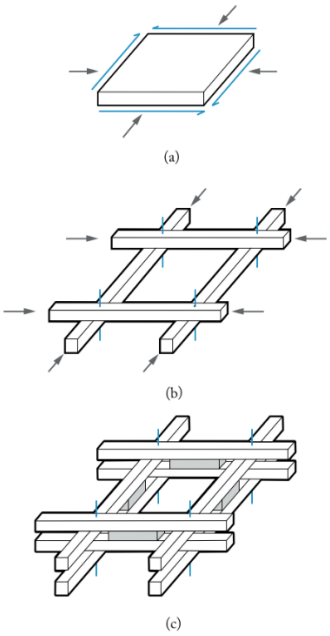


Figure 12: Structural scheme for (a) continuous shell; (b) single-layered gridshell; (c) double-layered gridshell. (*Source: Adriaenssens et al., 2014*)

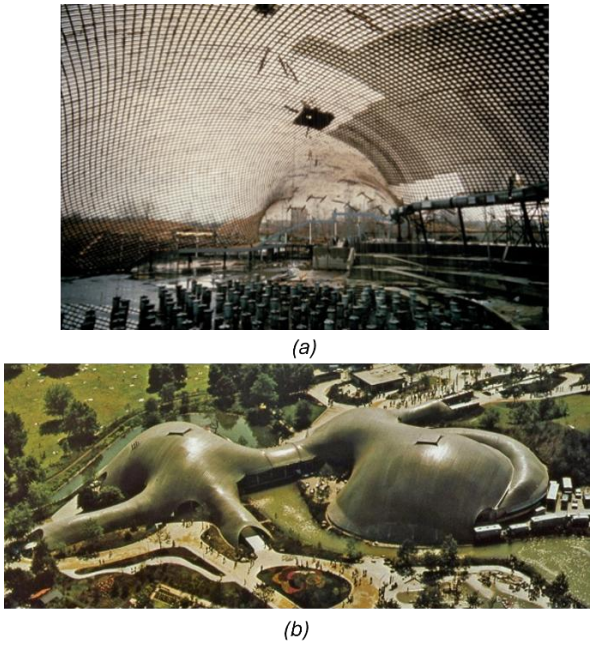


Figure 13:Frei Otto - Multihalle, Herzogriedenpark, Mannheim, Germany. (a) inside photo; (b) outside photo. (*Source: Adriaenssens et al., 2014*)

One additional point of attention relating to gridshells that is not necessarily required for continuous shells is form stability. It refers to their ability to maintain structural integrity and resist excessive deformations under applied loads, including self-weight, wind, and snow. Instability can happen when considering the structure as a whole (Figure 14) or in-plane (Figure 15). This phenomenon is inherently linked to factors such as curvature, nodal connections, boundary conditions, and the load-bearing capacity of individual elements. By not making a gridshell form-stable out-of-plane bending moments and in-plane bending moments can be developed depending on the load case. Achieving form stability requires a balance between geometry, joint and connection design, material properties, and structural optimization to create efficient, durable, and aesthetically compelling structures.

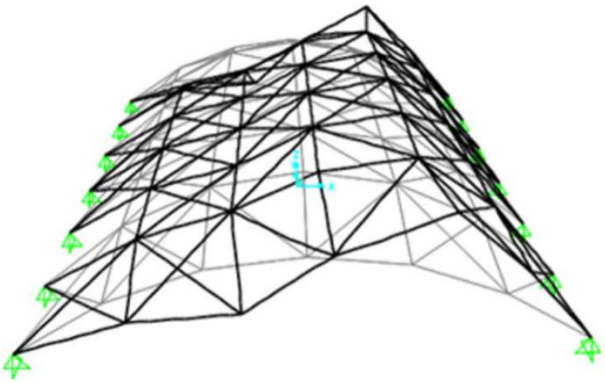


Figure 14: Form-stability of the whole structure. (Source: Grande et al., 2017b).

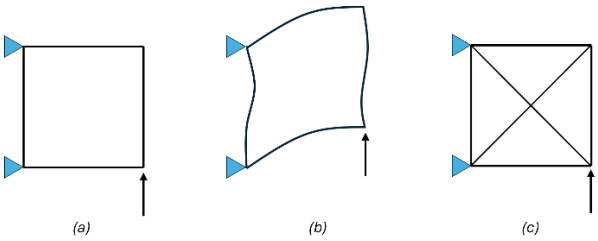


Figure 15: Form-stability in-plane. (a) Unstable configuration; (b) deformed state; (c) stable configuration.

Several remarkable gridshell structures showcase the versatility and efficiency of this construction method. The British Museum gridshell, designed by Foster + Partners with structural engineering by Buro Happold, was completed in 2000 and covers the Great Court with a striking steel and glass gridshell (Figure 16). Its flowing, doubly curved form maximizes natural light while creating a seamless architectural connection between historic and modern elements. Another outstanding example is the courtyard roof of the Museum of Hamburg History, built by Jörg Schlaich and completed in 2020. This lightweight steel-glass gridshell provides a transparent yet protective canopy over the museum's courtyard, blending contemporary engineering with historic preservation. The Dutch maritime museum gridshell, completed in 2011, features a spectacular glass and steel lattice structure that covers the museum's central courtyard. Inspired by 17th century nautical maps, its geometric pattern mimics compass lines, adding both structural efficiency and symbolic meaning to the design (Sigrid Adriaenssens. 2019). These examples highlight how gridshells can be adapted for different boundary conditions, creating visually striking, structurally efficient, and sustainable architectural solutions.

Figure 16: Queen Elizabeth II Great Court, The British Museum, London, UK. (*Source: Buro Happold, 2019*)

Figure 17: Courtyard Roof of the Museum of Hamburg (*Source: Ermias Y., 2013*)

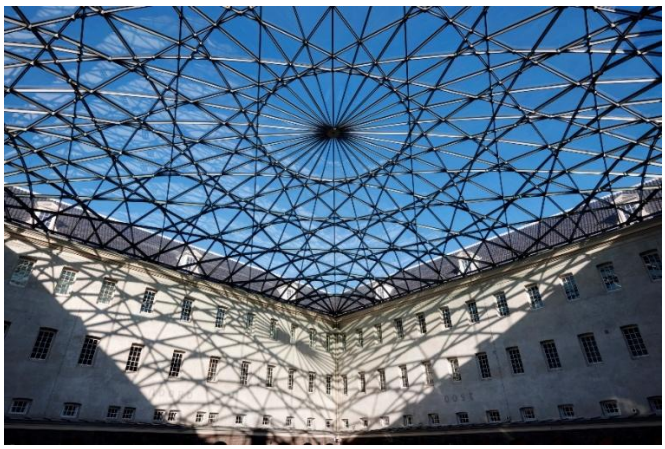


Figure 18: Glass Roof Dutch Maritime Museum. (*Source: Ney & Partners 2011*).

In recent years, the research done on gridshells has been steadily increasing from only 10 publications in 2011 to 327 in total in 2021 (Dyvik et al., 2021) as shown in Figure 19a. The right bar chart categorizes research contributions by discipline, showing that Structural Engineering dominates the field with 278 publications, far surpassing other areas. This indicates that gridshell research is primarily concerned with load-bearing capacity, stability, and optimization rather than aesthetic or material innovations.

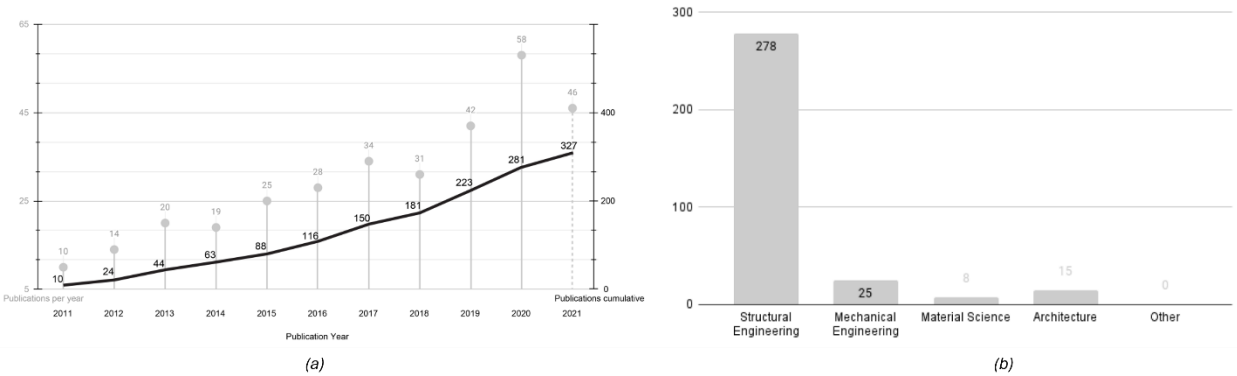


Figure 19: (a) gridshell publications per year; (b) research contribution per year. (*Source: Dyvik et al., 2021*)

Structural optimization techniques can typically be classified in three main categories depending on the design variables considered: shape, size and topology optimization (Gythiel & Schevenels, 2022). *Shape optimization* in this thesis focuses on refining the form (z-coordinates of the nodes) of the gridshell and the in-plane position of the nodes on the obtained surface (x- and y- coordinates of the nodes) to enhance structural performance (e.g. minimal bending moments, deflections). *Size optimization* involves adjusting the cross-sectional dimensions of the individual members in the

grid to balance material use and structural strength while ensuring compliance with constraints such as buckling resistance and maximum deflection.

In the past different researchers have studied how to optimize the design of gridshells using various strategies and algorithms. For example, Wang et al. (2019) have developed a physically-based bubble-packing model and a geometry edge operation to achieve triangular grids for complex free-form surfaces. Grande et al. (2017a) have shown the potential for combining different optimization strategies such as form finding, sizing optimization and topologic optimization using genetic algorithms to obtain light structural solutions for gridshells. On the other hand, Gythiel & Schevenels (2022) have used gradient-based algorithm to optimize the size, shape and topology of a single-layer reticulated (i.e. grid) shell under a distributed load. Furthermore, Saka (2007) has analysed the optimum geometry design for geodesic domes which is a type of gridshell by employing harmony search algorithms. Finally, Richardson et al. (2013) developed a coupled form-finding and grid optimization approach to design efficient gridshell structures by integrating geometric form-finding with structural and performance-based optimization using a genetic algorithm.

In the next subchapter, an overview of the current state-of-the-art algorithms used in structural engineering field for optimization problems is provided. Also, a case is made that Bayesian optimization can serve as an efficient framework for optimizing gridshells by leveraging uncertainty modelling to guide exploration of the design space, reducing computational costs through informed sampling, and improving convergence toward optimal structural configurations.

2.2. Common optimization algorithms in structural engineering

In recent years, the integration of artificial intelligence (AI) and machine learning (ML) with structural engineering (SE) has gained significant momentum, as seen by the increasing number of publications in this area. Figure 20 illustrates a notable rise in the number of articles published annually between 2011 and 2020, covering various intersections between structural engineering and advanced computational techniques. This surge underscores the growing interest and advancements in applying AI tools like neural networks, decision trees, and genetic algorithms to enhance structural analysis and design.

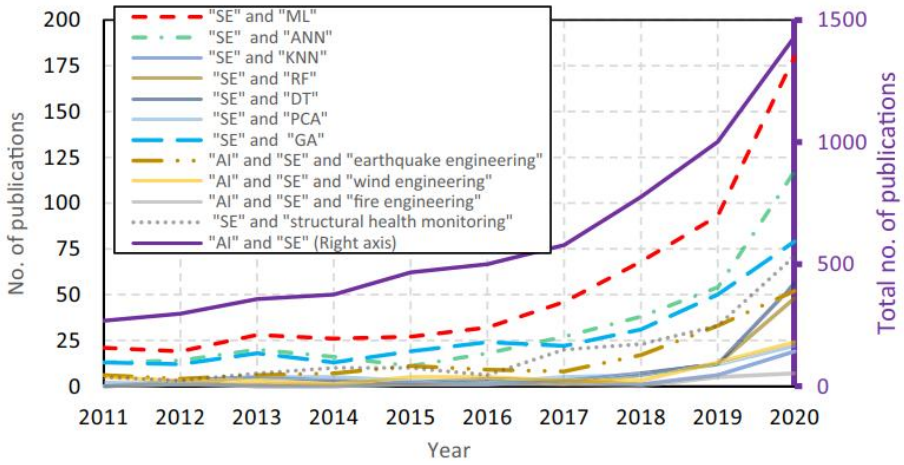


Figure 20: Number of publications from 2011 to 2020 (Tapeh & Naser, 2023).

Furthermore, Figure 21 highlights some of the journals frequently publishing these advancements, with titles such as "Computer-Aided Civil and Infrastructure Engineering" and "Construction and Building Materials" leading the distribution of this innovative research. These figures emphasize the pivotal role that AI and ML play in driving forward the future of structural engineering through interdisciplinary collaboration and exploration of potential areas of application for these techniques.

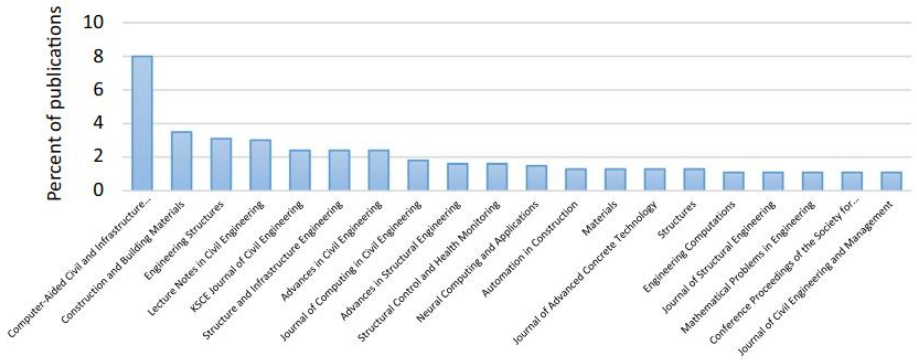


Figure 21: Journals publishing about AI and the structural engineering field (Tapeh & Naser, 2023).

In general, on a high level the hierarchical relationship between AI, machine learning (ML) and deep learning (DL) can be represented as shown in Figure 22 (Atul, 2025).

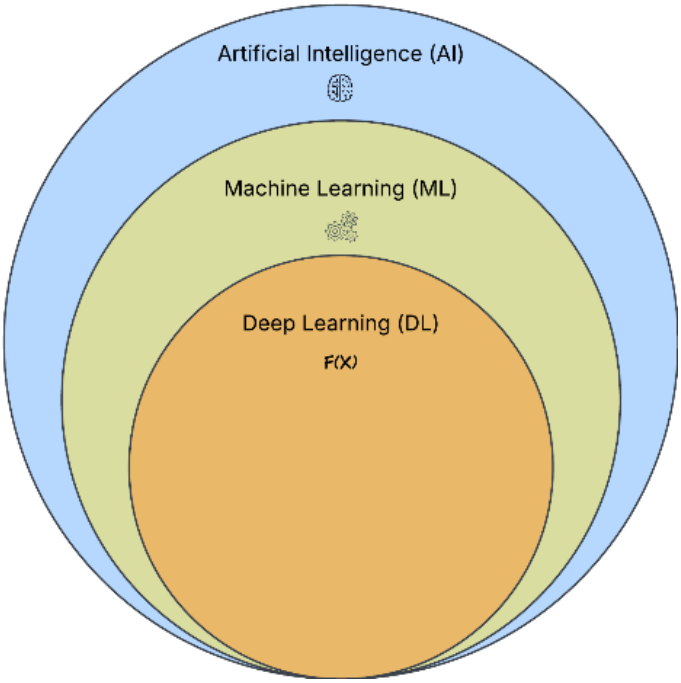


Figure 22: Hierarchical relationship between AI, ML and DL.

AI is the field of computer science focused on creating systems capable of performing tasks that typically require human intelligence. These tasks include reasoning, learning, problem-solving, language understanding, and decision-making.

ML is a subset of artificial intelligence that involves training algorithms to learn patterns from data and make decisions or predictions without being explicitly programmed for specific tasks. It enables systems to improve their performance as they are exposed to more data over time.

DL is a subset of machine learning that uses neural networks with multiple layers to model complex patterns in data. It is very good at tasks involving large amounts of structured or unstructured data, such as image and speech recognition.

Furthermore, these three areas can split into several subcategories. Figure 23 gives a non-exhaustive list of some examples. Bayesian regression has been highlighted by the red outline.

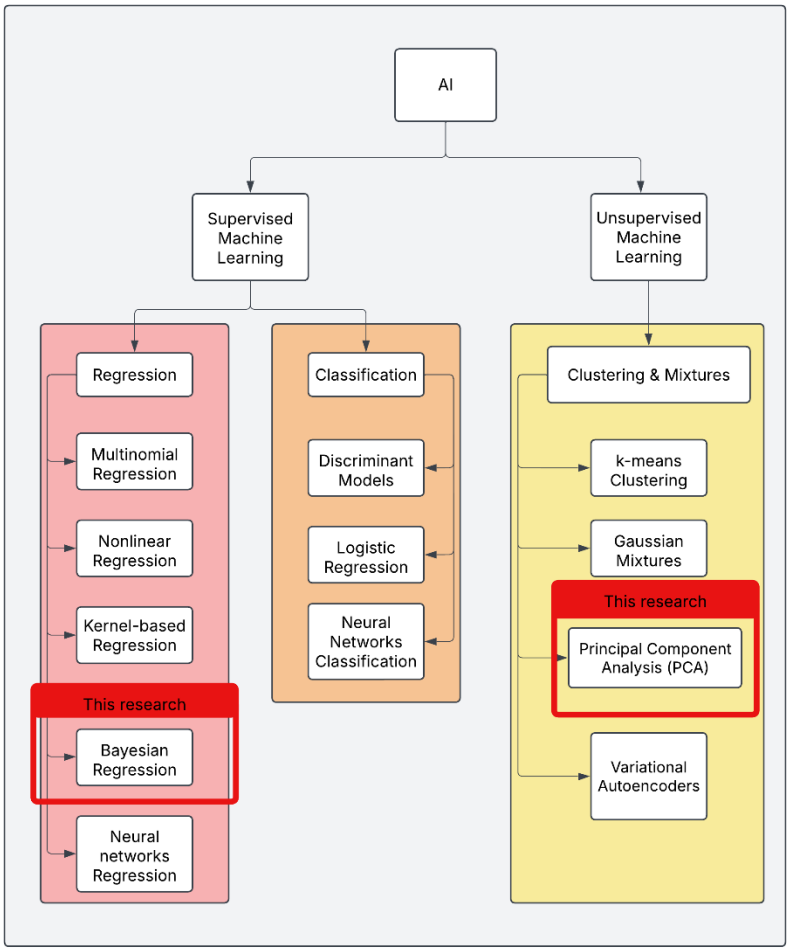


Figure 23: Subcategories of AI, ML and DL.

Supervised learning is used when both the outcome and the governing variables are known, such as a structural member and its cross section. This learning type can be categorized into regression (when predicting a quantity) or classification (when identifying a label or class). In contrast, unsupervised learning is applied when data is unlabelled, helping engineers discover the underlying structure, such as determining whether a signal from an onsite sensor indicates a structural crack. Deep learning methods are not used often in the field of structural engineering yet (Tapeh & Naser, 2023), although there are some recent examples of physics-informed NN being used for complex beam systems (Kapoor et al., 2023).

This thesis focuses on optimization of macro structures, namely trusses and gridshells. Optimization algorithms that aim to search for the best result for a set of variables under given constraints to achieve a given goal/objective are analysed. The following paragraphs include a

brief description of current state-of-the-art methods that are commonly used in practice and in literature.

An overview of most commonly used optimization algorithms is given in Figure 24.

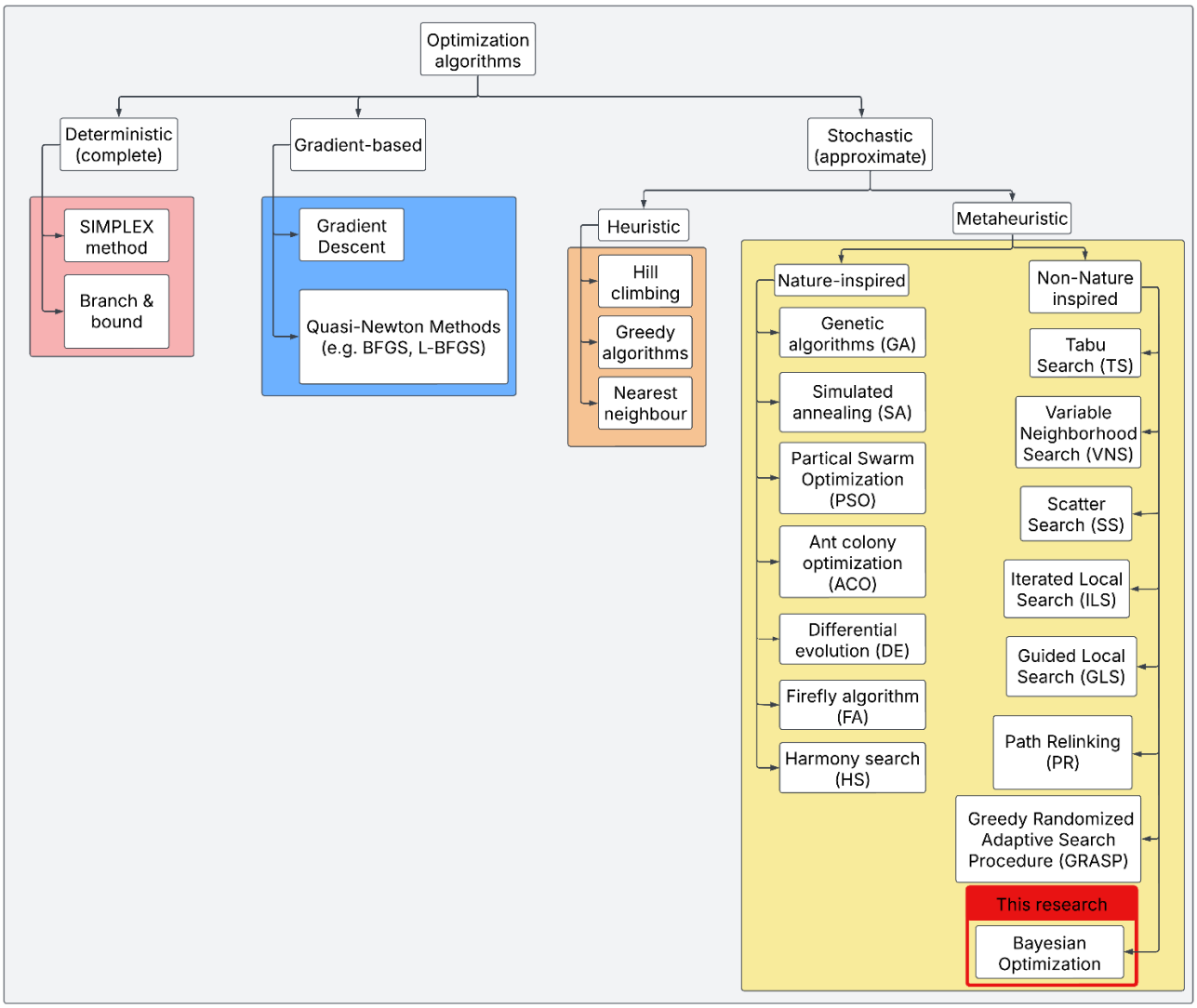


Figure 24: Overview of optimization algorithms. (Zavala et al., 2013).

The deterministic SIMPLEX method efficiently solves linear programming problems by traversing the vertices of the feasible region, while Branch & Bound systematically explores and prunes a search tree to find optimal solutions in integer and combinatorial optimization.

Gradient-based methods such as gradient descent are widely applied to continuous optimization problems with differentiable objective functions and are efficient and scalable for convex functions. However, they are prone to getting stuck in local maxima or minima when the function is multi-peak (non-convex).

On the other hand, stochastic algorithms, divided further into heuristic and metaheuristic methods, are well-suited for non-convex objective functions which could be known or unknown (black-box), where the solution landscape is more complex and varied. They are more likely to find the global optimum of a given function due to the random component inherent in their structure. However, this can be computationally expensive. The difference between convex and non-convex functions is illustrated in Figure 25.

Heuristic methods include approaches like Hill Climbing and Greedy Algorithms, which focus on finding approximately optimal solutions quickly in lower dimensional solution space utilizing the gradients of the functions if they are available.

Metaheuristic methods, such as Genetic Algorithms, Tabu search, and Particle Swarm Optimization, provide more robust solutions by exploring the solution space more thoroughly, making them ideal for challenging optimization scenarios such as constrained multi-objective non-convex problems in higher dimensional solution space (Blum & Roli, 2001). Another advantage is that they can be applied to a wide variety of problems without requiring domain specific information which makes them task independent. Nevertheless, this flexibility comes at the cost of fine tuning a lot of parameters to achieve good results which can increase the computational cost.

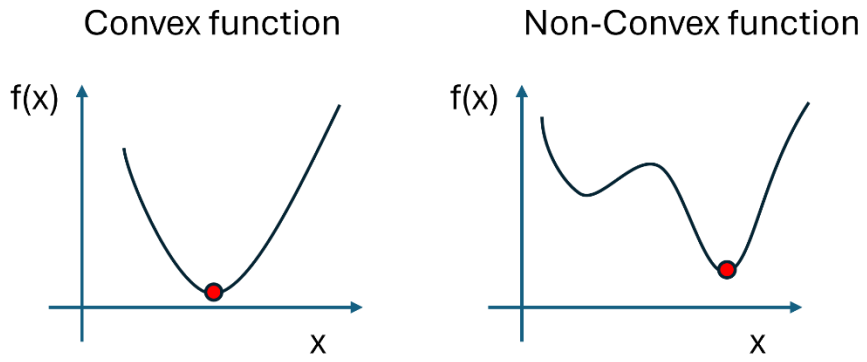


Figure 25: Convex and non-convex functions.

Furthermore, metaheuristic algorithms are often combined into hybrid approaches to achieve a good balance between the exploration and exploitation of the solution space of a given problem.

As discussed in a paper by Saka (2009) the nature-inspired optimization algorithms are beneficial in the optimization of combinatorial problems in terms of computational costs and near optimal results. A description of some algorithms is provided below. A summary of some of the papers that used them for structural optimization problems are provided in Table 14 in Appendix A.

Genetic Algorithms (GA): Simulate the Darwinian natural selection by evolving a population of candidate solutions through operations like selection, crossover, and mutation. They are well-suited for complex optimization problems with discrete variables and intricate constraints. Examples include shape and size optimization of trusses using parallel genetic algorithm (Wei et al., 2011), improved genetic algorithm (Tang et al., 2005), Cellular Genetic Algorithm (Rajasekaran, 2001).

Differential Evolution (DE): It works by iteratively improving candidate solutions using mechanisms similar to genetic algorithms, such as mutation and crossover, and is known for solving complex, real-valued optimization problems robustly and efficiently. Structural problems that have been optimized using this algorithm or a hybrid version of it include but are not limited to truss optimization with frequency constraints (Pham, 2016), improved DE (Ho-Huu et al., 2016), hybrid DE with symbiotic organisms search (Nguyen-Van et al., 2021), adaptive elitist DE -aeDE (Ho-Huu, Nguyen-Thoi, et al., 2016).

Harmony Search (HS): This algorithm simulates the creative process of musical harmony improvisation, adjusting solution vectors through memory consideration, randomization, and pitch adjustment to find optimal solutions. Studies that have used this technique (or variation of it) for optimization problems in structural engineering include improved harmony search (Degertekin, 2012), hybrid harmony search (Cheng et al., 2016), HS and firefly algorithm (Miguel & Miguel, 2012).

The following two algorithms are often classified under the agent-based models.

Particle Swarm Optimization (PSO): Inspired by the flocking behaviour of birds (agents) and schooling of fish, PSO optimizes a problem by iteratively updating candidate solutions (particles) based on their personal and collective best experiences. It is widely accepted for its simplicity and ability to quickly converge to good solutions. Some notable studies on layout optimization of trusses under various constraints have been published such as Cellular PSO (Gholizadeh, 2013), integrated particle swarm optimizer (Mortazavi & Toğan, 2016), particle swarm algorithm (Gomes, 2011), binary PSO (Luh & Lin, 2011), heuristic PSO (Li et al., 2007).

Ant Colony Optimization (ACO): ACO is inspired by the behaviour of ants (agents), using virtual pheromone trails to guide a population of solutions toward optimal paths. It is effective in solving routing and scheduling problems. Papers where variations of this algorithm have been used for structure optimization problems include space truss design (Camp & Bichon, 2004), streel frames including elemental warping effect (Aydoğdu & Saka, 2011).

2.3. Bayesian optimization

The algorithms outlined in the previous sub chapter are widely used in structural optimization problems as analysed by Hasançebi et al. (2009). However, this thesis focuses on the implementation of a Bayesian optimization framework for the shape and size optimization of two structural typologies - truss and gridshell, with the goal of minimizing the total weight of the structure given a set of spatial and structural constraints. As mentioned before, this approach has been used successfully in the fields of structural health monitoring (Huang et al., 2022), material microstructure optimization (Coelho et al., 2025), hyperparameter tuning (Snoek et al., 2012), and design of auxetic metamaterials (Tran et al., 2019). However, to the best of the author's knowledge it has not been used for the optimization of macrostructures. Its use in the optimization of global structural models is still underexplored with one recent example from literature on the design of origami folding structures that discusses the potential of the approach and recommends its use for higher dimensional design problems (Shende et al., 2021). It was concluded that Bayesian optimization requires fewer finite element solutions compared to traditional methods, making it a

promising choice for such non-convex optimization problems (presence of multiple local optima). Additionally, it consistently outperforms the other methods analysed in the paper, delivering previously undiscovered designs for the origami structure. More details about the paper are provided in Table 15 in Appendix A. Similar conclusions and recommendations about the potential of Bayesian optimization were reached by Mathern et al. (2020) who were able to achieve optimal results for the design of a concrete beam by leveraging the cheap evaluation cost of the objective function while modelling the constraints using Gaussian Processes due to their expensive computational evaluations.

Another very recent paper has shown significant improvements in performance and quality, especially in nonlinear settings, through various design scenarios employing Bayesian optimization. It reduces the number of required experiments and demonstrates its potential to enhance design methodologies in both material and structural engineering compared to established data-driven approaches. The paper has implemented Principal Component Analysis (PCA) for dimensionality reduction which further enhances efficiency and reduces computational burden (Coelho et al., 2025). The benefits are particularly notable in complex scenarios with geometric or material non-linearity, where it reduces the number of experiments needed to achieve target objectives. The authors have focused on both single-objective and multi-objective optimization and in both scenarios Bayesian optimization has been shown to perform well.

On the other hand, Moriconi et al., (2020) discusses potential limitation of the Bayesian optimization in higher dimensions ($D > 20$) due to the response surface learning and optimal input selection via the acquisition function being computationally heavy. However, the paper suggests that high dimensional data often can be represented via its lower intrinsic dimensionality representing a certain underlaying pattern in the data, which can be exploited by the optimization framework.

Bayesian Optimization (BO) can be classified under Stochastic Metaheuristic methods as shown in Figure 24 by the red highlight. It is stochastic because it relies on probabilistic models (usually Gaussian Processes) to guide the search for optimal solutions. It can be considered metaheuristic, as it is a high-level strategy designed to explore the search space efficiently, often outperforming traditional heuristics in black-box optimization problems as mentioned in the previous paragraphs. However, it is not explicitly nature-inspired but it can be classified under general probabilistic search methods.

3. Theory of Bayesian Optimization for macrostructures

This chapter introduces the theoretical foundations of Bayesian Optimization (BO) in the context of macrostructural design. The chapter begins with a general definition and background of the method, followed by a detailed discussion of its core components, including kernels, acquisition functions, and the exploration–exploitation trade-off. Practical considerations such as hyperparameter tuning and sampling strategies are then addressed, highlighting their role in ensuring robust and efficient optimization. Finally, the chapter presents Principal Component Analysis (PCA) as a dimensionality reduction technique that can enhance the performance of BO in high-dimensional structural design problems. Together, these sections provide the theoretical basis for the subsequent application of BO to structural case studies.

3.1. Definition/Background

Bayesian optimization is a probabilistic model-based approach for optimizing objective functions that are expensive to evaluate, or lack analytic expressions. Its foundation lies in the principles of Bayesian inference, which allow the incorporation of prior knowledge and the systematic update of beliefs about an uncertain quantity as new information is acquired. The core of this approach is Bayes' Theorem, which describes how to update the probability of a hypothesis as more evidence becomes available. Mathematically, Bayes' Theorem is expressed as:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

where $P(\theta|D)$ is the posterior probability of the parameters θ given observed data D, $P(D|\theta)$ is the likelihood of the data under the parameters, $P(\theta)$ is the prior probability of the parameters, and $P(D)$ is the marginal likelihood.

In the context of optimization, Bayesian optimization treats the unknown objective function as a random function and places a prior over it which is commonly a Gaussian Process (GP) completely defined by its mean and covariance:

$$y(\bar{x}) \approx GP(m(\bar{x}), k(\bar{x}, \bar{x}'))$$

where $m(\bar{x})$ is a mean function and $k(\bar{x}, \bar{x}')$ is a kernel function that defines the covariance between any two points $\bar{x}$ and $\bar{x}'$. In the next subchapter the discussion about kernel functions is extended.

As function evaluations are performed, these results constitute the data *D*, and the posterior distribution over the objective function is updated according to Bayes' Theorem. This probabilistic surrogate model captures both the current understanding of the function and the associated uncertainty in regions that have not yet been explored.

An example of a fictious function $f(x) = \sin(x) + 0.2\cos(3x)$ is presented in Figure 26 below.

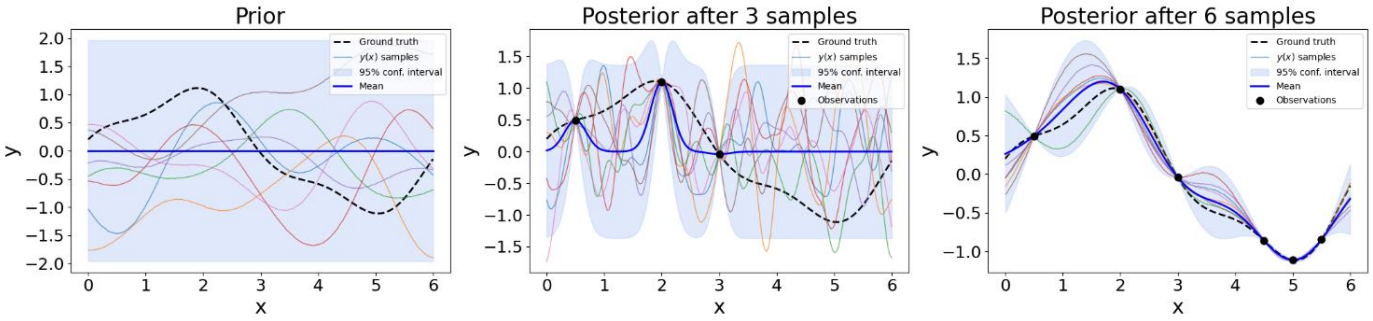


Figure 26: Example of a Gaussian Process used in a regression problem.

The optimization process leverages this surrogate model to make intelligent decisions about where to evaluate the objective function next. This is achieved through the use of an acquisition function (see subchapter 3.3), which balances the exploration of uncertain regions with the exploitation of areas likely to yield optimal values. In subchapter 3.4 how to achieve this balance is discussed. The acquisition function is computed using the posterior distribution, ensuring that each new evaluation provides the maximum expected improvement or utility given the current knowledge. As such, Bayesian optimization provides an efficient and principled framework for solving challenging black-box optimization problems by iteratively refining its probabilistic model and strategically selecting new sample points.

3.2. Kernels

Kernels play a central role in the theory and application of Gaussian processes and other machine learning algorithms such as Kernel Logistic Regression used for classification of data that is not linearly separable. A kernel function defines a measure of similarity or correlation between data points in a possibly high-dimensional feature space, enabling nonlinear modelling while maintaining computational efficiency. By selecting an appropriate kernel, prior knowledge about the underlying structure can be encoded such as the smoothness of the function to be learned, thus shaping the model's flexibility and generalization capabilities. The choice and design of kernel functions are therefore fundamental in capturing complex patterns and ensuring robust predictive performance in both regression and classification tasks.

In Figure 27 the Matérn kernel is shown. The effect of the smoothness determined by the hyperparameter v in the kernel is highlighted. Larger values of v correspond to smoother functions while smaller values produce functions with limited degree of differentiability, i.e. choppy functions. In the first row, with $v=\infty$, the Matérn kernel reduces to the squared exponential (RBF) kernel (shown in the first row in Figure 28), yielding samples that are infinitely differentiable and thus smooth. The covariance matrix is shown on the left and darker tones represent lower values and lighter tones higher values. In this case it displays gradual transitions, and the process samples exhibit minimal variation and high regularity meaning that the input data points/features are highly correlated. For the intermediate case ($v=1.5$), as shown in the second row, the functions become less smooth, i.e. only once differentiable with more pronounced fluctuations and moderately rougher samples, as reflected in both the covariance structure and the diversity of process samples. In the third row, with $v=0.5$, the kernel generates the least smooth samples. These functions are continuous but nowhere differentiable, resulting in highly erratic behaviour and a rapidly decaying covariance structure. Overall, the figure demonstrates that as $\boldsymbol{v}$ decreases, the Matérn kernel produces samples with increasing roughness, making it a flexible tool for modelling functions with varying degrees of smoothness in the Bayesian optimization used in chapter 5. More examples of different combinations of hyperparameters are shown in Figure 165 in Appendix D.

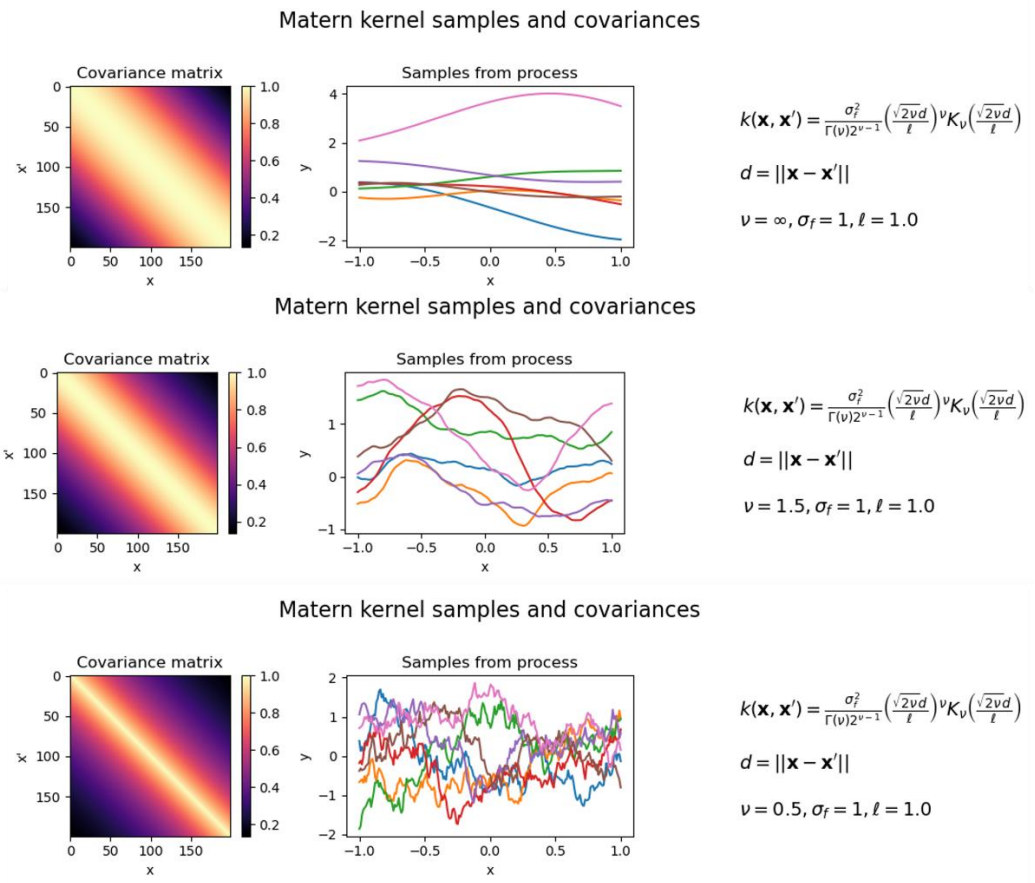


Figure 27: Matérn kernel and samples.

In Figure 28 the squared exponential (SE) kernel is shown. This is one of the most widely used covariance functions in Gaussian Process modelling due to its smoothness and flexibility. In literature it is equivalent to the radial basis function (RBF) kernel with infinite number of basis functions. In the definition of the SE kernel the d=||x−x′|| is the Euclidean distance between inputs points, ℓ is the length-scale parameter, and σ_f denotes the variance/amplitude of the samples. The length-scale parameter ℓ controls the smoothness of the resulting functions: *larger* values of ℓ correspond to broader correlations between points, resulting in smoother sample functions with gentle variation, as seen in the first row. As ℓ decreases (middle and bottom rows), the kernel function becomes increasingly localized, and the resulting samples display higher-frequency fluctuations and rougher behaviour, as the covariance decays more rapidly with distance. This property is reflected in both the structure of the covariance matrix and the increasing complexity of the drawn samples. Overall, the SE kernel is infinitely differentiable, making it suitable for modelling processes that are expected to be very smooth, with the length-scale parameter providing a direct means of tuning the modelled smoothness.

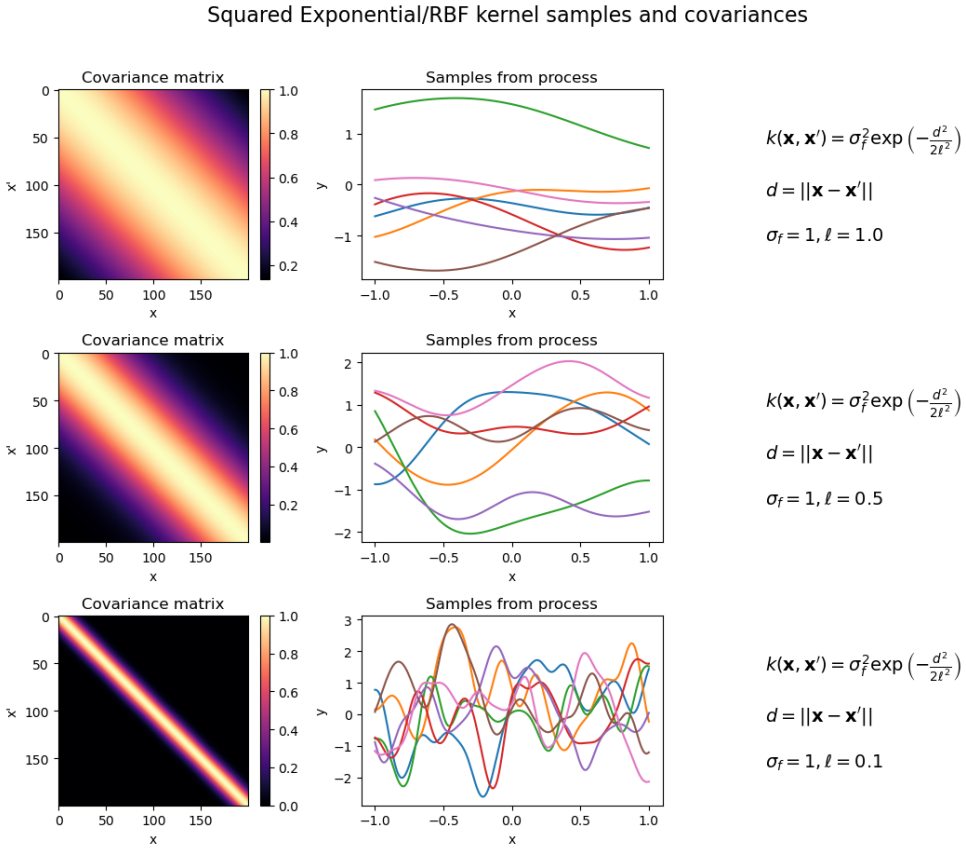


Figure 28: Squared Exponential/RBF kernel and samples.

Given the basic kernels above, they can serve as the building blocks of hybrid kernels such as the one shown Figure 29. Hybrid kernels can be a result of linear combination of different kernels as long as the covariance for any set of function values is *positive definite*. This property ensures that all variances are positive, all pairwise relationships are consistent, and, in probabilistic models,

guarantees that the associated multivariate distribution is well-defined and that the matrix is invertible.

Figure 29 presents the characteristics of a hybrid kernel constructed as a linear combination of a Matérn kernel and a squared exponential (RBF) kernel. This formulation allows the model to capture a richer class of functions by blending the distinct properties of each component. The Matérn kernel, parameterized by its smoothness ν and length-scale ℓ_1, provides control over function roughness, while the RBF kernel, governed by ℓ_2, ensures smooth and infinitely differentiable behaviour. In the first example, where the Matérn component is dominant due to the constant $C_1 = 1$, the resulting functions are relatively smooth but exhibit some moderate variability due to the contribution of the RBF term (ℓ_2 =0.3). In the second example, the process samples display a bit more complexity and more pronounced local fluctuations, as evidenced by both the covariance structure and the sample paths. The third example, where the Matérn kernel has both a higher weight (C_1=1.5) and a smaller length-scale (ℓ_1 = 0.2), produces functions that are even rougher and exhibit higher frequency variations. Overall, the hybrid kernel's flexibility enables it to model data exhibiting both global smoothness and local irregularities, as controlled by the relative weights and length-scales of its constituent kernels.

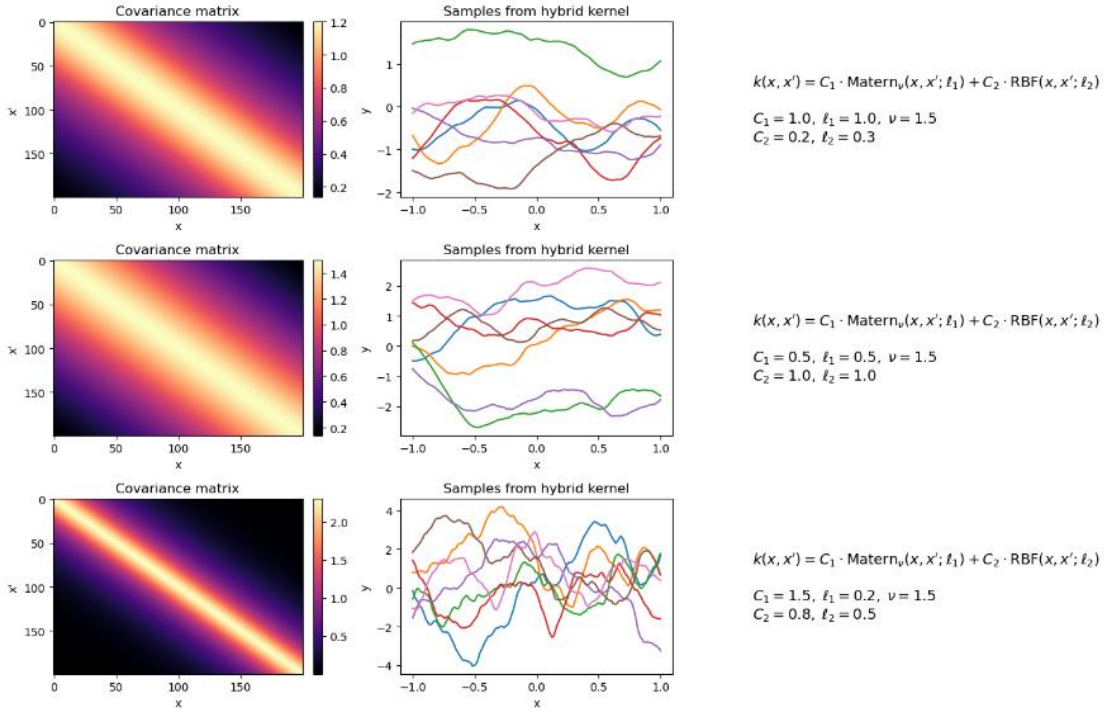


Figure 29: Hybrid kernel and samples.

3.3. Acquisition function

In Bayesian optimization, the acquisition function is the function that guides the selection of the next query point to evaluate the expensive objective function at. After constructing a surrogate probabilistic model, commonly a Gaussian Process as discussed in the previous subchapter, the acquisition function quantifies the utility or potential benefit of evaluating the objective at each point in the design space, balancing *exploration* (uncertainty reduction) and *exploitation* (searching near the current optimum). A widely used acquisition function is *Expected Improvement (EI),* which measures the expected gain in objective value over the current best observation. It is defined with the following equations Brochu et al. (2010):

$$\alpha_{EI}(\bar{x}) = \begin{cases} (\mu(\bar{x}) - f(\bar{x})_{best} - \xi)\Phi(\bar{Z}) + \sigma(\bar{x})\phi(\bar{Z}) \; if \sigma(\bar{x}) > 0 \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad\quad if \sigma(\bar{x}) = 0 \end{cases} \tag{3.3.1}$$

$$\bar{Z} = \frac{\mu(\bar{x}) - f(\bar{x})_{best} - \xi}{\sigma(\bar{x})} \tag{3.3.2}$$

where where the $\Phi(\cdot)$ and the $\phi(\cdot)$ denote the CDF and PDF of the standard normal distribution respectively and ξ is the exploration-exploitation trade-off parameter which is discussed in more details in the following subchapter.

EI is particularly effective in unconstrained optimization, as it naturally trades off between sampling where the surrogate model predicts high values and where the uncertainty is large. However, many real-world problems involve constraints especially in the structural engineering field. The *Constrained Expected Improvement (cEI)* acquisition function extends EI by incorporating the feasibility probability that a candidate point satisfies all constraints, enabling optimization in feasible regions only. It is defined by the following equations:

$$\alpha_{cEI}(\bar{x}) = \alpha_{EI}(\bar{x}) \times \prod_{i=1}^{M_{elements}} P(f_i(\bar{x}) \leq constr_i) \tag{3.3.3}$$

where $P(f_i(\bar{x}) \leq constr_i) = \Phi\left(\frac{constr_i - \mu(\bar{x})_i}{\sigma(\bar{x})_i}\right)$ and $\mu(\bar{x})_i$, $\sigma(\bar{x})_i$ are the mean and uncertainty for the GPs trained on every element.

While EI is computationally efficient and well-suited for noiseless, unconstrained scenarios, it cannot handle constraints directly. On the other hand, cEI handles constraints very well but typically requires additional surrogate models for the constraints and can be more computationally intensive. Overall, the choice between EI and cEI depends on whether the problem includes constraints or not. In addition, some studies in the literature (Ament et al. 2024) apply the logarithm of the Expected Improvement (EI) to mitigate numerical instabilities that can arise in its standard implementation, specifically when EI values become very small, making it difficult to effectively maximize the acquisition function.

Another widely used acquisition function in Bayesian optimization is the Upper Confidence Bound (UCB). The UCB acquisition function selects the next evaluation point by maximizing a sum of the surrogate model's predictive mean and a scaled version of its uncertainty (standard deviation). Formally, UCB is defined as $UCB(x) = \mu(x) + \kappa\sigma(x)$, where $\mu(x)$ and $\sigma(x)$ are the posterior mean and standard deviation of the surrogate model at point x, and $\kappa \geq 0$ is a parameter controlling the trade-off between exploration and exploitation. Larger values of κ encourage exploration of

uncertain regions, while smaller values focus more on exploitation near the current optimum. UCB is especially useful when an explicit balance between risk-taking and reward is desired, or when theoretical regret bounds are important. However, UCB does not naturally handle constraints, and its performance can be sensitive to the choice of κ.

Finally, the Probability of Improvement (PI) is sometimes used for optimization problems. Its definition is similar to the EI, but it is considered to exploit rather than explore the design space Brochu et al. (2010):

$$\alpha_{PI}(\bar{x}) = \Phi\left(\frac{\mu(\bar{x}) - f(\bar{x})_{best}}{\sigma(\bar{x})}\right) \tag{3.3.4}$$

A comparison between all of the acquisition functions mentioned above is shown in Figure 30. The emphasis is on the difference in how the next point is picked going from one iteration to the next.

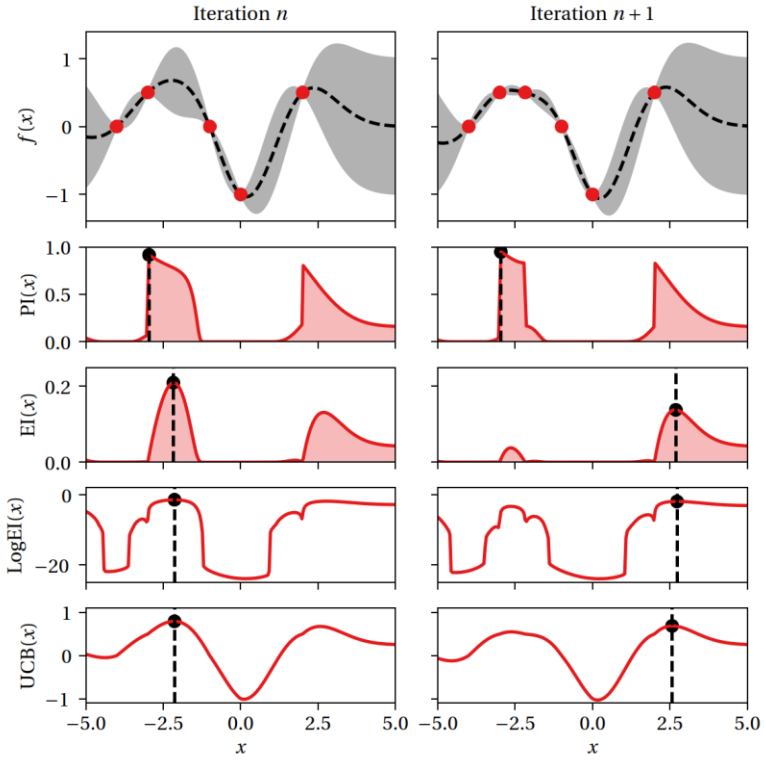


Figure 30: Acquisition function comparison for the choice of the next sampling point. (*Source: Coelho, 2025*)

In this thesis constraints on structural performance metrics are essential to achieve the desired objectives for the different cases studied in Chapter 5. Therefore, the *Constrained Expected Improvement (cEI)* acquisition function is chosen as it provides the most benefits in these scenarios.

3.4. Exploration vs Exploitation

In Bayesian Optimization, an effective balance between *exploration* and *exploitation* is vital for discovering high-performing solutions within a limited evaluation budget. Exploration refers to the investigation of unsampled or poorly understood regions of the design space, while exploitation targets areas that the surrogate model already predicts to be promising. A successful optimization strategy must dynamically adjust this balance as more information becomes available.

This trade-off is typically controlled by acquisition functions such as Expected Improvement (EI) or Constrained Expected Improvement (CEI) which are explained in the previous section. Both of which can be tuned using an exploration parameter ξ. To adaptively manage this parameter throughout the optimization process, dynamic schedules for ξ can be employed based on the current iteration $\boldsymbol{n}$ and the total number of allowed iterations n_{max}. Two commonly used adaptive formulations are the *linear* and *exponential* decay strategies, defined as follows:

Linear adaptative ξ:

$$\xi = \xi_{max} - (\xi_{max} - \xi_{min}) * \frac{n}{\mathrm{n}_{max}} \tag{3.4.1}$$

Exponential adaptive ξ:

$$\xi = \xi_{min} - (\xi_{max} - \xi_{min}) * \exp\left(-\lambda * \frac{n}{\mathrm{n}_{max}}\right) \tag{3.4.2}$$

where, n – current iteration, n_{max} – maximum number of predefined iterations and λ – decay rate.

In the linear scheme, ξ is interpolated from an initial high value to a lower bound in direct proportion to the current iteration, ensuring that a smooth and predictable transition from exploration to exploitation is enforced. In the exponential-decay function, ξ is decreased following an exponential law, resulting in rapid early reduction of exploration that than transitions to exploitation of the best known region in the design space.

3.5. Hyperparameter tuning

Hyperparameter tuning for the kernels used in Gaussian processes is a crucial step in constructing accurate surrogate models for Bayesian optimization. The kernel hyperparameters, such as the length scale and variance, govern the smoothness, amplitude, and overall flexibility of the Gaussian process, directly influencing its ability to capture the underlying structure of the objective function. In practice, these hyperparameters are typically optimized by maximizing the marginal likelihood *P(D)* of the observed data:

$$P(D) = \mathcal{N}(D|\mathbf{0}, K(\boldsymbol{X}, \boldsymbol{X}) + \beta^{-1}\boldsymbol{I}) \tag{3.5.1}$$

where $K = (\boldsymbol{X}, \boldsymbol{X})$ is an $N \times N$ matrix where each element is given by $K_{ij} = k(\boldsymbol{x_i}, \boldsymbol{x_j})$ with representing the chosen kernel function (such as the squared exponential/RBF kernel), $\beta^{-1}\boldsymbol{I}$ is added observation noise and N is the number of training samples. The added observation noise term can be removed in applications where the observations are noiseless. Since the observations of the weight of the structure are obtained deterministically from RFEM6, the observation noise is assumed to be negligible. To prevent numerical instabilities, however, during the fitting of the

Gaussian process a small value for β is added to ensure a positive definite covariance matrix **K** (Rasmussen & Williams, 2005, p.80). This matrix encodes the prior assumptions about the similarity and correlation between all pairs of inputs and forms the covariance structure of the multivariate normal (Gaussian) distribution over the observed targets.

For efficient optimization, it is common to re-tune the kernel hyperparameters periodically during the Bayesian optimization loop. To do that the values of the hyperparameters that maximize the log-marginal likelihood according to Eq. 3.5.2 & 3.5.3 are calculated.

$$\ln\big(P(D)\big) = -\frac{1}{2}\ln(\bar{K} + \beta^{-1}\bar{I}) - \frac{1}{2}\bar{D}^T(\bar{K} + \beta^{-1}\bar{I})^{-1}\bar{D} - \frac{N}{2}\ln(2\pi) \tag{3.5.2}$$

where N is the number of training samples (observations).

$$\bar{h} = argmax\ ln(P(\bar{D}|\bar{h})) \tag{3.5.3}$$

where $\mathbf{h}$ is a vector containing the hyperparameters depending on the used kernels such as ℓ, σ_f and ν. The mathematical derivations related eq. 3.5.2 are outside the scope of this thesis. For further details, the reader is referred to the books of Rasmussen & Williams (2005) *on Gaussian processes for Machine Learning* and *Pattern Recognition and Machine Learning* by Bishop (2006).

To achieve this, the Python package SciPy and its implementation of the L-Broyden-Fletcher-Goldfarb-Shanno-B (L-BFGS-B) algorithm based on the paper of Byrd et al. (1995) is used. This step is necessary since no closed-form solution exists due to the hyperparameters being inside the kernel matrix **K** which has to be inverted. As a result there could be multiple local optima which cannot be maximized analytically with respect to the hyperparameters.

The logarithmic scale and reasonable bounds for the hyperparameters are essential to achieve good results. This approach ensures that the surrogate model remains well-calibrated as new data is acquired.

Additionally, for problems involving multiple input dimensions, automatic relevance determination (ARD) kernel variants are often employed. In ARD kernels, each input dimension is assigned its own separate length scale hyperparameter, allowing the Gaussian process to automatically identify and adapt to the most relevant features of the input space. An example of the squared exponential kernel is given below:

$$k(\bar{x}, \bar{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}\sum_i^D \frac{1}{\ell_i^2}\left|\left|\bar{x} - \bar{x}'\right|\right|^2\right) \tag{3.5.4}$$

where ℓ_i is the length scale for every input dimension D of the vector **x**.

This enhances model flexibility and interpretability, particularly in high-dimensional settings, but also increases the complexity of the optimization problem for hyperparameter learning.

3.6. Sampling strategies

In selecting sample points from the design space, a practical approach is to employ hypercube sampling, which ensures a well-distributed set of candidate points. Conceptually, a line represents a one-dimensional space, a square corresponds to two dimensions, and a cube extends this notion to three dimensions. By analogy, design spaces with four or more dimensions can be represented as hypercubes. One straightforward strategy for generating representative samples is to select points located at the corners of the hypercube, as well as at the midpoints of its edges, faces, volumes, and higher-dimensional analogues as shown in Figure 31.

An alternative approach is to apply the space-filling Latin hypercube sampling (Figure 32) which ensures that a single point is selected from each row and column of the design space, in a manner analogous to the structure of a sudoku puzzle.

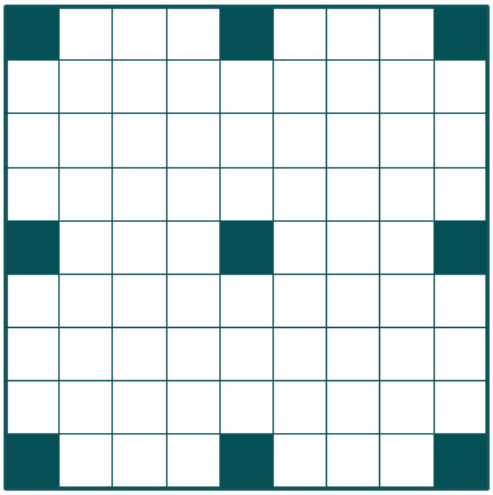


Figure 31: Hypercube sampling the vertices, edges and faces. (*Source: Debney, 2021*)

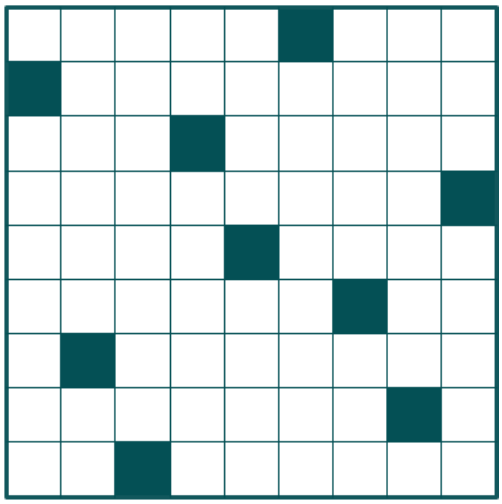


Figure 32: Latin hypercube sampling. (*Source: Debney, 2021*)

In Chapter 5, the Latin hypercube sampling strategy is chosen for the application of the constrained BO algorithm described in Chapter 4.

3.7. Data standardization

Standard preprocessing steps like scaling the inputs and outputs are often applied before running Bayesian optimization to ensure that all input features and hyperparameters live on comparable scales (Rasmussen & Williams, 2005). The standard scaler applies the following formula:

$$x' = \frac{x-\mu}{\sigma} \tag{3.7.1}$$

where μ and σ are the feature-wise mean and standard deviation (in this case the areas of the member), while the robust scaler uses

$$x' = \frac{x-median(x)}{Q_3-Q_1} \tag{3.7.2}$$

(with Q_1 and Q_3 the 25th and 75th percentiles) to reduce the influence of the outliers since for some areas picked by the algorithm the resulting stresses and displacements are significantly higher than for the rest of the profiles. When these scaled inputs are fed into a Gaussian process in the Bayesian optimization, the kernel's ability to measure similarity and speed convergence are greatly improved. The reason is that the optimizer's acquisition function can more reliably compare effects across dimensions that have been normalized to roughly the same range. In all of the case studies the standard scaler is used.

3.8. Principle Component Analysis (PCA)

Principal Component Analysis (PCA) is used as part of the extended constrained BO algorithm discussed in subchapter 4.2. In its deterministic formulation can be understood as a method for projecting high-dimensional data onto a lower-dimensional linear subspace, with the objective of either maximizing the variance of the projected data or minimizing the reconstruction error after projection (Bishop, 2006). For a dataset in D-dimensional space, PCA seeks an optimal subspace of dimension $M \ll D$, which can be interpreted geometrically as a line for $M=1$, a plane for $M=2$, or a hyperplane for higher values of M. In the maximum variance view, the goal is to identify a projection matrix U that maximizes the spread of the projected data, such that the directions of largest variance correspond to the eigenvectors of the sample covariance matrix. This ensures that the subspace captures the most informative features of the data distribution as shown in Figure 33.

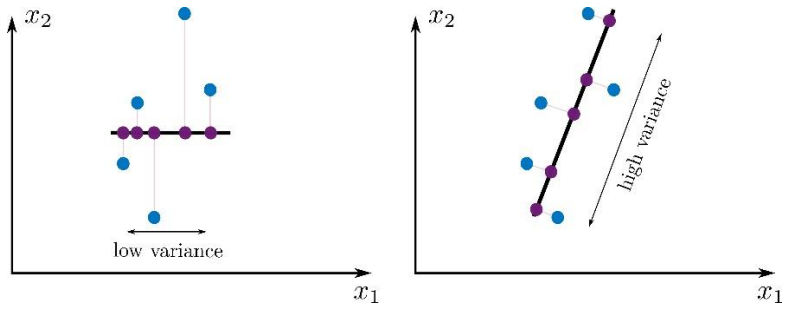


Figure 33: Optimal latent space maximizing the variance of the dataset. (*Source: DSAI, 2024*)

Alternatively, in the minimum error formulation, the projected data points are mapped back into the original space, and the subspace is chosen to minimize the reconstruction error between the original and reconstructed data (Figure 34).

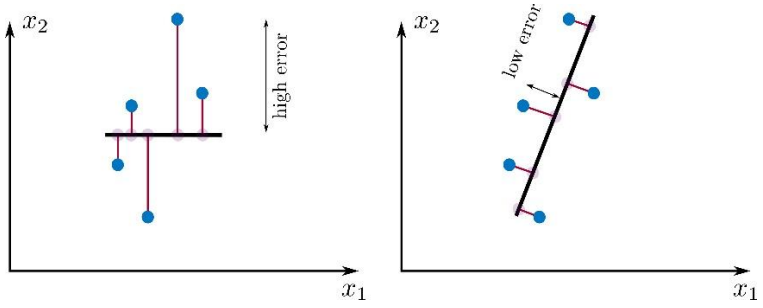


Figure 34: Optimal latent space minimizing the error between original and projected datasets. (*Source: DSAI, 2024*)

Interestingly, both approaches lead to the same optimal solution, as the subspace spanned by the principal eigenvectors of the covariance matrix simultaneously maximizes variance and minimizes reconstruction error.

4. Research Methodology

The objective of this chapter is to introduce the two versions of the constrained BO algorithm that are used in the analysis of the case studies described in Chapter 5. They are developed based on the literature review discussed in Chapter 3.

The Python implementation of the presented algorithms and their integration with RFEM6 can be found in the following GitHub repository: https://github.com/GeorgeNikolov/BO-Tool.

4.1. Constrained BO algorithm

The following pseudo-code outlines the procedure for constrained Bayesian optimization used in this thesis. The goal of it is to minimize a given objective function such as structural weight while enforcing performance constraints, such as stress and stability limits.

Algorithm 1: Constrained Bayesian Optimization for Structural Steel Design

Input : $d \in \mathbb{N}$ (number of design variables), $m \in \mathbb{N}$ (number of elements), bounds $\ell, u \in \mathbb{R}^d$ (lower/upper bounds for each variable), n_{init} (number of initial samples), T_{max} (max iterations), $\varepsilon > 0$ (improvement tolerance),

Output : best design inputs $\mathbf{x^*}$, total structural weight $f(\mathbf{x^*})$ and structural constraints $\mathbf{g(x^*)} = [g_1(x^*), \ldots, g_m(x^*)]^T$

```
# Problem setup
𝒳 ← {x ∈ ℝ^d : ℓ_i ≤ x_i ≤ u_i, i = 1…d}        ▷ Design domain
Define objective f: 𝒳 → ℝ                        ▷ total structural weight
Define constraints g_j: 𝒳 → ℝ,  j = 1,…,m        ▷ governing stress and buckling unity
                                                   checks per element

#Initial sampling
S_0 ← LHS(n_init, 𝒳)                             ▷ Latin Hypercube samples
D ← ∅                                            ▷ Dataset D = { (x_i, f_i, g_i) }
for each x ∈ S_0 do
    (f_x, g_x) ← RFEM(x)
    D ← D ∪ {(x, f_x, g_x)}
    F ← { (x,f,g) ∈ D : g_j ≤ 1 ∀j }             ▷ Subset with only feasible designs, 1 is the unity
                                                   check threshold value per element.
end for
#Data normalization
Fit standard scalers T_x, T_f, T_gj  on {x_i}, {f_i}, {g_{j,i}} respectively
Transform  x̃_i ← T_x(x_i),   f̃_i ← T_f(f_i),   ĝ_{j,i} ← T_gj(g_{j,i})  for all i,j

Initialize GP_f with kernel k_f(θ_f) using {(x̃_i, f̃_i)}, where θ is the set of hyperparameters of the
kernel
For j = 1..m: initialize GP_gj with kernel k_g(θ_gj) using {(x̃_i, ĝ_{j,i})}
#Optimization loop
f* ← argmin_{(x,f,g) ∈ F} f  (best valid design); x* ← x_best (x_best corresponds to the inputs to
obtain f*; no_improve ← 0
```

for $t = 1 \ldots T_{max}$ **do**
  if (t = 1) or (t mod 5 = 0) **then**
    Re-optimize θ_f, $\{\theta_{gj}\}$ by maximizing GP log-marginal likelihood
  end if

  Define acquisition $\alpha(x)$ on $\tilde{x} = T\_x(x)$:
    $\mu_f, \sigma_f^2 \leftarrow GP_f(\tilde{x})$ ▷ predictive mean/variance
    For j: $\mu_{gj}, \sigma_{gj}^2 \leftarrow GP_{gj}(\tilde{x})$
    $p_{feas}(x) \leftarrow \prod_{\{j=1\}}^{m} \Phi((1 - \mu_{gj})/\sigma_{gj})$ ▷ probability of feasibility, and 1 is the unity check threshold
    $EI(x) \leftarrow$ ExpectedImprovement(μ_f, σ_f^2; best $\tilde{f}$)
    $\alpha(x) \leftarrow EI(x) \cdot p_{feas}(x)$ ▷ constrained EI
  $x_{next} \leftarrow \text{argmax}_{\{x \in \mathcal{X}\}} \alpha(x)$ via Differential Evolution (global search) with L-BFGS-B for local refinement (optional)
  $x_{real} \leftarrow x_{next}$ ▷ (already in real space if α used x; else denormalize)
  $(f_{new}, g_{new}) \leftarrow RFEM(x_{real})$
  Append $(x_{real}, f_{new}, g_{new})$ to D
  Update transforms: $\tilde{x}_{new} \leftarrow T_x(x_{real})$, $\tilde{f}_{new} \leftarrow T_f(f_{new})$, $\hat{g}_{new} \leftarrow T_{gj}(g_{new})$
  Update GP_f and $\{GP_{gj}\}$ with the new normalized point
  if $g_{new,j} \leq 1 \ \forall j$ **then** ▷ Are all the constraints below 1?
    if $f_{new} < f^* - \varepsilon$ **then**
      $x^* \leftarrow x_{real}$; $f^* \leftarrow f_{new}$; no_improve ← 0
    else
      no_improve ← no_improve + 1
    end if
  else
    no_improve ← no_improve + 1
  end if

  if no_improve ≥ p **then break** ▷ early termination due no improvement in number of iterations p
end for
# Result extraction
return (x*, f*, g(x*)) ▷ Best feasible design, total structural weight, and the array of constraints per element

To enhance the understanding of the algorithm also the theoretical framework of the constrained BO framework is shown in Figure 35 on the next page. Also, a written description of the algorithm is included in Appendix E: Supplementary Material.

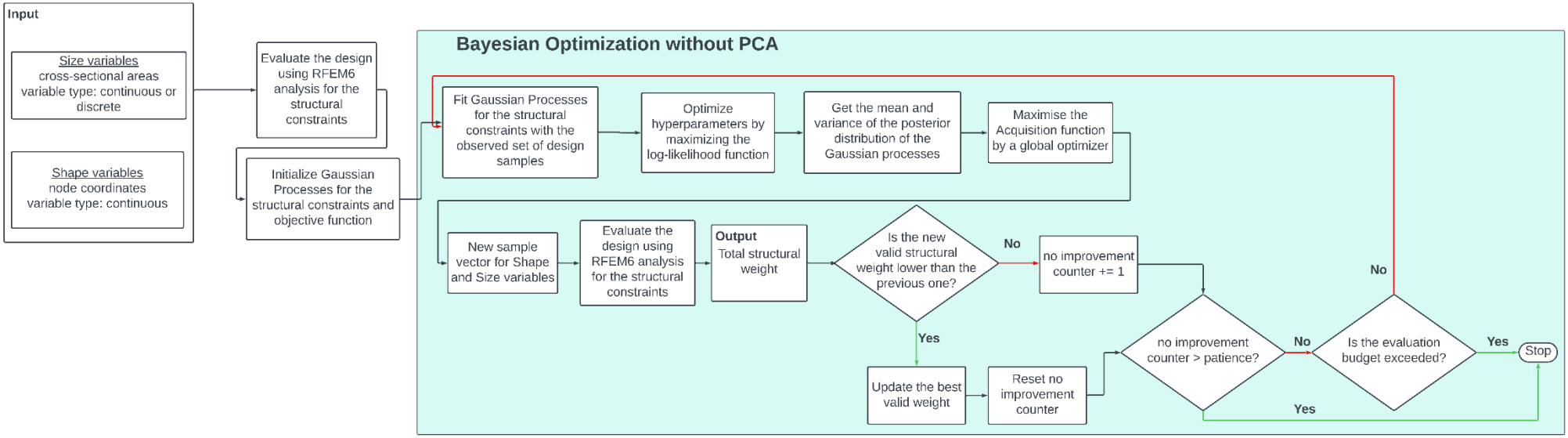


Figure 35: Theoretical framework of the Bayesian optimization without PCA.

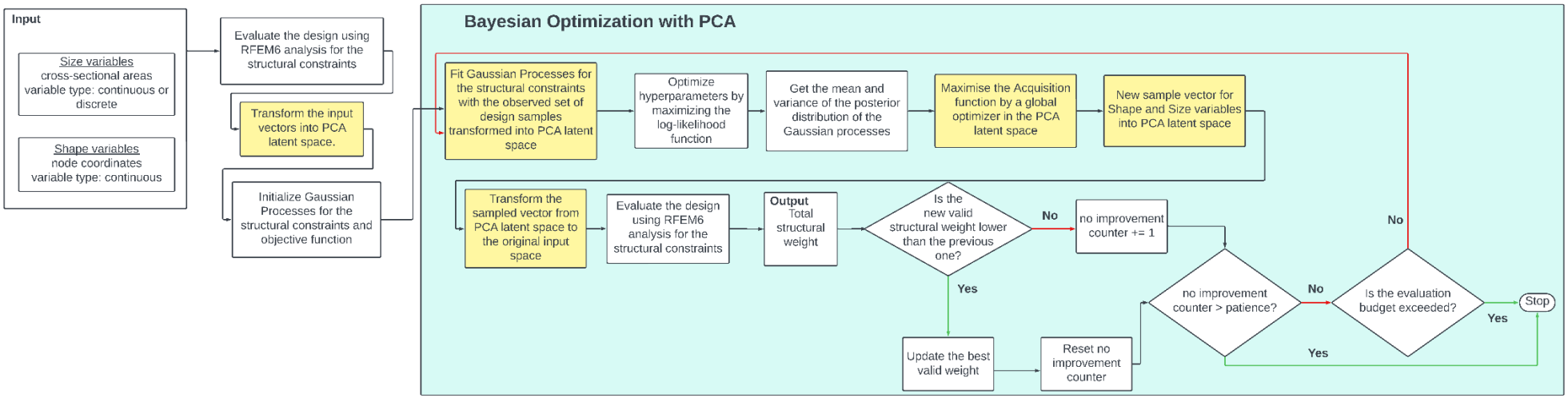


Figure 36: Theoretical framework of the Bayesian optimization with PCA. Changes compared to Figure 35 are highlighted in yellow.

4.2. Constrained BO with PCA algorithm

The following pseudo-code highlights the differences between the algorithm for the constrained BO and constrained BO combined with PCA for dimensionality reduction. This algorithm is used as an additional analysis in the gridshell case studies in subchapter 5.3 & 5.4.

Algorithm 2: Constrained Bayesian Optimization with PCA for Structural Steel Design

Input : $d \in \mathbb{N}$ (number of design variables), $m \in \mathbb{N}$ (number of elements), bounds $\ell, u \in \mathbb{R}^d$ (lower/upper bounds for each variable), n_{init} (number of initial samples), T_{max} (max iterations), $\varepsilon > 0$ (improvement tolerance), $\tau \in (0,1]$ (target explained variance for PCA)

Output : best design inputs $\mathbf{x^*}$, total structural weight $f(\mathbf{x^*})$ and structural constraints $\mathbf{g(x^*)} = [g_1(x^*), \ldots, g_m(x^*)]^T$

Problem setup
$\mathcal{X} \leftarrow \{x \in \mathbb{R}^d : \ell_i \le x_i \le u_i,\ i = 1 \ldots d\}$ ▷ Design domain
Define objective $f: \mathcal{X} \rightarrow \mathbb{R}$ ▷ total structural weight
Define constraints $g_j: \mathcal{X} \rightarrow \mathbb{R},\ j = 1,\ldots,m$ ▷ governing stress and buckling unity checks per element
#Initial sampling
$S_0 \leftarrow \text{LHS}(n_{init}, \mathcal{X})$ ▷ Latin Hypercube samples
$D \leftarrow \emptyset$ ▷ Dataset $D = \{ (x_i, f_i, g_i) \}$
for each $x \in S_0$ **do**
  $(f_x, g_x) \leftarrow \text{RFEM}(x)$
  $D \leftarrow D \cup \{(x, f_x, g_x)\}$
  $F \leftarrow \{ (x,f,g) \in D : g_j \le 1\ \forall j \}$ ▷ Subset with only feasible designs, 1 is the unity check threshold value per element.
end for
#Data normalization
Fit standard scalers T_x, T_f, T_{gj} on $\{x_i\}, \{f_i\}, \{g_{\{j,i\}}\}$ respectively
Transform $\tilde{x}_i \leftarrow T_x(x_i),\ \tilde{f}_i \leftarrow T_f(f_i),\ \hat{g}_{\{j,i\}} \leftarrow T_{gj}(g_{\{j,i\}})$ for all i,j
Choose PCA dimension $q = \min\{ k : \text{cumulative explained variance}(k) \ge \tau \}$
Fit PCA on $\{\tilde{x}_i\}$ to obtain mean μ and loading matrix $U \in \mathbb{R}^{d \times q}$
Map all scaled samples to latent space: $z_i \leftarrow U^T \times (\tilde{x}_i - \mu)$
Define latent search domain $\mathbb{Z}$ as a bounded box covering $\{z_i\}$
Initialize GP_f with kernel $k_f(\boldsymbol{\theta_f})$ using $\{(z_i, \tilde{f}_i)\}$, where $\boldsymbol{\theta}$ is the set of hyperparameters of the kernel
For j = 1..m: initialize GP_{gj} with kernel $k_g(\boldsymbol{\theta_{gj}})$ using $\{(z_i, \hat{g}_{\{j,i\}})\}$

#Optimization loop
$f^* \leftarrow \text{argmin}_{\{(x,f,g) \in F\}} f$ (best valid design); $x^* \leftarrow x_{best}$ (x_{best} corresponds to the inputs to obtain f^*; no_improve $\leftarrow 0$
for $t = 1 \ldots T_{max}$ **do**
  if $(t = 1)$ or $(t \bmod 5 = 0)$ **then**
    Re-optimize θ_f, $\{\theta_{gj}\}$ by maximizing GP log-marginal likelihood
  end if
  Define acquisition $\alpha(z)$ on latent $z \in \mathbb{Z}$:
    $\mu_f(z), \sigma_f^2(z) \leftarrow GP_f(z)$ ▷ predictive mean/variance

For j: $\mu_{gj}(z), \sigma_{gj}^2(z) \leftarrow GP_{gj}(z)$

$p_{feas}(z) \leftarrow \prod_{\{j=1\}}^{m} \Phi((1-\mu_{gj}(z))/\sigma_{gj}(z))$ ▷ probability of feasibility, and 1 is the unity check threshold

$EI(z) \leftarrow$ ExpectedImprovement($\mu_f(z), \sigma_f^2(z)$; best $\tilde{f}$)

$\alpha(x) \leftarrow EI(z) \cdot p_{feas}(z)$ ▷ constrained EI

$z_{next} \leftarrow \text{argmax}_{\{z \in \mathbb{Z}\}} \alpha(z)$ via Differential Evolution (global search) with L-BFGS-B for local refinement (optional)

$\hat{x}_{next} \leftarrow \mu + U \times z_{next}$ ▷ inverse PCA to scaled space

$x_{next} \leftarrow T_x^{-1}(\hat{x}_{next})$ ▷ denormalize to real values in real space

$(f_{new}, g_{new}) \leftarrow RFEM(x_{next})$

normalize and update models

$\hat{f}_{new} \leftarrow T_f(f_{new})$; $\hat{g}_{new} \leftarrow T_{gj}(g_{new})$; $z_{new} \leftarrow U^T (T_x(x_{next}) - \mu)$

Augment GP_f with $(z_{new}, \hat{f}_{new})$; For j: augment GP_{gj} with $(z_{new}, \hat{g}_{\{j,new\}})$

if $g_{new,j} \leq 1 \ \forall j$ **then** ▷ Are all the constraints below 1?

  if $f_{new} < f^* - \varepsilon$ **then**

    $x^* \leftarrow x_{real}$; $f^* \leftarrow f_{new}$; no_improve ← 0

  else

    no_improve ← no_improve + 1

  end if

else

  no_improve ← no_improve + 1

end if

if no_improve ≥ p **then break** ▷ early termination due no improvement in number of iterations p

end for

Result extraction

return (x*, f*, g(x*)) ▷ Best feasible design, total structural weight, and the array of constraints per element

The theoretical framework of the constrained BO framework with PCA is shown in Figure 36 on the previous page and the highlighted blocks outline the difference with the previous implementation. Also, a written description of the algorithm is included in Appendix E: Supplementary Material.

4.3. Integration with RFEM6

The structural finite element models in RFEM6 used in this thesis are formulated using two distinct element types: *truss elements* and *beam elements*. Each element type embodies specific mechanical assumptions and is employed based on the expected load transfer mechanisms within the structural system.

Truss elements, illustrated in Figure 37, are idealized as members capable of sustaining only axial forces. These elements are defined by two nodes, denoted as *i* and *j*, and are formulated under the assumption that the member can transmit forces exclusively along its longitudinal axis. The kinematic constraints restrict each node to a single translational degree of freedom aligned with the element axis (the local x-direction). Consequently, rotational effects, shear deformations, and flexural actions are inherently neglected in this element. The internal forces are limited to axial forces, represented by N_i and N_j at the respective nodes. These elements are used in the modelling of the pin-connected cantilever truss structure in subchapter 5.2 where members are primarily subjected to tensile or compressive forces. This simplification yields computational efficiency but limits the representation of more complex structural phenomena, such as bending or torsional effects, however, these are not expected to occur in the studied case due to the fundamental characteristics of a truss structure.

Beam elements, shown in Figure 38, provide a more comprehensive representation of structural behaviour. Unlike truss elements, beam elements have six degrees of freedom per node: three translational (along the x, y, and z axes) and three rotational (about the x, y, and z axes). This formulation enables the simulation of axial, shear, bending, and torsional responses. At each node, the element can transmit not only axial forces (N_i, N_j), but also shear forces ($V_{y,i}$, $V_{z,I}$,$V_{y,j}$,$V_{z,j}$), torsional moments ($M_{x,i}$, $M_{x,j}$), and bending moments about the local axes ($M_{y,i}$, $M_{z,i}$, $M_{y,j}$, $M_{z,j}$). Beam elements are well-suited for the analysis of members in frames and continuous structures where combinations of axial force, bending moment, shear force, and torsion are significant. The enhanced formulation of beam elements enables the accurate modelling of a wide range of structural systems, including beams, columns, and frames, capturing the complex interplay of internal forces and deformations. These elements are used in the analysis of the various case studies in Chapter 5.

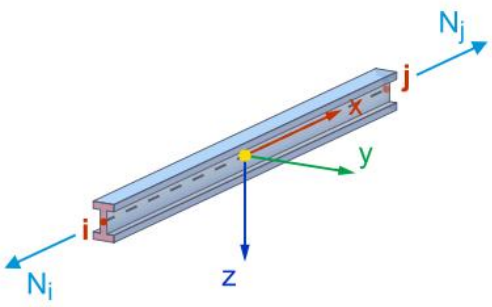


Figure 37: RFEM6 - Truss element.

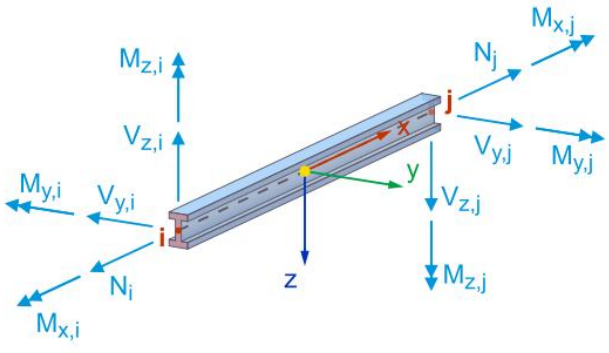


Figure 38: RFEM6 - Beam element.

In the following subchapters all the material-specific unity checks according to the common EN 1993-1-1:2005 that are used to model the constraints in the optimisation problems outlined in Chapter 5 are discussed. These checks are calculated per element by the RFEM6 software package.

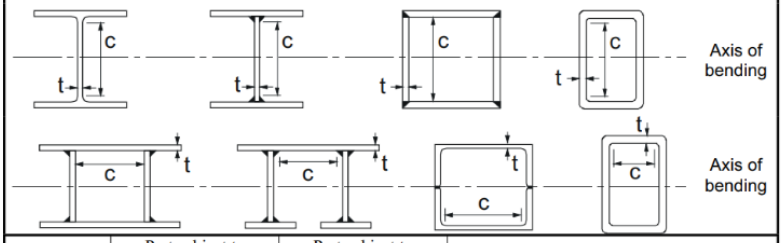
4.3.1. Cross-section classification

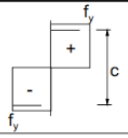
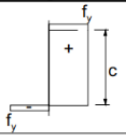
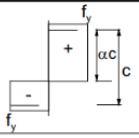
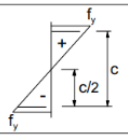
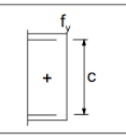
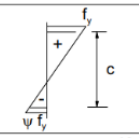
First all of the cross-sections have to be classified according to Table 5.2 in EN 1993-1-1:2005; 5.6 shown below. The cross-sections typologies mostly used in this thesis are the rectangular hollow sections (RHS) and circular hollow sections (CHS). The relevant parts of Table 5.2 are shown in Figure 39 and Figure 40. The lowest class of a subpanel determines the classification of the complete cross-section. For, example in an RHS profile if a one subpanel is classified in class 1 and another in class 3, the complete cross-section is classified in class 3. In case the subpanels fall outside the classes they are considered in class 4 and the reduced effective properties of the cross-section must be calculated according to EN 1993-1-5 Table 4.1 & 4.2 due to local buckling of the cross-section. This classification determines if the cross-sectional resistance to compression, bending and combination of the two are determined according to their plastic or elastic properties (e.g. area, section modulus). The general recommendation according to the Eurocode is the following:

- Class 1 - Plastic resistance
- Class 2 - Plastic resistance
- Class 3 - Elastic resistance
- Class 4 - Elastic resistance with effective properties

Table 5.2 (sheet 1 of 3): Maximum width-to-thickness ratios for compression parts

Internal compression parts



Class	Part subject to bending	Part subject to compression	Part subject to bending and compression
Stress distribution in parts (compression positive)			
1	$c/t \le 72\varepsilon$	$c/t \le 33\varepsilon$	when $\alpha > 0{,}5$: $c/t \le \frac{396\varepsilon}{13\alpha - 1}$ when $\alpha \le 0{,}5$: $c/t \le \frac{36\varepsilon}{\alpha}$
2	$c/t \le 83\varepsilon$	$c/t \le 38\varepsilon$	when $\alpha > 0{,}5$: $c/t \le \frac{456\varepsilon}{13\alpha - 1}$ when $\alpha \le 0{,}5$: $c/t \le \frac{41{,}5\varepsilon}{\alpha}$
Stress distribution in parts (compression positive)			
3	$c/t \le 124\varepsilon$	$c/t \le 42\varepsilon$	when $\psi > -1$: $c/t \le \frac{42\varepsilon}{0{,}67 + 0{,}33\psi}$ when $\psi \le -1^{*)}$: $c/t \le 62\varepsilon(1-\psi)\sqrt{(-\psi)}$

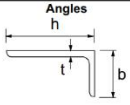
$\varepsilon = \sqrt{235/f_y}$	f_y	235	275	355	420	460
	ε	1,00	0,92	0,81	0,75	0,71

*) $\psi \le -1$ applies where either the compression stress $\sigma \le f_y$ or the tensile strain $\varepsilon_y > f_y/E$

Figure 39: Classification of internal sub-panels. (*Source: EN 1993-1-1:2005*)

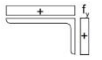
Table 5.2 (sheet 3 of 3): Maximum width-to-thickness ratios for compression parts

Angles

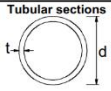


Refer also to "Outstand flanges" (see sheet 2 of 3)

Does not apply to angles in continuous contact with other components

Class	Section in compression
Stress distribution across section (compression positive)	
3	$h/t \le 15\varepsilon$: $\frac{b+h}{2t} \le 11{,}5\varepsilon$

Tubular sections



Class	Section in bending and/or compression
1	$d/t \le 50\varepsilon^2$
2	$d/t \le 70\varepsilon^2$
3	$d/t \le 90\varepsilon^2$ **NOTE** For $d/t > 90\varepsilon^2$ see EN 1993-1-6.

$\varepsilon = \sqrt{235/f_y}$	f_y	235	275	355	420	460
	ε	1,00	0,92	0,81	0,75	0,71
	ε^2	1,00	0,85	0,66	0,56	0,51

Figure 40: Classification of tubular sections. (*Source: EN 1993-1-1:2005*)

After the cross-section is classified the checks for strength, stiffness and stability have to be calculated. They are outlined in subchapter 4.2 to 4.4.

4.3.2. Strength checks (ULS)

In this subchapter the checks that determine the strength of the cross-section under the Ultimate Limit State combinations are shown. The general formulas are shown for clarity, however, the cross-sectional properties are axis dependent and should be calculated in the major principle y-axis and minor principle z-axis of any given cross-section.

Compression & Tension $N_{c,Rd} = \frac{A * f_y}{\gamma_{M0}}$ for class 1,2 and 3 under compression/tension	EN 1993-1-1:2005; 6.2.3, 6.2.4; Eq. (6.6 & 6.10)
$N_{c,Rd} = \frac{A_{eff} * f_y}{\gamma_{M0}}$ for class 4 under compression	EN 1993-1-1:2005; 6.2.4; Eq. (6.11)
$$\frac{N_{Ed}}{N_{c,Rd}} \leq 1.0$$ where N_{Ed} is the applied design compression force. $N_{c,Rd}$ is the design resistance of the cross-section. γ_{M0} is the partial factor for the cross-sectional resistance and is equal to 1 for steel. A_{eff} is the reduced cross-sectional area due to local buckling effects.	EN 1993-1-1:2005; 6.2.4; Eq. (6.9)
Shear $$V_{pl,Rd} = \frac{A_v * (f_y/\sqrt{3})}{\gamma_{M0}}$$ where $V_{pl,Rd}$ is the design plastic shear resistance A_v is the shear area calculated as: $A_v = Ah/(b+h)$ for RHS with uniform thickness and load parallel to the depth. $A_v = 2A/\pi$ for CHS with uniform thickness.	EN 1993-1-1:2005; 6.2.6; Eq. (6.18)
$$\frac{V_{Ed}}{V_{pl.Rd}} \leq 1.0$$ where V_{Ed} is the applied design shear force.	EN 1993-1-1:2005; 6.2.6; Eq. (6.17)
Shear & Torsion In the analysis of the gridshells in subchapter 5.3 and 5.4 often some small torsional moments are observed. This is accounted for with following	

reduction to the design plastic shear resistance for hollow sections: $$V_{pl,T,Rd} = [1 - \frac{\tau_{t,Ed}}{(f_y/\sqrt{3})/\gamma_{M0}}]V_{pl,Rd}$$ where $\tau_{t,Ed}$ is the design shear stresses due to St. Venant torsion.	EN 1993-1-1:2005; 6.2.7; Eq. (6.28)
$$\frac{V_{Ed}}{V_{pl.T,Rd}} \leq 1.0$$	EN 1993-1-1:2005; 6.2.7; Eq. (6.25)
Bending moment $M_{c,Rd} = M_{pl,Rd} = \frac{W_{pl}f_y}{\gamma_{M0}}$ for class 1 and 2	EN 1993-1-1:2005; 6.2.5; Eq. (6.13)
$M_{c,Rd} = M_{el,Rd} = \frac{W_{el,min}f_y}{\gamma_{M0}}$ for class 3	EN 1993-1-1:2005; 6.2.5; Eq. (6.14)
$M_{c,Rd} = \frac{W_{eff,min}f_y}{\gamma_{M0}}$ for class 4 where W_{pl} is the plastic section modulus $W_{el,min}$ and $W_{eff,min}$ are elastic section modulus corresponding to the maximum elastic stress	EN 1993-1-1:2005; 6.2.5; Eq. (6.15)
$$\frac{M_{Ed}}{M_{c,Rd}} \leq 1.0$$	EN 1993-1-1:2005; 6.2.5; Eq. (6.12)
Bending moment & shear Acc. to 6.2.8(2) or 6.2.10(2), the effect of the shear force/stress in the y-, z- principle axes of the cross-section can be neglected if it is less than half of the shear resistance. Otherwise, the moment resistance should be calculated with a reduced yield strength by: $$(1 - \rho)f_y$$ where $\rho = (\frac{2V_{Ed}}{V_{pl,Rd}} - 1)^2$ *or* $(\frac{2V_{Ed}}{V_{pl,T,Rd}} - 1)^2$ if torsion is present.	EN 1993-1-1:2005; 6.2.8; Eq. (6.29)
Axial & shear stress For the elastic verification the following yield criterion can be used:	

$$\left(\frac{\sigma_{x,Ed}}{f_y/\gamma_{M0}}\right)^2+\left(\frac{\sigma_{z,Ed}}{f_y/\gamma_{M0}}\right)^2-\left(\frac{\sigma_{x,Ed}}{f_y/\gamma_{M0}}\right)\left(\frac{\sigma_{z,Ed}}{f_y/\gamma_{M0}}\right)+3\left(\frac{\tau_{x,Ed}}{f_y/\gamma_{M0}}\right)^2\leq 1.0$$	EN 1993-1-1:2005; 6.2.1; Eq. (6.1)
Biaxial bending & axial force & shear force Acc. to 6.2.8(2) or 6.2.10(2), the effect of the shear force/stress in the y-, z- principle axes of the cross-section can be neglected if it is less than half of the shear resistance. Otherwise, refer to the Bending moment & shear section above.	
Allowance for the effect of the axial force on the plastic moment resistance does not have to be made when both of the following criteria are satisfied: $N_{Ed}\leq 0.25N_{pl,Rd}$	EN 1993-1-1:2005; 6.2.9; Eq. (6.33)
$N_{Ed}\leq\frac{0.5h_w t_w f_y}{\gamma_{M0}}$ In case allowance has to be made:	EN 1993-1-1:2005; 6.2.9; Eq. (6.34)
$a_w=(A-2bt)/A$ but $a_w\leq 0.5$ for hollow sections $a_f=(A-2ht)/A$ but $a_f\leq 0.5$ for hollow sections	
$$n=\frac{N_{Ed}}{N_{pl,Rd}}$$	EN 1993-1-1:2005; 6.2.9; Eq. (6.39)
$$M_{N,y,Rd}=\frac{M_{pl,y,Rd}(1-n)}{1-0.5a_w}\ but\ M_{N,y,Rd}\leq M_{pl,y,Rd}$$	EN 1993-1-1:2005; 6.2.9; Eq. (6.40)
$$M_{N,z,Rd}=\frac{M_{pl,z,Rd}(1-n)}{1-0.5a_f}\ but\ M_{N,z,Rd}\leq M_{pl,z,Rd}$$	EN 1993-1-1:2005; 6.2.9; Eq. (6.41)
$$[\frac{M_{y,Ed}}{M_{N,y,Rd}}]^{\alpha}+[\frac{M_{z,Ed}}{M_{N,z,Rd}}]^{\beta}\leq 1.0$$ in which α and β are constants which may be taken conservatively as 1 or calculated as: α = 2; β = 2 for circular hollow sections	

$\alpha = \beta = \frac{1.66}{1 - 1.13n^2}\ but\ \alpha = \beta \leq 6$	

4.3.3. Stability checks (ULS)

In addition to checks for strength, flexural buckling is evaluated for each element around both its major and minor axes according to EN 1993-1-1:2005. The calculations are performed by the RFEM6 software package and discussed below for completeness.

Flexural buckling	
$N_{cr,y} = \pi^2 * E * \frac{I_y}{L_{cr,y}^2}$	EN 1993-1-1:2005; 6.3.1.2(1)
$N_{cr,z} = \pi^2 * E * \frac{I_z}{L_{cr,z}^2}$	EN 1993-1-1:2005; 6.3.1.2(1)
where $N_{cr,y}$ is the elastic critical force around the major y-axis $N_{cr,z}$ is the elastic critical force around the minor z-axis E is the modulus of elasticity I_y is the moment of inertia around the major y-axis I_z is the moment of inertia around the minor z-axis $L_{cr,y}$ is the buckling length for pin-pin support conditions in the major y-axis. It is equal to the length of the element. $L_{cr,z}$ is the buckling length for pin-pin support conditions in the minor z-axis. It is equal to the length of the element.	
$\overline{\lambda_y} = \sqrt{\frac{Af_y}{N_{cr,y}}}\ for\ Class\ 1,2,and\ 3$	EN 1993-1-1:2005; 6.3.1.3(1); Eq. 6.50
$\overline{\lambda_z} = \sqrt{\frac{Af_y}{N_{cr,z}}}\ for\ Class\ 1,2,and\ 3$	EN 1993-1-1:2005; 6.3.1.3(1); Eq. 6.50
$\bar{\lambda} = \sqrt{\frac{A_{eff}f_y}{N_{cr}}}\ for\ Class\ 4$	EN 1993-1-1:2005; 6.3.1.3(1); Eq. 6.51
where	

$\bar{\lambda}$ is the non-dimensional slenderness around the corresponding axis Buckling effects can be neglected for $\bar{\lambda} \leq 0.2 \; or \; \frac{N_{Ed}}{N_{cr}} \leq 0.04$	EN 1993-1-1:2005; 6.3.1.2(4)
The following calculations are axis dependent but are written only once for brevity. $\Phi = 0.5[1 + \alpha(\bar{\lambda} - 0.2) + \bar{\lambda}^2]$ where Φ is a value to determine reduction factor χ α is the imperfection factor depending on the buckling curve (see Figure 42) for the cross-section. This is determined according to Table 6.1 and 6.2 shown in Figure 41 and Figure 43, respectively.	EN 1993-1-1:2005; 6.3.1.2(1)
$\chi = \frac{1}{\Phi + \sqrt{\Phi^2 - \bar{\lambda}^2}} \; but \; \chi \leq 1.0$	EN 1993-1-1:2005; 6.3.1.2(1); Eq. 6.49
Then: $N_{b,Rd} = \frac{\chi A f_y}{\gamma_{M1}} \; for \; Class \; 1,2 \; and \; 3$	EN 1993-1-1:2005; 6.3.1.1(3); Eq. 6.47
$N_{b,Rd} = \frac{\chi A_{eff} f_y}{\gamma_{M1}} \; for \; Class \; 4$	EN 1993-1-1:2005; 6.3.1.1(3); Eq. 6.48
Finally, $\frac{N_{Ed}}{N_{b,Rd}} \leq 1.0$ is checked for both the major y-axis and minor z-axis	EN 1993-1-1:2005; 6.3.1.1(1); Eq. 6.46

Table 6.1: Imperfection factors for buckling curves

Buckling curve	a_0	a	b	c	d
Imperfection factor α	0,13	0,21	0,34	0,49	0,76

Figure 41: Table 6.1 for imperfection factors. (*Source: EN 1993-1-1:2005*)

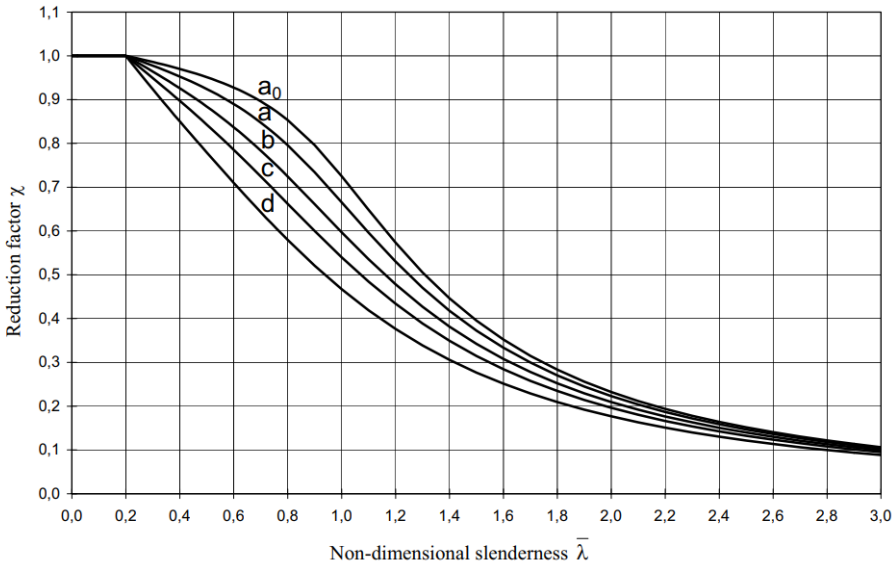


Figure 6.4: Buckling curves

Figure 42: Buckling curves used to determine the reduction factor χ. (Source: EN 1993-1-1:2005)

Table 6.2: Selection of buckling curve for a cross-section

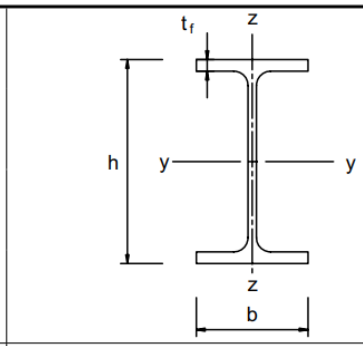
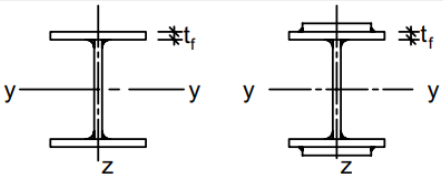
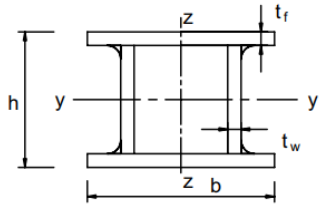
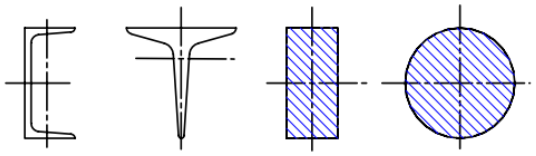
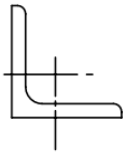
Cross section		Limits		Buckling about axis	Buckling curve S 235 S 275 S 355 S 420	Buckling curve S 460
Rolled sections		$h/b > 1,2$	$t_f \leq 40$ mm	y – y z – z	a b	a_0 a_0
			40 mm < $t_f \leq 100$	y – y z – z	b c	a a
		$h/b \leq 1,2$	$t_f \leq 100$ mm	y – y z – z	b c	a a
			$t_f > 100$ mm	y – y z – z	d d	c c
Welded I-sections			$t_f \leq 40$ mm	y – y z – z	b c	b c
			$t_f > 40$ mm	y – y z – z	c d	c d
Hollow sections			hot finished	any	a	a_0
			cold formed	any	c	c
Welded box sections			generally (except as below)	any	b	b
			thick welds: $a > 0,5t_f$ $b/t_f < 30$ $h/t_w < 30$	any	c	c
U-, T- and solid sections				any	c	c
L-sections				any	b	b

Figure 43: Table 6.2 used for selection of the buckling curved depending on the cross-section and material. (*Source: EN 1993-1-1:2005*)

5. Application of Bayesian optimization

In this chapter, different cases are described that show the application of the constrained Bayesian optimization algorithms discussed in Chapter 4 to the weight optimisation of macrostructures. Each case increases in number of elements analysed and complexity of the optimization problem with the objective of ultimately analyse the 3D symmetric 9x9 gridshell discussed in subchapter 5.4. The goal of the optimization in all of the case studies is to minimize the total structural weight expressed by the following formula:

$$W = \sum_i \rho_i \, A_i L_i \tag{5.1}$$

where W is the total weight of the gridshell [kg], ρ is the material density [kg/m³], A_i is the cross-sectional area of element i [m²] and L_i is the length of element i [m].

Furthermore, for all of the studied cases discussed in this chapter the update of the hyperparameters is chosen to be every 5th iteration as a balance between computation cost and prediction quality. Similar suggestion is given by Shende et al. (2021) who state that estimating the hyperparameters every iteration might lead to overfitting of the data.

The computer processor used in this thesis is 11th Gen Intel(R) Core(TM) i7-14700K, 3.40 GHz which coupled with the time necessary to communicate between the Python script and RFEM6 determines the total calculation time for all the analysed cases.

5.1. 1D optimization case: Cantilever Beam with size variables

The objective of this first case study is to evaluate the performance of the BO framework, described in the previous chapter, on a simple 1D structural system with a database of 516 most commonly used standardized profiles as a starting point. This case also serves as a proof-of-concept for integrating a Python-based BO framework with the RFEM6 software via its dedicated Python-based API.

5.1.1. Problem Definition & Analysis

In this test case, a basic structural system is considered: a single-bar cantilever beam, illustrated in Figure 44. The corresponding RFEM6 model with beam element is shown in Figure 45. The member has a total length of L=4 m and is fully restrained at node 1, where all translations and rotations in the x-, y-, and z-directions are fixed. At the free end at node 2, a concentrated point load of F=10 kN is applied, creating a load case for analysis.

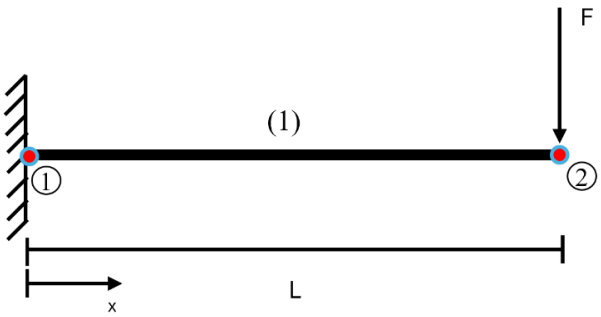


Figure 44: Case study: Cantilever beam - Structural layout.

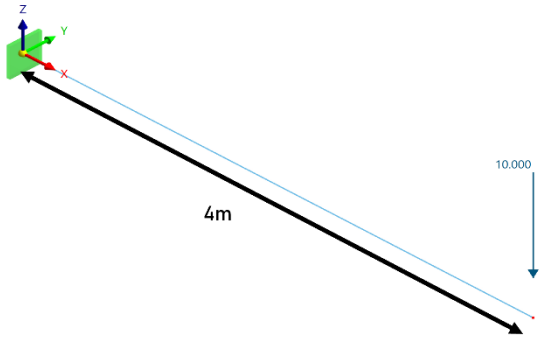


Figure 45: Case study: Cantilever beam - RFEM6 model.

The beam is assumed to be made of structural steel grade S235, with a yield strength of $f_y = 235MPa$. The material strength introduces a clear constraint which is that the maximum stress within the element must remain below this threshold. In addition to the strength requirement, deflection requirements are also considered for this case study. The maximum vertical deflection at the free end must not exceed L/250 which, for the present case, results in a deflection limit of 16 mm. This is a conservative choice, since in practice often 2L/250 is used as a constraint for cantilever structures. In this way the optimisation task becomes more challenging for the BO framework.

The optimization problem is formulated as a size optimization task, where the cross-sectional profile of the beam is selected from a database of standardised or in mathematical terms - discrete profiles provided in Table 16, Appendix B: 1D optimization problem: Cantilever Beam. The objective is to minimize the self-weight of the structure while satisfying both the stress and deflection constraints.

To validate the outcomes of the optimization process, the results obtained from the finite element analysis in RFEM6 are compared with the best-performing cross-sections derived from the analytical solution.

After calculating the unity check for strength and stiffness for the case above the valid results are sorted based on the area in ascending order. Table 1 shows the results of the top 10 most optimal cross sections out of the total 516 elements based on an analytical solution using the formulas (Eq. 5.1.1 and Eq. 5.1.2) below. From the values it becomes clear that the deflection of the beam governs the design.

$$\sigma = \frac{N}{A} + \frac{M_y}{W_y} + \frac{M_z}{W_z} \leq f_y = 235\ MPa \tag{5.1.1}$$

$$\delta = \frac{1}{3}\frac{Fl^3}{EI_y} \leq \frac{L}{250} = 16mm \tag{5.1.2}$$

Table 1: Cantilever beam case: Optimal cross-sections in terms of weight. Analytical solution.

Profile	Area [mm^2]	Weight [kg]	U.C strength	U.C stiffness
CHS 323.9x5	5009,0	157,28	0,433	0,997
IPE 300	5381,0	168,93	0,306	0,760
UNP 300	5880,0	184,63	0,318	0,791
RHS 300x200x6.3	6099,0	191,54	0,326	0,811
CHS 323.9x6.3	6286,0	197,38	0,348	0,801
IPE 330	6261,0	198,76	0,239	0,539
SHS 260x6.3	6351,0	199,39	0,326	0,935
RHS 260x180x8	6715,0	206,11	0,346	0,994
CHS 355.6x6.3	6913,0	216,97	0,287	0,602
IPE 360	7273,0	226,39	0,188	0,390

For this case study, the following kernels with their respective hyperparameter settings have been found empirically to produce good results. Noteworthy is that an *Automatic Relevance Determination (ARD) kernel* version has been used for both RBF and Matérn kernels) in which each input dimension is equipped with its own length–scale hyperparameter that is updated every 5th iteration:

- Gaussian process for the weight:
  - *Matérn ($\boldsymbol{\ell}$, v=0.5) + RBF($\boldsymbol{\ell}$), $\boldsymbol{\ell} \in [0.01, 1000] \in R^D$, D = 1 dimension*
- Gaussian process for the stress constraint:
  - *Matérn ($\boldsymbol{\ell}$, v=0.5), $\boldsymbol{\ell} \in [0.01, 10] \in R^D$, D = 1 dimension*
- Gaussian process for the displacement constraint:
  - *Matérn($\boldsymbol{\ell}$, v=0.5), $\boldsymbol{\ell} \in [0.01, 10] \in R^D$, D = 1 dimension*

In Figure 46 the optimization progress is shown. It can be seen that the algorithm converges around the 4th iteration to a stable solution which is the third most optimal profile – UNP 300 for the given test case when compared to the analytical solution in Table 1. The maximum stress is 74.75 MPa due to bending around the y-axis and the maximum deflection is 12.8mm both of which are within the constraint limits. The FEM results are shown in Figure 125 and Figure 126 in Appendix B: 1D optimization problem: Cantilever Beam.

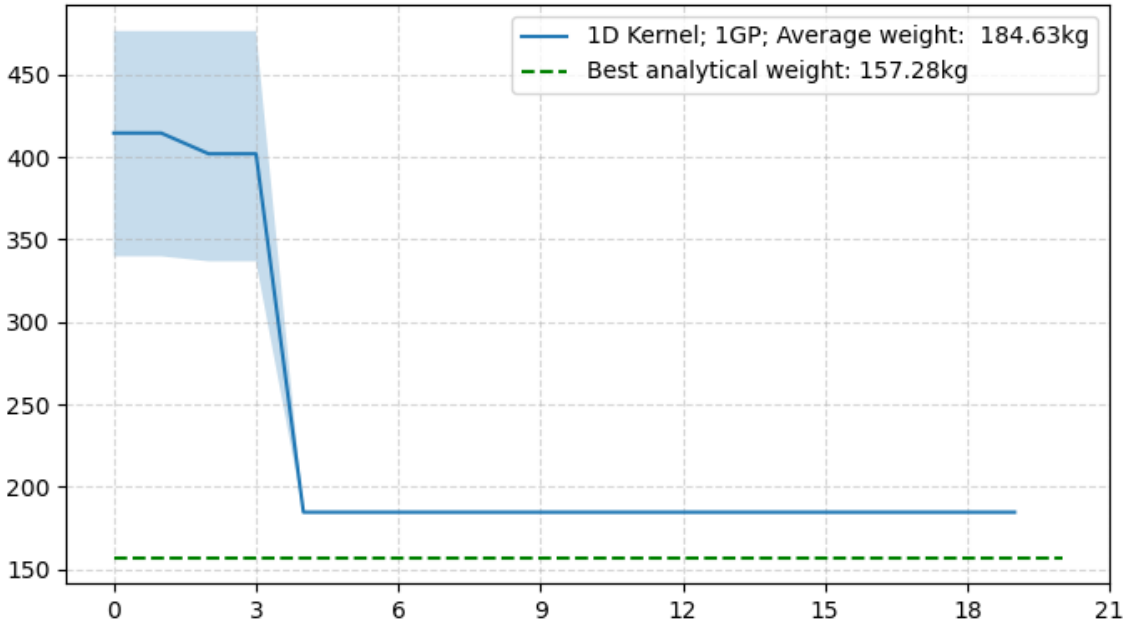


Figure 46: Bayesian optimization progress based on the full database of 516 profiles. Median objective function values and corresponding 95% confidence interval.

It is found that for this case, the optimal result is not reached due to the use of the many *diverse* cross-section types that are included in the analysis. This diversity introduces a lot of variability for the moments of inertia around the principle axes of the cross-sections which poses a challenge for the Gaussian process to model the functions for the constraints in a reasonable way. For example, symmetric cross-sections such as CHS have the same moment of inertia in both the weak and strong axes whereas the asymmetric ones such as HEA, RHS etc. have different values depending on the principal direction. Also, when a new design point is picked by the algorithm to be evaluated in RFEM6, it might be translated to a different cross-section due to similar areas. For example, CHS 323.9x6.3, IPE 330, SHS 260x6.3 in Table 1 have areas corresponding to 6286.0, 6261.0, 6351.0 mm^2 which leads to similar weight of the structure. A further challenge lies in distinguishing between open and closed cross-sections. Open sections are generally more prone to lateral torsional buckling, whereas closed sections provide greater resistance against it. More insights into the behaviour of the algorithm is presented in Figure 106 - Figure 124 in Appendix B: 1D optimization problem: Cantilever Beam.

To put the result in perspective the best analytical weight and the one obtained by the algorithm in relation to all possible weights in the design domain are shown in Figure 47. This represents the objective function for the total weight of the structure.

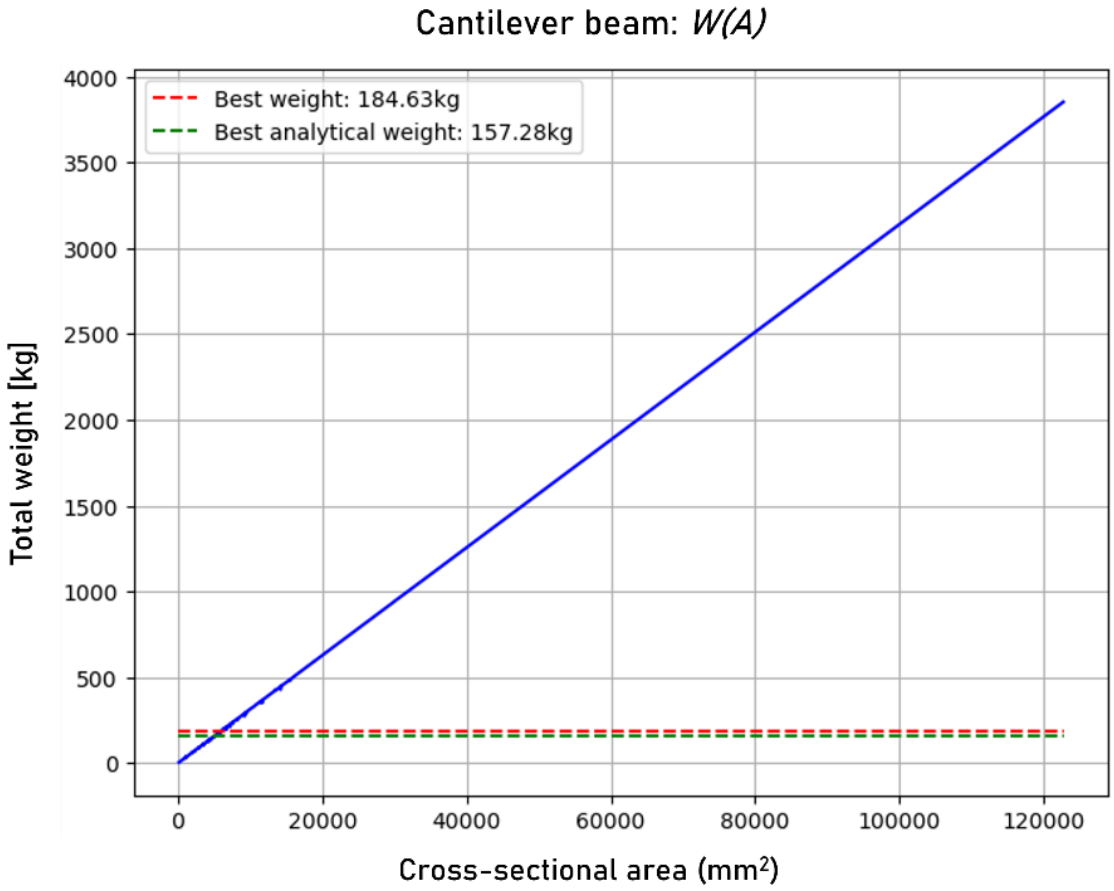


Figure 47: Optimal weight in relation to the full dataset of cross-sections.

In Figure 48 the same chart is shown but this time resulting from the BO process and the found optimal profile is highlighted. The Gaussian process is correctly approximating a linear function where the samples have been fitted. In the region where there are no samples the mean prediction with an increasing uncertainty is plotted as the function moves away from the samples.

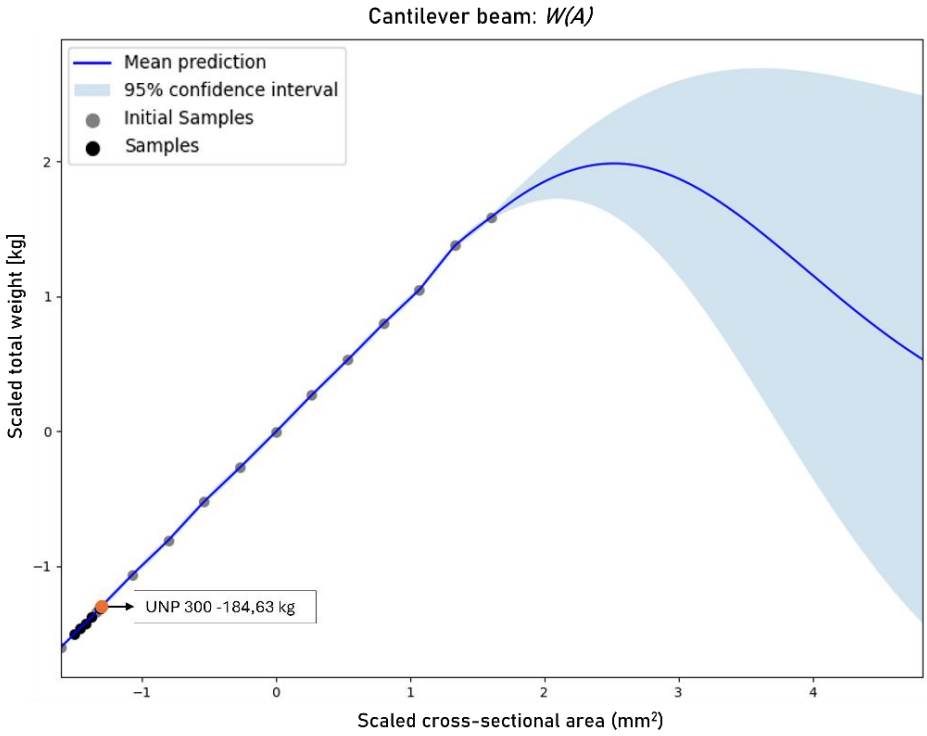


Figure 48: 1D Gaussian process for the weight objective function.

The choice for the adaptive ξ function is based on the functions presented in Chapter 3.4 each of which is designed to introduce ξ in a distinct manner over the course of the optimization. Although, exponential decay enables aggressive early exploration in this case is not optimal. It is found that the linear adaptive function delivered the best behaviour and the most reliable convergence in this application. Consequently, the linear version is used in all of the remaining case studies in this chapter.

Furthermore, since this is a minimization problem the ξ_{max} and ξ_{min} values should be negative since the scaled best weight and the scaled predicted mu value are both negative. If this would be a maximization problem, then all of these variables would be positive. The values of $\xi_{max} = -1$ and $\xi_{min} = -0.01$ are found to be the best for reliable convergence. These values have also been suggested in the PhD thesis by Lizotte D. (2008), who argues that an adaptive cooling schedule for the ξ parameter going from exploration (higher values of ξ) to exploitation (smaller values of ξ) slightly improves the performance for short runs (smaller than 30 iterations).

In conclusion for this case study, the integration of the Python of implementation of the constrained BO with the RFEM6 API proved to be successful. Furthermore, using a *diverse* set of cross-sections is undesirable because of the large variability in moments of inertia about their principal axes. Therefore, for the subsequent case studies, a single cross-section type is adopted.

5.2. 2D optimization case: Cantilever Truss with shape and size variables

This chapter presents the transition from the one-dimensional to the two-dimensional case study. The purpose is to investigate a more complex structural system for which reference solutions are available in literature, thereby enabling a rigorous evaluation of the performance of the BO framework.

5.2.1. Problem Definition & analysis

The 18-bar cantilever truss introduced by Gholizadeh (2013) is considered (Figure 49). The corresponding model in RFEM6 is set up using the *truss elements* (see Chapter 4.3 for more information) and shown in Figure 50.

This theoretical benchmark problem is frequently employed in the structural optimization literature and therefore provides a reliable basis for comparison with existing studies. The truss spans a total length of 1,250 inches (31.75 m) with a height of 250 inches (6.35 m). Its established role as a reference case makes it an appropriate choice for validating the optimization methodology introduced in Chapter 4.

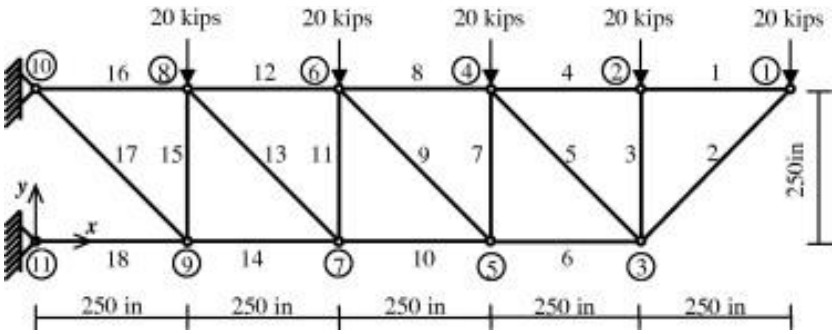


Figure 49: Case study: Cantilever truss - Structural layout. (*Source: Gholizadeh, 2013*)

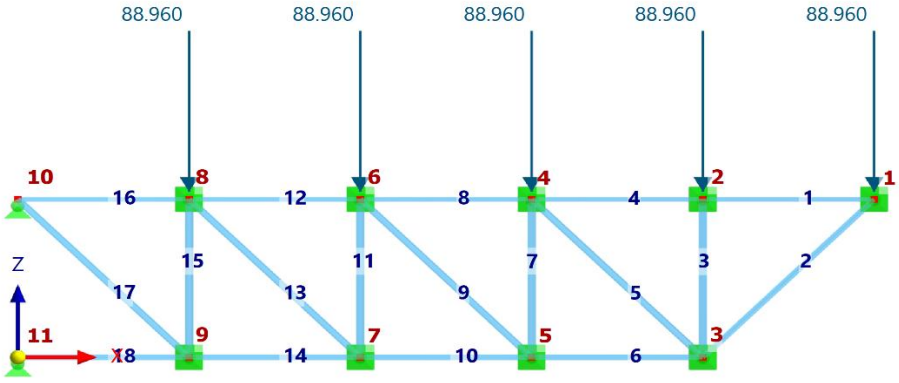


Figure 50: Case study: Cantilever truss - RFEM6 model.

For the numerical study, the truss is subjected to a test load case consisting of a point force of $F = 20\ kips\ (88.96\ kN)$ applied at nodes nr. 8, 6, 4, 2, and 1. In order to maintain consistency with the benchmark formulation, the self-weight of the structure is neglected in this case study.

The boundary conditions are defined such that nodes nr. 10 and 11 are fully restrained in translation along the x-, y-, and z-directions, while their rotational degrees of freedom remain unconstrained. All other nodes are restricted only in the out-of-plane y-direction, ensuring sufficient stability of the structure while allowing realistic deformations to occur. This configuration provides the necessary constraints for a stable analysis and enables the accurate calculation of internal forces.

The material properties of the truss are defined in accordance with the description of Gholizadeh (2013). A density of $\rho = 0.1\ lb/in^3 (2767.99\ kg/m^3)$ is adopted, together with a modulus of elasticity of $E = 10^4\ m^3 (68{,}947.57\ MPa)$. These parameters provide the basis for evaluating the structural response under the prescribed loading conditions.

The reference study does not specify the cross-sectional profiles of the truss members. Based on the observations of the previous case study, a single type of cross-section is used for all of the steel elements of the structure. Therefore, a standardized (discrete) database of 216 circular hollow sections (CHS) profiles is used for the optimization process. The complete database can be found as a subset in Table 16 in Appendix B. This closed profile is chosen due to its symmetric configuration leading to the same moment of inertia around both principle axes of the cross-section. This provides a linear relationship between the area of the cross-section and the moments of inertia, namely as the area increases the moment of inertia increases proportionally. Furthermore, to account for the continuous nature of the Gaussian processes used for the modelling of the objective and constraint functions, a parametric CHS profile is introduced with a fixed standard thickness of $t = 10mm$ and a variable diameter. This parametrization is required because the cross-sectional area is the only input in the algorithm, which can be used to determine only one of the two defining parameters of the section. Naturally, the designer can choose a different thickness or a different parametrization approach.

The first set of constraints is outlined below:

- Geometry (layout variables):

$$(19.685m)775\ in. \leq x_3 \leq 1225\ in. (31.115m)$$
$$(13.335m)525\ in. \leq x_5 \leq 975\ in. (24.765m)$$
$$(6.985m)275\ in. \leq x_7 \leq 725\ in. (18.415m)$$
$$(0.635m)25\ in. \leq x_9 \leq 475\ in. (12.065m)$$
$$(-5.715m) - 225\ in. \leq y_3, y_5, y_7 y_9 \leq 245\ in. (6.223m)$$

- Cross-sections:

$A_1 = A_4 = A_8 = A_{12} = A_{16}$.
$A_2 = A_6 = A_{10} = A_{14} = A_{18}$.
$A_3 = A_7 = A_{11} = A_{15}$.
$A_5 = A_9 = A_{13} = A_{17}$.
$A_i \in \{2.00, 2.25, 2.50, \ldots, 21.25, 21.50, 21.75\}$ (in.2) for the i^{th} element.
$A_i \in \{1290.32, 1451.61, 1612.90, \ldots, 13709.65, 13870.94, 14032.23\}$ (mm^2) for the i^{th} element. The groups of cross-sections are shown in Figure 51.

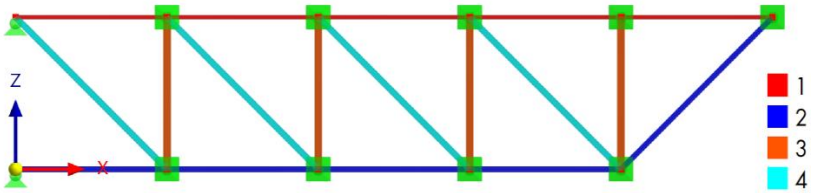


Figure 51: Cantilever truss: Cross-section groups.

Additionally, two sets of structural constraints are imposed to ensure structural safety and stability. First, strength requirements are enforced by limiting the axial stress in all members to within $\pm 20\ ksi$ $(137.895\ MPa)$. Second, stability is addressed through adapted Euler buckling limits for slender members as described in Gholizadeh (2013). The critical buckling strength of each element is defined as $4EA/L^2$, where E is the elastic modulus, A is the cross-sectional area, and L is the member length. These constraints collectively ensure that the optimized truss design remains both structurally efficient and stable under the applied loading conditions. Since RFEM6 does not support a customizable buckling strength calculation required for this case study, the necessary structural values are instead computed using a Python script.

It must be noted that by relying only on the cross-sectional area rather than the moment of inertia as in the standard Euler buckling formula $\sigma^2 = \frac{\pi^2 EI}{AL^2}$ to characterize each element's buckling strength, the formulation in the reference paper inherently treats all directions of bending resistance as equivalent. In practical terms, this simplification discards the different stiffness properties that arise from asymmetric profiles. A profile that is, for example, much deeper than it is wide would buckle more easily in the "weak" axis if moment of inertia are accounted for. Therefore, implicit in this choice is the assumption of a symmetric cross section whose second moments are identical about all principal axes even though the specific geometry is not defined in the reference paper. This is another motivation for the choice of the CHS profiles since RFEM6 does not work with unspecified cross-sectional geometry. These assumptions make the problem more theoretical than practical, understanding that directional buckling vulnerabilities have been neglected.

The optimization objective is minimize the weight of the truss via two distinct components as outlined in Gholizadeh (2013). *Size optimization* is performed by varying the cross-sectional properties of the truss members to minimize the overall structural weight. Second, *shape optimization* is introduced by adjusting the x- and y-coordinates of selected nodes (nodes 3, 5, 7, and 9), denoted as $(x_3, y_3), (x_5, y_5), (x_7, y_7), (x_9, y_9)$. Through the combination of size and shape

optimization, the design space is broadened, allowing for more efficient structural configurations while maintaining compliance with the imposed constraints. This can be considered as an extension of the previous case study where only size optimization is performed which shows the potential of the constrained BO to be used for the simultaneous size and shape optimization of structural systems.

Two different approaches for the set-up of the GPs are adopted to represent the structural metrics as functions in the 12-dimensional input design space. In the first approach, 4 GPs are used to model the weight of the structure, maximum compressive stress across all elements, maximum tensile stress across all elements and maximum buckling unity check across all elements.

In the second approach, the number of GPs is increased to 18 to independently model the compressive, tensile stress and buckling unity check for each individual structural member. Noteworthy, is that a member can only experience tension or compression. In the first case a simple unity check for the cross-section under tension is recorded while in the second case the maximum utilization ratio based on the compressive stress and buckling is recorded for each member.

For the first approach, the kernels with their respective hyperparameters are shown in Table 2. Noteworthy is that *Automatic Relevance Determination (ARD)* is used for the Constant, RBF and Matérn kernels as explained in subchapter 3.5.

Table 2: 4 GP approach: Surrogate model kernels for each 12-dimensional function.

Function	Kernels
Weight of the structure	*ConstantKernel(value=1) * Matérn ($\boldsymbol{\ell}$, v=1.5) + ConstantKernel(value=1) * RBF($\boldsymbol{\ell}$), where $\boldsymbol{\ell} \in [0.01,10] \in \boldsymbol{R}^D$, D = 12 dimensions*
Max compressive stress across all elements	*ConstantKernel(value=1) * Matérn ($\boldsymbol{\ell}$, v=1.5) + ConstantKernel(value=1) * RBF($\boldsymbol{\ell}$), where $\boldsymbol{\ell} \in [0.01,10] \in \boldsymbol{R}^D$, D = 12 dimensions*
Max tensile stress across all elements	*ConstantKernel(value=1) * Matérn ($\boldsymbol{\ell}$, v=1.5) + ConstantKernel(value=1) * RBF($\boldsymbol{\ell}$), where $\boldsymbol{\ell} \in [0.01,10] \in \boldsymbol{R}^D$, D = 12 dimensions*
Max buckling unity checks across all elements	*ConstantKernel(value =1) * Matérn ($\boldsymbol{\ell}$, v=2.5) + ConstantKernel(value=1) * RBF($\boldsymbol{\ell}$) , where $\boldsymbol{\ell} \in [0.01,10] \in \boldsymbol{R}^D$, D = 12 dimensions*

For the second approach, the kernels for the weight and the surrogate models for the utilization ratio of each member that yield the best results are shown in Table 3 below. The ARD is used again for the used kernels.

Table 3: 18 GP approach: Surrogate model kernels for each 12-dimensional function.

Function	Kernels
Weight of the structure	*Matérn ($\boldsymbol{\ell}$, v=2.5) + RBF(ℓ = 10), where $\boldsymbol{\ell} \in [0.01,10] \in \boldsymbol{R}^D$, D = 12 dimensions*
Utilization ratio of each element	*Matérn ($\boldsymbol{\ell}$, v=1.5), where $\boldsymbol{\ell} \in [0.01,10] \in \boldsymbol{R}^D$, D = 12 dimensions*

Due to the stochastic nature of the initial samples, the design shown in Figure 49 is added in the initial samples population to prevent the absence of any acceptable designs from stopping the optimization loop. In this way, the optimization gains a valid reference point for the weight based on which subsequent samples can be generated via the acquisition function described in subchapter 3.3.

Another point of attention, in the implementation for the 4GP approach is that the critical element is determined by explicitly iterating over every member, computing its buckling ratio, and tracking the maximum value—while simultaneously recording the elements experiencing the most extreme tensile and compressive stresses. Although straightforward, this approach incurs an *O(N)* computational cost (linear cost) per optimization iteration (where N is the total number of elements), so as more elements are introduced, the time spent in this calculation loop grows in proportion.

The convergence progress for the 4GP & 18GP approach is presented in Figure 52.

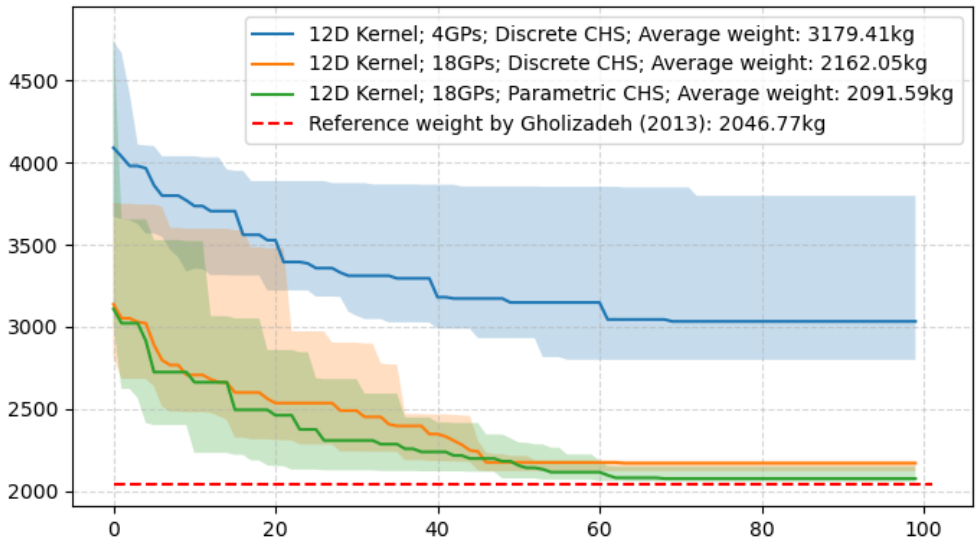


Figure 52: BO progress based on the discrete CHS database and parametric CHS profile. 4GP & 18GP approach. Median objective function values and corresponding 95% confidence interval.

Clearly, increasing the number of GPs from 4 to 18 leads to a substantial improvement in optimization efficiency for the discrete CHS case: the average structural weight decreases from approximately 3179 kg to 2162 kg. This demonstrates the benefit of element-wise surrogate modelling in high-dimensional design spaces, where multiple GPs can better capture the underlying constraint landscapes for each element.

Furthermore, introducing parametric CHS cross-sections further reduces the average weight to 2091 kg, which is much closer to the reference benchmark weight of 2046.77 kg reported by Gholizadeh (2013). This indicates that parametric representations enhance the flexibility of the

optimization process, enabling the algorithm to exploit the continuous nature of the Gaussian process models more effectively compared to purely discrete choices.

Finally, the shaded confidence regions show that parametric CHS not only achieves lower weights but also exhibits narrower variability in the final solutions, highlighting a greater robustness of this formulation. By contrast, discrete CHS with fewer GPs shows both higher mean weights and larger variance, reflecting less reliable convergence behaviour.

An image of the optimal layout and cross-sectional dimensions for the cantilever truss as found by Gholizadeh are reproduced in RFEM6 and are shown in Figure 53. The results of the two approaches for the Bayesian optimization are shown in Figure 54 and Figure 55 using discrete CHS profiles. As can be seen the results deviate largely from the optimal geometry found by Gholizadeh (2013). The result seen in Figure 56 for the parametric CHS with 18GPs shows a good similarity with the reference solution.

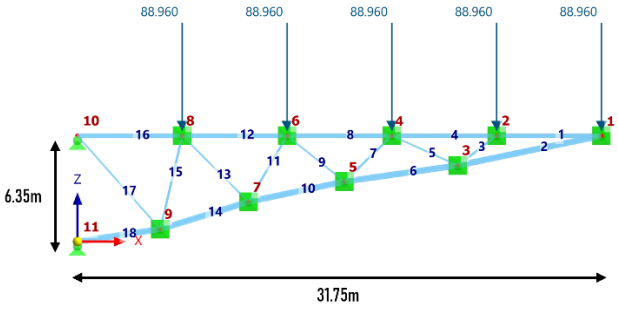


Figure 53: Optimal shape and size truss configuration according to Gholizadeh (2013).

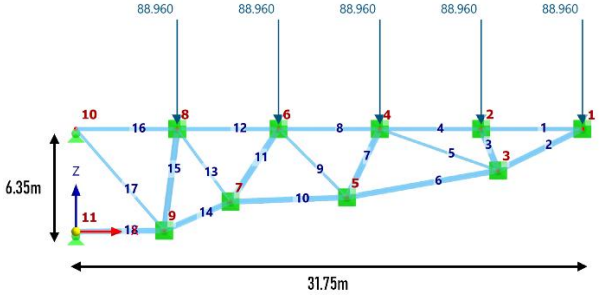


Figure 54: BO optimal shape and size truss configuration: 4GPs approach. Discrete CHS cross-sections.

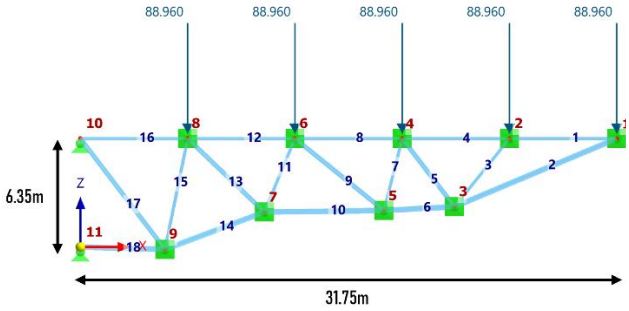


Figure 55: BO optimal shape and size truss configuration: 18GPs approach. Discrete CHS cross-sections.

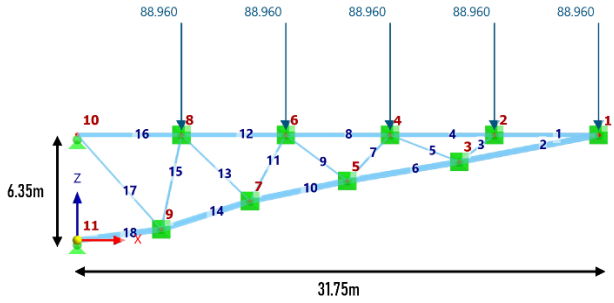


Figure 56: BO optimal shape and size truss configuration: 18GPs approach. Parametric CHS cross-sections.

A summary of the optimized design variables obtained by Gholizadeh (2013) next to the ones produced by the two approaches are presented in Table 4. The best areas for the different truss member groups are converted to the closest areas outlined in the problem definition and the differences are shown in brackets below the areas. The complete list of areas converted from in^2 to mm^2 and the corresponding parametric CHS profiles are given in Table 17 in Appendix C.

Noteworthy, is that the proposed algorithm achieves these updated parameter values with only 250 FEM evaluations (including the 150 initial samples) which is *1/18* of the 4500 FEM evaluations reported by Gholizadeh, thereby demonstrating a dramatic improvement in computational efficiency. This efficiency gain is accompanied by a small increase in total mass, rising from 2046.77kg to 2057kg (0.5% increase) for the 18 GP approach with parametric CHS cross-sections and 16% increase for the discrete CHS cross-sections. This increase is even larger for the 4GPs with discrete CHS cross-sections, namely 37%, demonstrating the efficiency of the element-wise modelling of the constraints in the 18GP approach. This highlights a trade-off between evaluation cost and weight minimization in the two optimization strategies. Moreover, on average the 250 FEM evaluations took 50 minutes and 70 minutes to complete for the 4GPs and 18 GPs approach, respectively. This improvement in computational efficiency when compared to the reference paper is crucial for the dynamic environment in which practicing engineers have to deal with numerous design changes in tight deadlines.

Table 4: Comparison between best results (Gholizadeh, 2013) and two approaches. 18 bar truss structure. 8 shape variables and 4 size variables.

Variables	**(Gholizadeh, 2013)** ***Parametric CHS cross-sections***	**4GPs** ***Discrete CHS cross-sections***	**18GPs** ***Discrete CHS cross-sections***	**18GPs** ***Parametric CHS cross-sections***
A_1 [mm^2]	*8064.50* (CHS 267.3x10)	*7370.0* (CHS 244.5x10)	*5651.10* (CHS 193.7x10)	*5468.98* (CHS 184.6x10)
A_2 [mm^2]	*11290.30* (CHS 369.4x10)	*13500.0* (CHS 355.6x12.5)	*12894.27* CHS 273x16	*12164.52* (CHS 400.2x10)
A_3 [mm^2]	*3709.67* (CHS 128.1x10)	*7920.0* (CHS 406.4x6.3)	*4934.46* (CHS 168.3x10)	*5046.34* (CHS 174.3x10)
A_4 [mm^2]	*2419.35* (CHS 87x10)	*6491.0* (CHS 177.8x12.5)	*2949.08* (CHS 193.7x5)	*3548.05* (CHS 122.9x10)
x_3 [m]	23.04	26.46	22.15	24.60
y_3 [m]	4.57	3.78	2.36	5.33
x_5 [m]	16.17	16.99	17.97	17.97
y_5 [m]	3.60	2.12	2.14	4.19
x_7 [m]	10.36	9.69	10.89	11.65
y_7 [m]	2.39	1.85	2.05	1.83
x_9 [m]	5.05	5.54	5.01	5.47
y_9 [m]	0.75	0.07	-0.14	-0.085
Number of FEM evaluations	4500	250	250	250
Number of violated constraints	0	0	0	0
Best Weight [kg]	2046.77	2797.00	2119.49	2057.00
Worst [kg]	-	3829.56	2186.84	2155.19
Mean [kg]	-	3179.41	2162.05	2091.59
Standard deviation [kg]	-	443.75	29.91	44.15

The two proposed approaches are run 10 times with the same hyperparameters to obtain a distribution of results for the weight of the truss. The best, worst, mean and the standard deviation are presented. The 18GP approach yields a lower best weight, a significantly better worst-case performance, and a lower mean weight when compared to the 4GPs approach. Additionally, the

18GP approach exhibits much lower variability in both cases compared to the 4GPs approach. These results further indicate that the 18GP approach delivers more reliable and consistent solutions. The statistics reported by Gholizadeh (2013) are not included since they are generated using a different optimization algorithm than BO.

Clearly, the algorithm converges to a very similar value for the area of the bottom chords which is critical for the constraints on the compression stresses and buckling to be fulfilled. For the top chords, where tensile stresses are dominant the BO framework has identified a smaller area to be sufficient in accommodating the developed stresses which results in decrease in the weight.

The corresponding structural metrics for the best-found configurations are presented in Table 5. It is evident that the Bayesian optimization algorithm has a bit more material efficient results for the profiles experiencing the maximum tensile stresses and the buckling unity check compared to the values obtained when analysing the structure proposed by Gholizadeh (2013). The stresses calculated based on a first-order static analysis are shown in Figure 127 - Figure 129, respectively in Appendix C: 2D optimization problem: Cantilever Truss. In Table 6 the maximum utilization ratios for every member are shown when the 18GPs approach is used with discrete CHS cross-sections and continuous parametric CHS cross-sections.

Table 5: Structural metrics comparison between the 4GP & 18 GP approach: 8 layout + 4 size variables.

Structural metric	**(Gholizadeh, 2013)** ***Parametric CHS cross-sections***	**4GPs** ***Discrete CHS cross-sections***	**18GPs** ***Discrete CHS cross-sections***	**18GPs** ***Parametric CHS cross-sections***
Max tensile stress [MPa]	137.846	134.547	135.229	137.868
Max compressive stress [MPa]	-119.494	-99.188	-109.163	-118.011
Max Buckling U.C.	0.95	0.82	0.99	0.95

Table 6: Member utilization ratios. 18GPs approach.

Member no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
18 GPs – Discrete CHS cross-sections																		
Util. ratio	0.21	0.37	0.31	0.29	0.29	0.29	0.56	0.48	0.59	0.80	0.89	0.95	0.01	0.91	0.36	0.98	0.88	0.99
18 GPs – Parametric CHS cross-sections																		
Util. ratio	0.43	0.44	0.39	0.54	0.42	0.58	0.81	0.79	0.32	0.72	0.99	0.96	0.08	0.81	0.95	0.99	0.99	0.95

After the Bayesian framework has been validated with the reference paper of Gholizadeh (2013), several modifications are implemented in order to transition from the original theoretical problem formulation to a revised setup which is closer to engineering practice and serves as a building block towards the gridshell cases in subchapters 5.3 & 5.4. The changes are outlined in the following paragraph.

The symmetric circular hollow section (CHS) cross-section dataset used previously is replaced by the asymmetric rectangular hollow section (RHS) dataset to emphasize the difference in buckling strength around the weak and strong axis of the structural profiles which are implemented as design constraints. This type is chosen because the gridshell cases studied in subchapters 5.3 & 5.4 are composed of this type of cross-section. The reason is that in practice this type of closed cross-section is widely used for this type of structures due to their greater resistance to lateral torsional buckling and torsion load cases.

Furthermore, the material properties are updated to reflect those of structural steel S235, increasing the density from 2767.99 kg/m³ to 7850 kg/m³.

Additionally, the structural constraints for the strength and stability are calculated according to EN 1993-1-1 as described in Chapter 4 instead of the adapted theoretical formulas mentioned previously. The calculations are done by RFEM6 rather than a Python script as for the more theoretical version of this case study.

To validate the new setup of the problem, first a simple case where all the 18 elements share the same cross-section is used to compare the BO performance to the analytical solution for a given set of RHS cross-sections. This reduces the dimensionality of the optimization problem to 1D.

To achieve this, the geometry of the 18-bar cantilever truss is now fixed to the coordinates found by Gholizadeh (2013). This allows for a good comparison of the outcomes with the analytical solution for this simplified case (see Figure 57) and allows for plotting the progression of the algorithm as shown from Figure 131 to Figure 150 in Appendix B.

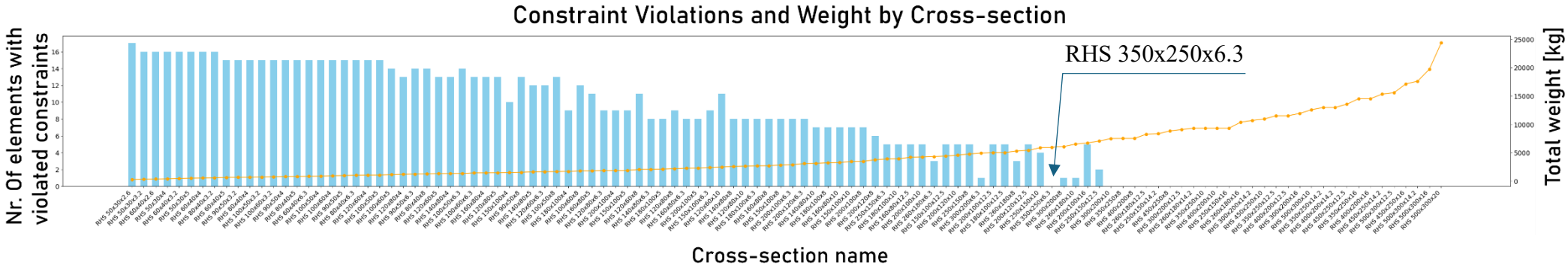


Figure 57: Analytical solution. 1 size variable problem. RHS cross-section dataset.

The final optimal values are shown in Table 7. As expected, the flexural buckling around the weak axis is governing in the design.

Table 7: BO optimal structural metrics for the cantilever truss: 1 size variable. RHS cross-section dataset.

Structural metric	**18GPs – 1D** ***Parametric RHS cross-sections***
Area [mm^2]	7359.0 (RHS 350x250x6.3)
Tensile Stress U.C.	0.644
Compressive Stress U.C.	0.867
Buckling Strong Axis U.C.	0.905
Buckling Weak Axis U.C.	0.936
Weight [kg]	5975.38

It can be seen that RHS 350x250x6.3 is the optimal cross-section with 0 violated constraints and lowest total weight of the truss equal to 5975.38kg which coincides with the result in Figure 57. This outcome suggests that the new setup is implemented correctly.

In the extended formulation of the optimization problem, additional modifications are introduced to increase design flexibility.

The geometry of the truss is no longer held constant, instead, the coordinates of the bottom nodes are allowed to vary, enabling a more representative exploration of feasible configurations. The structural members are divided into the same four distinct groups again, each assigned an independent parametric RHS cross-section, thereby increasing the dimensionality of the design space. The parametric RHS cross-sections are assigned a standard thickness $t = 10mm$ and a ratio of $width/height = 1/2$. This parametrisation is chosen since the only input is the area of the cross-section which allows for solving only one out of the two variables that define the cross-section. This introduces a linear dependency of both buckling capacities as a function of the area. In other words, as the area of the cross-section increases both the buckling capacity around the strong and weak principal axis also increase and vice versa. The ratio of 1:2 is selected as it represents the first rounded integer proportion that characterizes an asymmetric RHS. A ratio of 1:1 would instead correspond to a symmetric square hollow section (SHS), which possesses identical moments of inertia about its principal axes and therefore exhibits structural behaviour comparable to that of a symmetric CHS. This would undermine the intended purpose of the extended formulation of the case study, which is to evaluate the performance of the constrained BO algorithm for a structural system with cross-sections with different buckling strengths in both principal axes of the cross-section. Both parametric values, thickness and ratio, can be changed by the designer.

In addition, parametric CHS cross-sections are used for comparison since their buckling capacity does not depend on the principal axes of the cross-section.

Table 8 summarizes the optimal BO results, considering both parametric CHS and RHS. The weight is noticeably lower (33%) than the single size variable case shown in Table 7, namely 4017.67kg, as is expected when the additional degrees of freedom are introduced which increase the flexibility of the design. RHS designs are more consistent across optimization runs but result in a slightly heavier structure.

Table 8: BO optimal results for the cantilever truss: 8 layout variables and 4 size variables; CHS and RHS parametric cross-sections.

Variables	**18GPs** ***Parametric CHS cross-sections***	**18GPs** ***Parametric RHS cross-sections***
A_1 [mm^2]	*3795.10* (CHS 130.8/10)	*4036.93* (RHS 151x76x10)
A_2 [mm^2]	*6991.03* (CHS 232.5/10)	*8032.59* (RHS 285x142x10)
A_3 [mm^2]	*2482.22* (CHS 89.0/10)	*2815.02* (RHS 111x55x10)
A_4 [mm^2]	*3061.94* (CHS 107.5/10)	*2884.96* (RHS 113x57x10)
x_3 [m]	24.47	24.17
y_3 [m]	5.46	5.40
x_5 [m]	17.43	20.39
y_5 [m]	3.94	4.93
x_7 [m]	11.86	11.23
y_7 [m]	2.22	2.15
x_9 [m]	5.46	5.14
y_9 [m]	-0.46	-0.24
Number of FEM evaluations	250	250
Number of violated constraints	0	0
Best Weight [kg]	3677.54	4017.67
Average weight [kg]	3700.09	4048.54
Standard deviation [kg]	31.90	17.37

As can be seen in Figure 58 and Figure 59 both CHS and RHS design have a very similar geometry to each other despite the fact that the latter has a different buckling capacity in two directions. In Table 9 the utilization ratios for the parametric CHS and RHS cross-sections based on the 18GPs approach are shown.

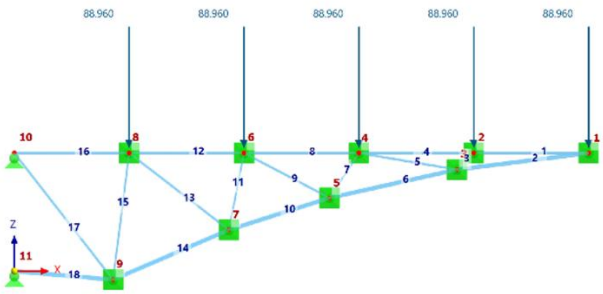


Figure 58: Optimal shape and size truss configuration resulting from BO. Parametric CHS profiles. 18GPs approach.

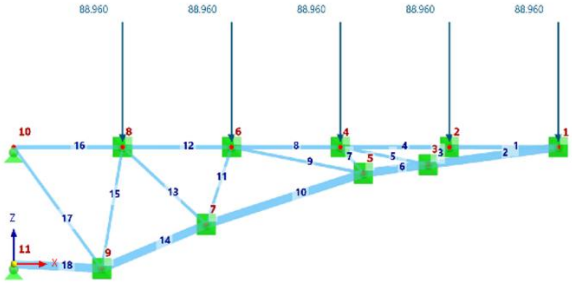


Figure 59: Optimal shape and size truss configuration resulting from BO. Parametric RHS profiles. 18GPs approach.

Table 9: Member utilization ratios. 18GPs approach. CHS and RHS parametric cross-sections.

Member no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Parametric CHS cross-sections																		
Util. ratio	0.82	0.67	0.24	0.92	0.26	0.75	0.31	0.99	0.01	0.70	0.45	1	0.14	0.82	0.97	0.99	0.99	0.99
Parametric RHS cross-sections																		
Util. ratio	0.75	0.59	0.24	0.87	0.36	0.77	0.31	0.99	0.37	0.90	0.31	0.98	0.06	0.78	0.62	0.99	0.95	1

The results of this case study highlight two important aspects for the subsequent gridshell optimization problems First, parametric cross-sections are employed in all subsequent cases to better exploit the capability of Gaussian processes in modelling continuous functions to achieve better and consistent results. Second, the calculation of the structural constraints is performed as outlined in Chapter 4, ensuring that all final designs remain feasible with respect to strength and buckling limitations throughout the optimization process. Together, these elements establish a robust framework for extending the Bayesian Optimization methodology to more complex gridshell structures with 3D geometry analysed in the following subchapters.

5.3. 3D optimization case: Symmetric 4x4 Grid with size variables

In this subchapter a transition is made from the 2D geometry to a 3D structural system. The performance of the BO framework is tested on the symmetric 4x4 grid optimization case with asymmetric boundary conditions which serves as a building block towards the last case study in subchapter 5.4.

5.3.1. Problem Definition & analysis

The structural model is defined as a regular grid configuration consisting of 4 × 4 nodes. The plan dimensions are set to a length and width of 4 m, while the overall height of the structure is 1.43 m with a total of 28 elements. The geometry is fixed throughout the analysis and is illustrated in Figure 60 and Figure 61.

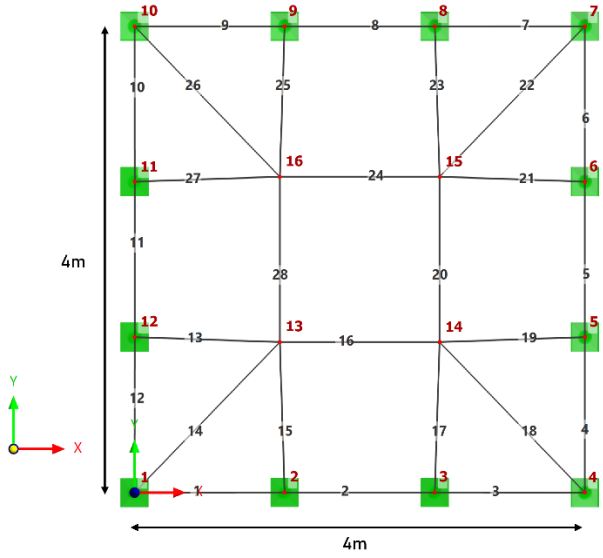


Figure 60: 4x4 gridshell layout. Member and node enumeration.

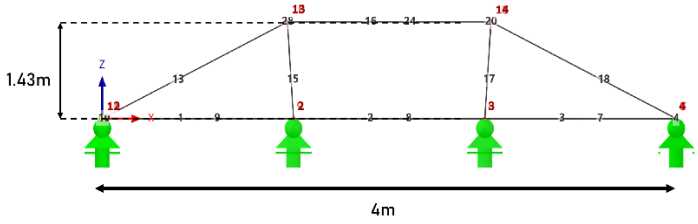


Figure 61: 4x4 gridshell. Side view.

Two load combinations are considered in the analysis:

- Validation load combination: *selft-weight + imposed load of 1 kN/m²*
- Test load combination: *1.35 * selft-weight + 1.35 * 50 * imposed load of 1kN/m²*

This test load is not derived from Eurocode recommendations, but rather represents a fictitious scenario developed empirically to evaluate the effectiveness of the BO framework. The validation and test load combinations are illustrated in Figure 151 and Figure 152 in Appendix D1. 4x4 Gridshell Additional Figures, respectively.

The boundary conditions applied to the structure are illustrated in Figure 62 to Figure 65. In these figures, arrows indicate the directions in which the supports are permitted to move, thereby defining the degrees of freedom at each support location. This representation provides a clear

overview of the structural restraints considered in the analysis. The choice of boundary conditions is adopted to enhance the realism of the gridshell model where the structure is allowed to freely expand and contract due to varying temperature loads.

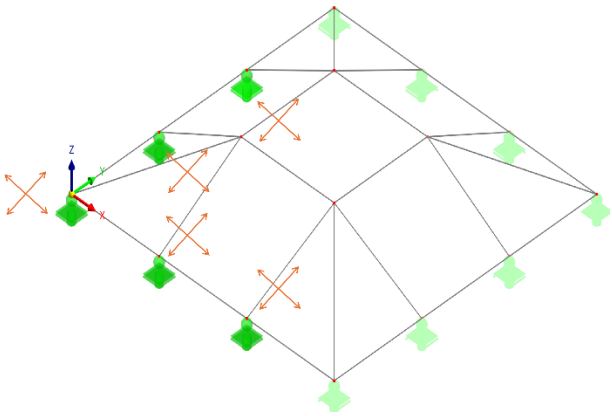


Figure 62: Supports: Translation in x-free, y-free, z-fixed. Rotations in x-,y-,z- directions are free.

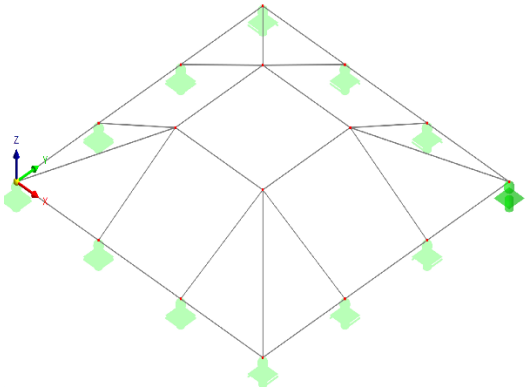


Figure 63: Supports: Translation in x-fixed, y-fixed, z-fixed. Rotations in x-,y-,z- directions are free.

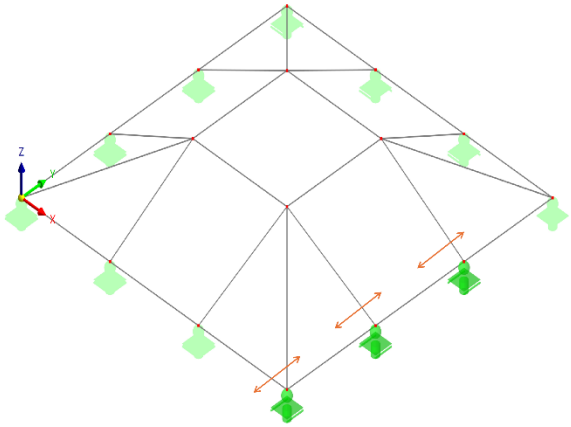


Figure 64: Supports: Translation in x-fixed, y-free, z-fixed. Rotations in x-,y-,z- directions are free.

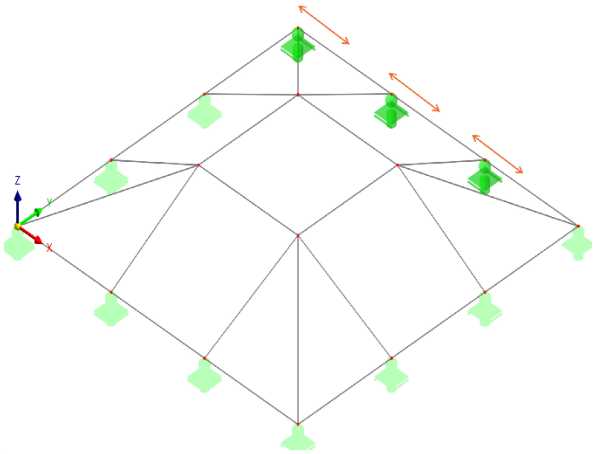


Figure 65: Supports: Translation in x-free, y-fixed, z-fixed. Rotations in x-,y-,z- directions are free.

The structure is modelled using steel grade S235, which is characterized by a yield strength of $f_y = \sigma^{max} = 235N/mm^2$. This material specification is applied to all members of the model.

The structural members are modelled using parametric Rectangular Hollow Sections (RHS). For each element *i*, the axial cross-sectional area is constrained within the bounds $1892mm^2 \leq A_i \leq 21692mm^2$, as illustrated in Figure 66 and Figure 67, respectively. The rationale for selecting these bounds is provided in the following paragraphs.

Lower area bound is kept at 1892 mm^2 when a thickness of t = 10mm is assumed and the inner and outer radius of the cross-section are accounted for by the following two equations according to EN 10210-2 for hot finished cross-sections, respectively: $r_o = 1.5 * t$ and $r_i = 1.0 * t$. Upper area bound is set at 21692mm^2 due to the limit for h_w/t_w for plates that require stiffeners which is set at $\lambda_{lim} = 72 * \sqrt{235/f_y}$ according to EN 1993-1-5:2006; 6.2.6(2); Eq. 6.22. Therefore, the height of the RHS cross-section must not exceed 740 mm when accounting for a thickness of 10mm. Otherwise, a web stiffener must be modelled for the cross-section and shear buckling considered. The same ratio for $width/height = 1/2$ is kept as in the cantilever truss case described in previous subchapter to account for the different buckling capacity depending on the direction. This parameter can be changed by the designer depending on project requirements.

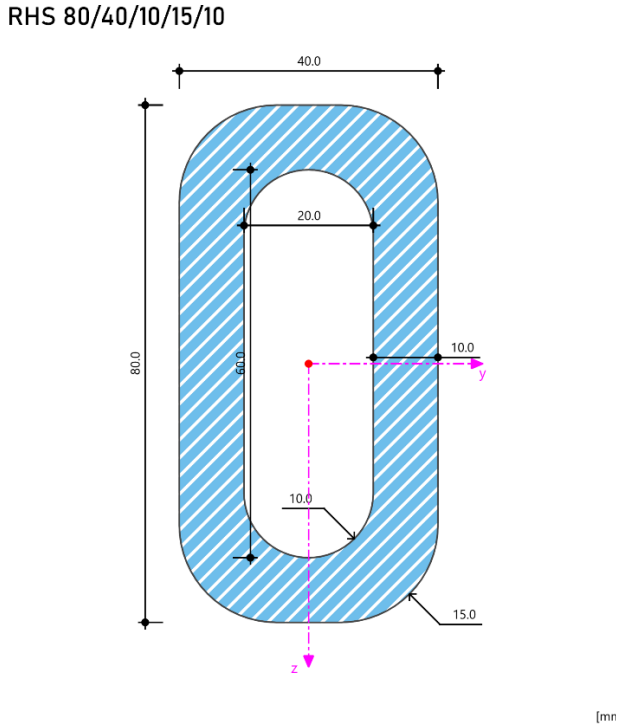


Figure 66: Lower bound for the area of the RHS profiles (A = 1892mm^2).

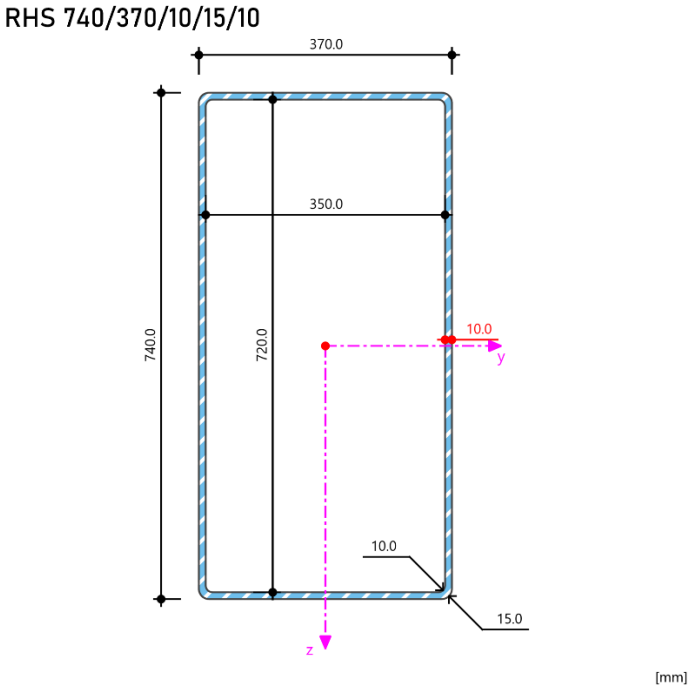


Figure 67: Upper bound for the area of the RHS profiles (A = 21692 mm^2).

The structural constraints are outlined below:

- Strength:
  - Stress constraints are selected to be $\sigma^{max} - |\sigma_i| \geq 0$, where the σ^{max} is the maximum allowable yield stress of the material, and σ_i is the stress in the i^{th} element and are calculated as described in Chapter 4.3.2.
- Stability:
  - The flexural buckling capacity of each beam element is calculated according to the EN 1993-1-1, section 6.3.1 as described in Chapter 4.3.3 in RFEM6. The effective buckling length factor is chosen as 1 (conservatively) for the pin-pin condition since the optimization takes place in the preliminary design phase when the rigidity of the joints is unknown.

The optimization task is formulated as a *size optimization problem*, in which the parametric cross-sections of the structural members are adjusted with the objective of minimizing the overall weight of the structure.

The validation is performed by introducing a load case with low values called "validation load combination" such that the lower bound for the cross-sectional area of the RHS profiles is sufficient to satisfy the constraints. This way the optimal profiles are known prior to optimizing the structure. The results of the BO framework are then compared to this baseline in Figure 70.

Two designs are chosen for the analysis based on a 1 dimensional and 17 dimensional input feature space. In the former (Figure 68) all structural members share the same cross-sectional profile and in the second case (Figure 69) each group of members is assigned an independent cross-section. The highlighted members illustrate the number of degrees of freedom considered in the optimization.

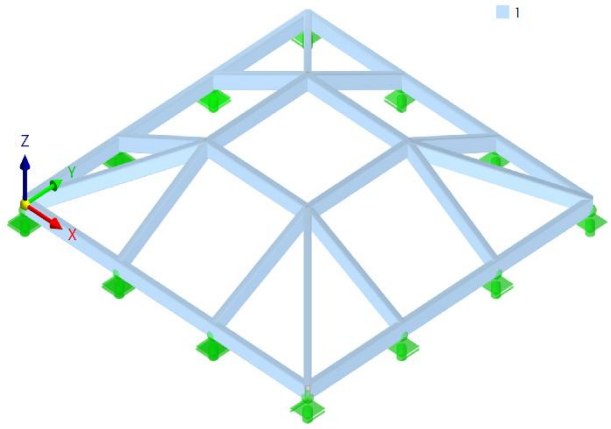


Figure 68: 4x4 Gridshell. 1D design space.

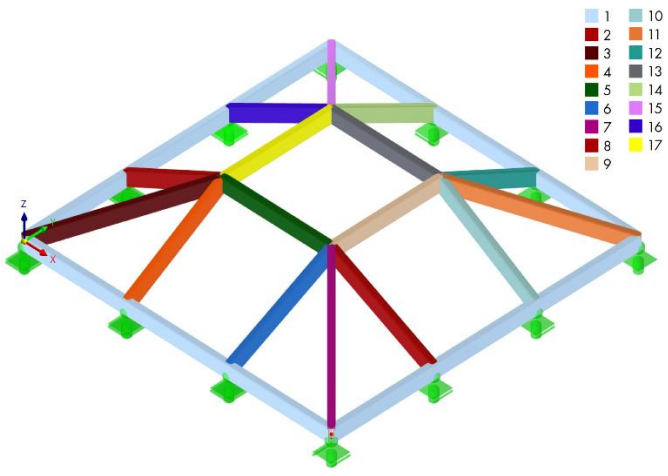


Figure 69: 4x4 Gridshell. 17D design space.

Figure 70 shows the convergence behaviour of the BO algorithm for both cases. The adopted heuristic is 1 point per input dimension resulting in 10 initial samples and 17 initial samples for the 1D case and the 17D, respectively. The 1D case converges to the known analytical optimum of 606.35kg within the first 5 iterations while the 17D case converges around the 10th iteration. For the 17D problem, the optimization converges to a solution with a total weight of 655.38 kg. This corresponds to a difference of approximately 8%, which is deemed acceptable considering the complexity of the high-dimensional search space and the limited number of iterations.

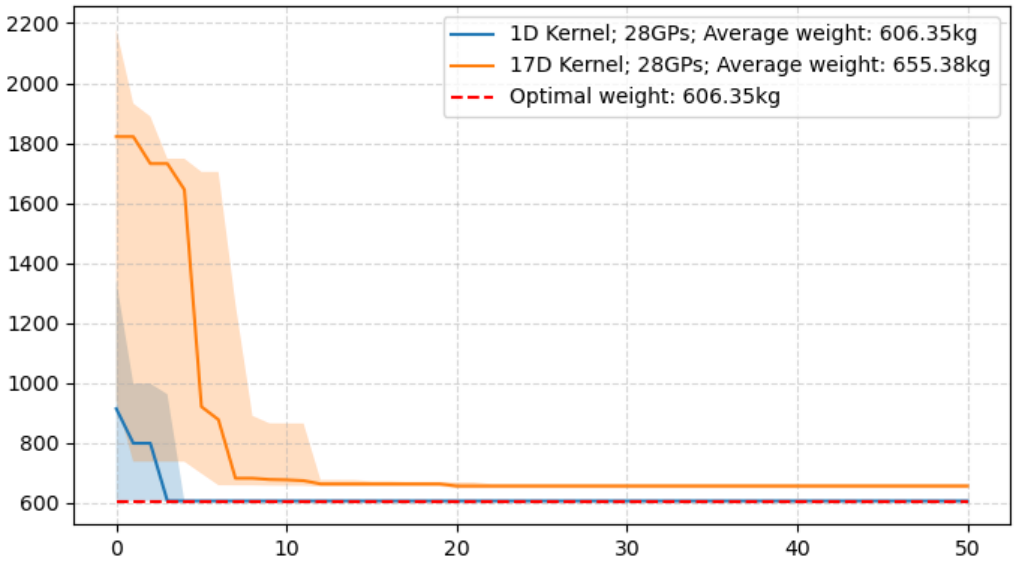


Figure 70: Bayesian optimization progress: 1D & 17D Gridshell model validation. Median objective function values and corresponding 95% confidence interval.

The corresponding optimized structures for 1D case and the 17D case are presented in Figure 71 and Figure 72. The 1D case converges exactly to the analytical solution, confirming the validity of the BO framework in a reduced design domain. In contrast, the 17D case results in slightly oversized members in certain regions, which accounts for the deviation from the theoretical optimum. Nonetheless, the BO approach demonstrates its capability to efficiently navigate a high-dimensional, constrained design space and approximate the global optimum within a small margin of error which can be corrected for with small amount of post-processing.

Overall, these results validate the robustness of the proposed BO framework for structural size optimization in a this 3D 4x4 gridshell case study. The method is able to reproduce analytical solutions in low-dimensional problems while providing near-optimal designs in high-dimensional cases, where traditional methods such as the ones discussed in Chapter 2 may require larger number of iterations to achieve similar results leading to increased computational cost.

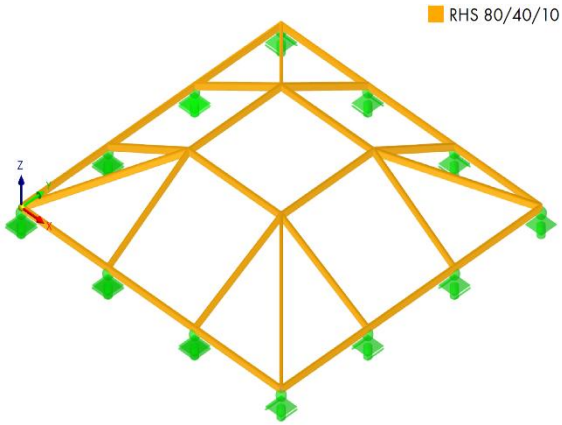


Figure 71: 4x4 Gridshell. Analytical optimal solution & optimal solution in 1D input design space. Unit of the cross-sections is [mm].

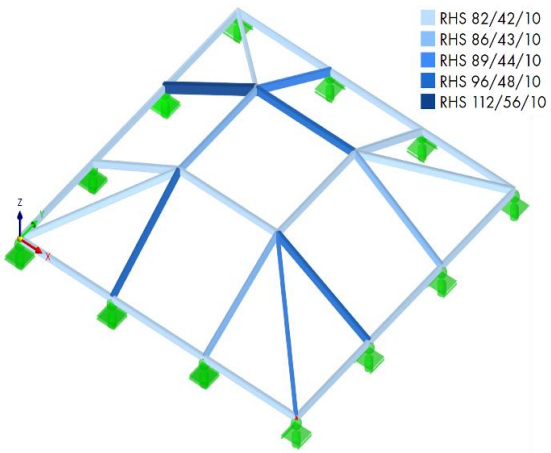


Figure 72: 4x4 Gridshell. Optimal solution in 17D input design space. Unit of the cross-sections is [mm].

Table 10 summarizes the results of the validation study under the simplified load combination. The table reports the optimal cross-sectional areas obtained from three approaches.

For the 1D Bayesian Optimization case, all members converge to the same cross-sectional area of 1892 mm², confirming the correctness of the BO framework when applied to a reduced problem. In contrast, the 17D case yields a distribution of cross-sectional areas across the members, reflecting the increased flexibility of the design space. While some members remain close to the analytical solution, others adopt larger profiles (e.g., A_{15}, A_{19}, A_{27}), leading to an overall higher structural weight. Some of the values are very close to each other such as the $A_{1\text{-}12}$ and A_{13}. The recurring similarity among some of the elements creates a recognizable pattern, as illustrated in Figure 72. This observation coupled with low standard deviation of 4.76 kg for the 17D case indicates the presence of an inherent lower-dimensional structure within the design space, which can be exploited through dimensionality reduction methods such as PCA, as outlined in Chapter 3.8 to reduce the computational cost of the optimization. This analysis is investigated more in depth for the test load combination where the pattern is more explicit.

The summary statistics at the bottom of the table highlight the performance differences between the cases.

Table 10: Validation load combination: Optimal areas and 28 GPs with 17D kernels.

Variables	**Analytical solution** ***Parametric RHS cross-sections***	**28 GPs – 1D** ***Parametric RHS cross-sections***	**28 GPs – 17D** ***Parametric RHS cross-sections***
$A_{1\text{-}12}$ [mm^2]	1892	1892	1919.19
A_{13} [mm^2]	1892	1892	1920.56
A_{14} [mm^2]	1892	1892	1897.17
A_{15} [mm^2]	1892	1892	2198.04
A_{16} [mm^2]	1892	1892	1967.04
A_{17} [mm^2]	1892	1892	2073.86
A_{18} [mm^2]	1892	1892	2160.77
A_{19} [mm^2]	1892	1892	2252.69
A_{20} [mm^2]	1892	1892	1958.73
A_{21} [mm^2]	1892	1892	2044.33
A_{22} [mm^2]	1892	1892	1901.50
A_{23} [mm^2]	1892	1892	1947.98
A_{24} [mm^2]	1892	1892	2368.73
A_{25} [mm^2]	1892	1892	2129.13
A_{26} [mm^2]	1892	1892	1976.27
A_{27} [mm^2]	1892	1892	2854.56
A_{28} [mm^2]	1892	1892	2058.54
Average Number of FEM evaluations	-	12	117
Number of violated constraints	0	0	0
Best Weight [kg]	606.35	606.35	648.29
Worst [kg]	606.35	606.35	661.11
Mean [kg]	606.35	606.35	655.382
Standard deviation [kg]	0	0	4.76
* Note: The corresponding cross-sections are shown in the legend of Figure 71 and Figure 72. The cross-section groups are formed based on the similarity of the area.			

After being evaluated under the validation load case, the BO framework is then tested under the increased load combination which represents a more practical use case for an optimization routine.

The results for the weight optimization of the 4x4 gridshell for the test load combination are shown in Figure 74. It can be seen that when increasing the complexity or the degrees of freedom in the system the weight reduces from the 1D case to the 17D case by 16.7%. The maximum complexity for this problem is achieved by allowing all the 28 elements to have their own cross-section. This means that the analysis is performed in 28-dimensional input design space which is above the empirically derived limit (~20D) of the Bayesian optimization framework as discussed in Chapter 2. The results show that the average optimal weight is increased to 973.61kg making it slightly

better than the 1D optimization results, but worse than the 17D results. Upon further investigation to establish the level of complexity at which the optimal weight starts to deteriorate it is found that this happens at 18D feature space. To clarify the degrees of freedom, the configuration is shown in Figure 73. The average weight of the structure increases by 1% which shows the limitation of the Bayesian optimization framework for this case study which is similar to what is stated in Moriconi et al., (2020). Therefore, the 17D design space is chosen for the rest of the analyses performed in this case study.

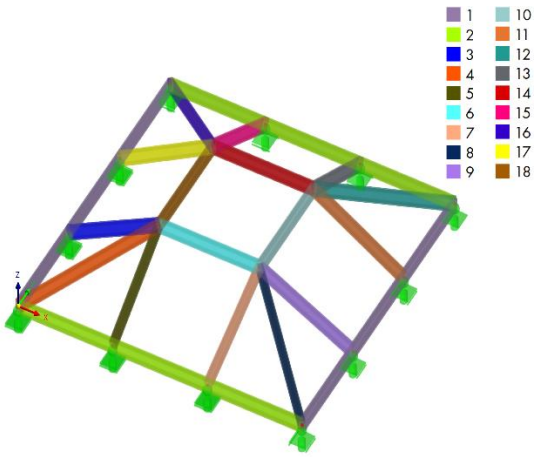


Figure 73: 4x4 Gridshell. 18D design space.

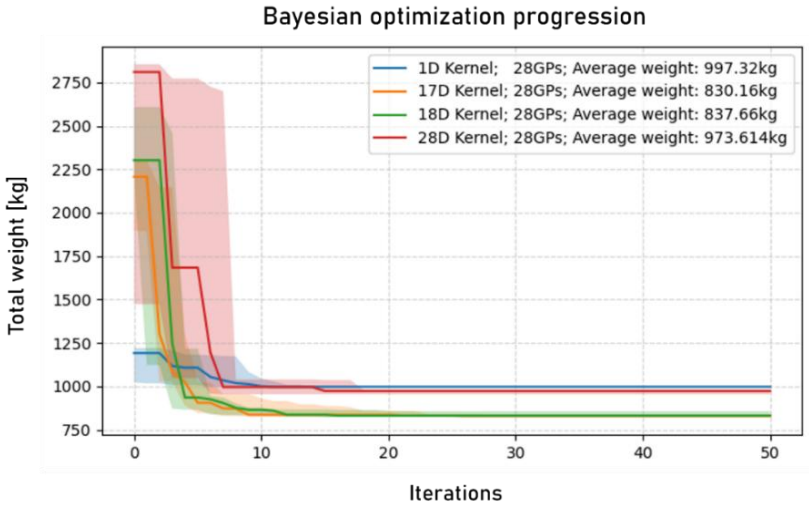


Figure 74: Bayesian optimization progress: 17D Gridshell model validation. Median objective function values and corresponding 95% confidence interval.

Table 11 presents the optimized cross-sectional areas and the corresponding cross-sections obtained under the test load combination for three design space configurations namely .

In the 1D case, all members converge to a uniform cross-section of 3117.99 mm², resulting in a total structural weight of 997.32 kg. This solution satisfies all constraints but is overly conservative, as members are over-dimensioned for the given load combination.

Table 11: Test load combination: Optimal areas for the 1D, 17D and 28D kernels.

Variables	**28 GPs – 1D** ***Parametric RHS cross-sections***	**28 GPs – 17D** ***Parametric RHS cross-sections***	**28 GPs – 28D** ***Parametric RHS cross-sections***
$A_{1\text{-}12}$ [mm^2]	*3117.99* (RHS 121/60/10)	*2762.63* (RHS 109/54/10)	shown below*
A_{13} [mm^2]	*3117.99* (RHS 121/60/10)	*2057.88* (RHS 86/43/10)	*1937.66* (RHS 91/45/10)
A_{14} [mm^2]	*3117.99* (RHS 121/60/10)	*3424.22* (RHS 131/66/10)	*3980.54* (RHS 178/89/10)
A_{15} [mm^2]	*3117.99* (RHS 121/60/10)	*2145.76* (RHS 88/44/10)	*2109.78* (RHS 118/59/10)
A_{16} [mm^2]	*3117.99* (RHS 121/60/10)	*3091.81* (RHS 120/60/10)	*4461.66* (RHS 115/58/10/)
A_{17} [mm^2]	*3117.99* (RHS 121/60/10)	*1967.23* (RHS 82/41/10)	*2868.86* (RHS 95/48/10)
A_{18} [mm^2]	*3117.99* (RHS 121/60/10)	*2557.73* (RHS 102/51/10)	*2465.08* (RHS 103/52/10)
A_{19} [mm^2]	*3117.99* (RHS 121/60/10)	*2298.61* (RHS 94/47/10)	*2947.79* (RHS 143/71/10)
A_{20} [mm^2]	*3117.99* (RHS 121/60/10)	*2508.59* (RHS 101/50/10)	*2697.70* (RHS 113/56/10)
A_{21} [mm^2]	*3117.99* (RHS 121/60/10)	*2497.58* (RHS 100/50/10)	*3878.22* (RHS 100/50/10)
A_{22} [mm^2]	*3117.99* (RHS 121/60/10)	*2294.89* (RHS 93/47/10)	*2900.36* (RHS 118/59/10)
A_{23} [mm^2]	*3117.99* (RHS 121/60/10)	*2249.69* (RHS 92/46/10)	*4658.60* (RHS 103/51/10)
A_{24} [mm^2]	*3117.99* (RHS 121/60/10)	*2109.90* (RHS 87/44/10)	*2191.26* (RHS 126/63/10)
A_{25} [mm^2]	*3117.99* (RHS 121/60/10)	*2297.62* (RHS 93/47/10)	*2171.62* (RHS 109/54/10)
A_{26} [mm^2]	*3117.99* (RHS 121/60/10)	*2477.11* (RHS 99/50/10)	*2456.31* (RHS 117/58/10)
A_{27} [mm^2]	*3117.99* (RHS 121/60/10)	*1995.09* (RHS 83/42/10)	*2603.96* (RHS 85/43/10/)
A_{28} [mm^2]	*3117.99* (RHS 121/60/10)	*3068.87* (RHS 119/60/10)	*3246.99* (RHS 134/67/10)
Average Number of FEM evaluations	10	210	300
Number of violated constraints	0	0	0
Best Weight [kg]	997.32	824.82	953.67
Worst [kg]	997.32	839.93	986.10
Mean [kg]	997.32	830.16	973.62
Standard deviation [kg]	0	4.77	12.79
* The areas for the first 12 elements in the 28D case are the following: $A_{1\text{-}12} \in$[4182.40; 3315.41; 3524.95; 2748.51; 3103.82; 2014.73; 4388.95; 2349.72; 2097.41; 2747.40; 3308.21; 3640.72] with corresponding cross-sections [RHS 138/69/10, RHS 146/73/10, RHS 130/65/10, RHS 108/54/10, RHS 96/48/10, RHS 90/45/10, RHS 100/50/10, RHS 89/44/10, RHS 98/49/10, RHS 159/80/10, RHS 93/46/10, RHS 139/70/10]			

The influence of the number of initial samples for the 17D kernels with the test load combination, is shown in Figure 75. As a reference a random search is also performed to showcase the efficiency of the Bayesian optimization framework. The adopted heuristic is 1 point and 10 points per input dimension resulting in 17 and 170 initial samples, respectively. It can be concluded that the latter produces better results in terms average weight and reduced variance between runs. However, this comes at a higher computational cost due to the increased number of data points N and the $O(N^3)$ scalability of the Gaussian processes. The difference in the final average weight between the two cases is 3.6%. Based on this sensitivity analysis, the balance between what number of initial samples to use and the computational cost is left to the user.

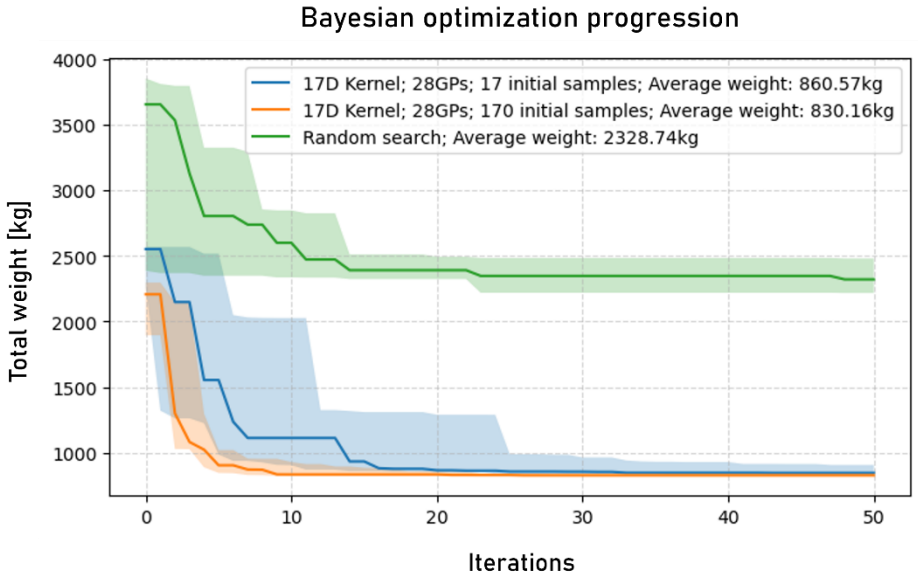


Figure 75: Bayesian optimization progress: 17D GPs & Random Search. Median objective function values and corresponding 95% confidence interval.

Further details on the influence of the number of initial samples on the hyperparameters of the Gaussian processes for 17D and 28D input space are shown in the heatmaps in Figure 152 to Figure 155 in Appendix D1.

The design shown in Figure 76 is obtained from a common rule of thumb of $(1/25) * span$ for the height of the cross-section of all of the elements in the gridshell. It provides a reasonable design for comparison with the optimized design obtained from the BO framework in the 17D design space shown in Figure 77.

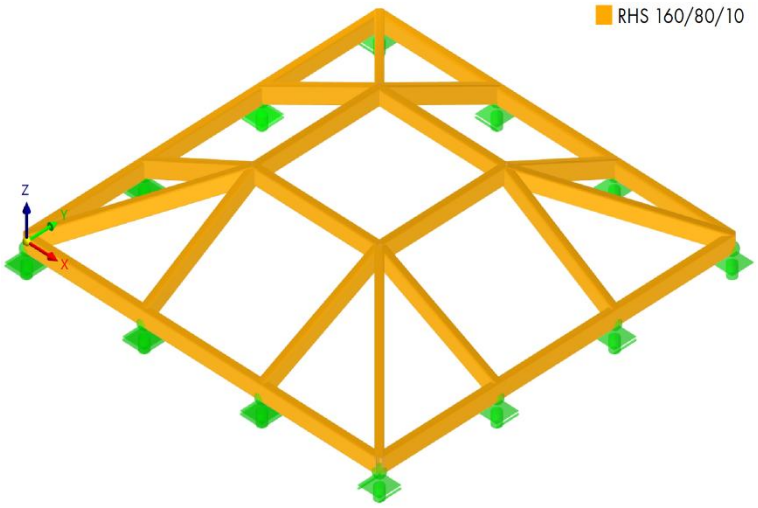


Figure 76: 4x4 Gridshell. Initial rule of thumb design. Span is 4m. Unit of the cross-sections is [mm].

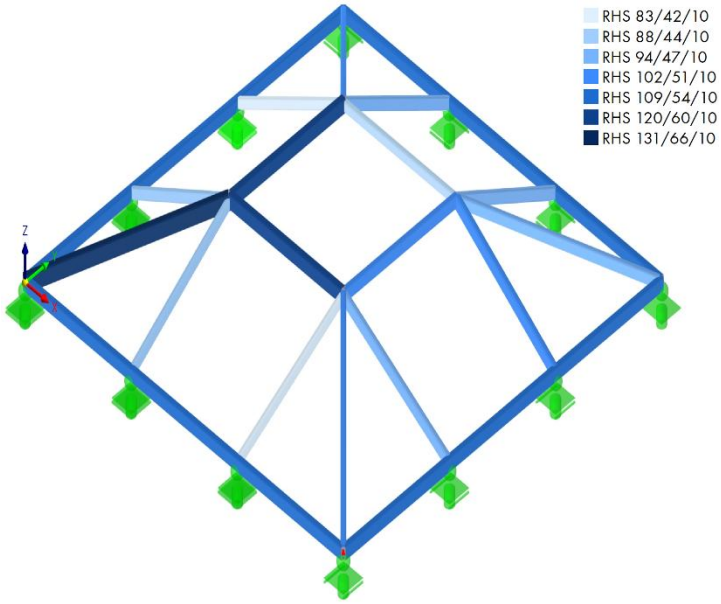


Figure 77: 4x4 Gridshell. Best optimal solution in 17D input design space.
Unit of the cross-sections is [mm].

The total structural weight of the first design is 1376 kg while the optimized one has a weight of 824.82 kg. This results in a *1.67× reduction* of steel material which reduces the cost and makes the gridshell more sustainable by reducing the CO_2 emissions for the production of the profiles.

As illustrated in Figure 77 and outlined in Table 11, the resulting design exhibits similarities between the areas of members and forms, suggesting the presence of an underlying low-dimensional structure. In practice, this means that the effective dimensionality of the problem is smaller than the nominal 17 design variables, since certain cross-sections evolve in a correlated manner to accommodate the load transfer within the gridshell.

Principal Component Analysis (PCA) as discussed in subchapter 3.8 offers a systematic means to exploit this redundancy by identifying the most influential directions of variation in the design space. By projecting the original 17D problem onto a reduced set of uncorrelated principal components, it becomes possible to retain the dominant structural patterns while eliminating redundant or weakly contributing variables. This dimensionality reduction can mitigate some of the computational burden of Bayesian Optimization while producing close to the found optimal designs in the original design space. As mentioned before 17D is chosen as the original complexity on which the PCA is applied.

The initial samples that are generated with the Latin Hypercube sampling strategy in the design domain for the areas of the cross-sections follow the pattern shown in Figure 77. This approach spreads samples across the entire input space in a space-filling manner on which the PCA is later fitted on. This approach can be regarded as an analogue to constructing multiple 4×4 gridshells produced by different manufacturers under comparable loading and boundary conditions, with the aim of identifying underlying patterns in their structural performance.

Table 12 present the results of applying Principal Component Analysis (PCA) to reduce the dimensionality of the 17D gridshell design space prior to Bayesian Optimization. Two reduced representations are considered, using 6 principal components (PCs) and 10 PCs, respectively. The rationale behind this choice is discussed in the following paragraph.

The explained variance plot in Figure 78 shows that the first few principal components capture most of the variability in the design space which is expected when there is a relatively clear pattern in the data. Specifically, the first 6 components account for approximately 98% of the variance as shown in cumulative variance plot Figure 79, while 10 components are sufficient to capture nearly the entire variance. This indicates that the effective dimensionality of the design problem is substantially lower than the original 17 variables, which is a similar result to the 8 effective cross-sections shown in Figure 77.

Table 12: 4x4 Gridshell: Statistics for the PCA with different number of principle components.

PCA number of components	**6 principle components**	**10 principle components**
Best Weight [kg]	1005.77	892.70
Worst Weight [kg]	1182.45	975.66
Mean [kg]	1062.23	928.42
Standard deviation [kg]	69.47	22.97

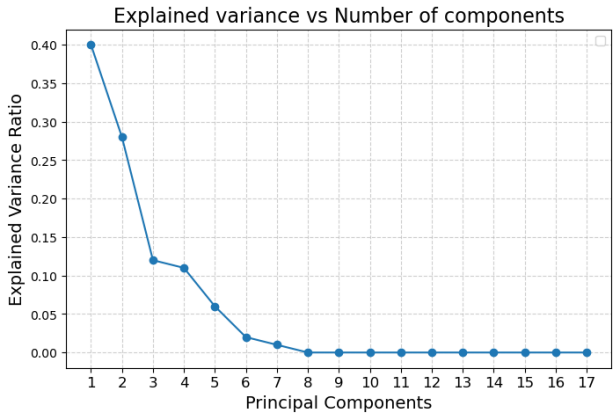


Figure 78: 4x4 Gridshell: Explained variance by each principle component.

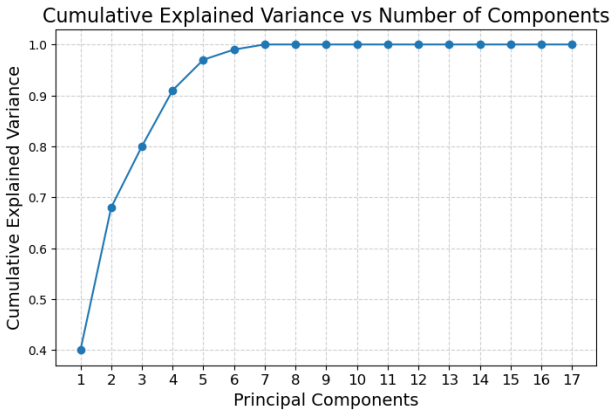


Figure 79: 4x4 Gridshell: Cumulative explained variance by each principle component.

The optimization results summarized in Table 16 reveal a clear trade-off between accuracy and computational efficiency. With 6 components, the best structural weight obtained is 1005.77 kg, with a mean weight of 1062.23 kg and relatively large standard deviation of 69.47 kg. Increasing the dimensionality to 10 components improves both accuracy and robustness: the best solution is reduced to 892.70 kg, with a mean of 928.42 kg and a lower standard deviation of 22.97 kg. However, both PCA cases remain above the best weight obtained without PCA of 824.82 kg, indicating that dimensionality reduction introduces a small optimality gap.

The convergence histories in Figure 80 further illustrate these effects. The shaded regions indicate the variance across the 10 runs, which is visibly larger for 6 components, highlighting reduced reliability when fewer components are retained in this case study.

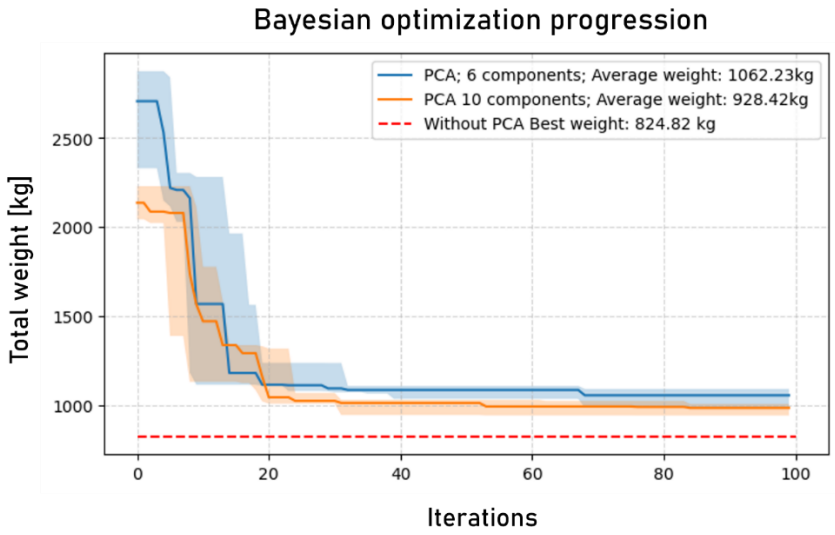


Figure 80: Bayesian optimization progress with PCA for 4x4 gridshell. Median objective function values and corresponding 95% confidence interval for the different number of principal components. Test load combination.

Finally, Figure 81 reports the average convergence time. The use of PCA significantly reduces computational effort: optimization with 6 components converges in approximately 20 minutes, with 10 components requiring around 50 minutes, compared to more than 100 minutes for the full 17D dimensional problem. This is due to the fact that both the maximisation of the acquisition function and the fitting of the Gaussian processes happens in a reduced design space leading to a faster execution of the algorithm per iteration..

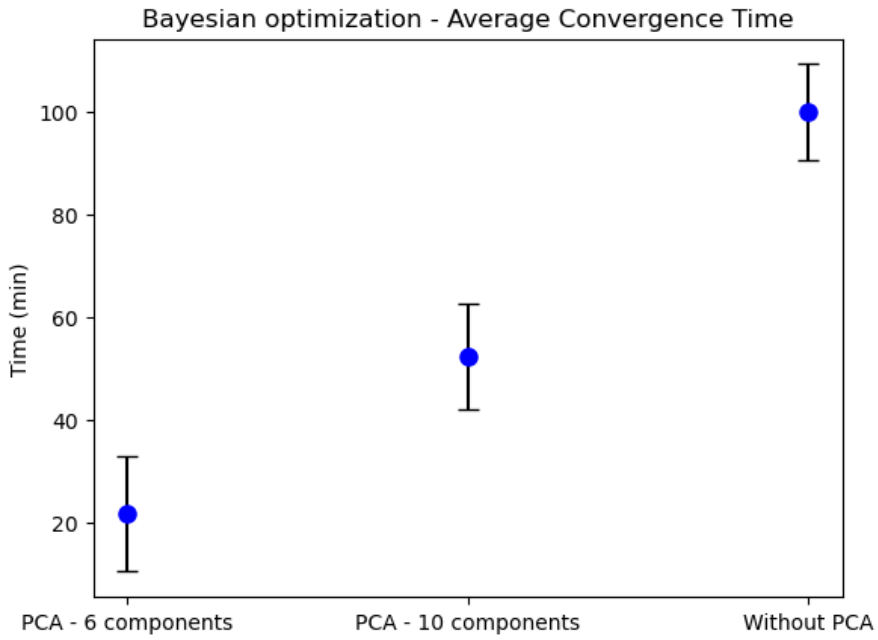


Figure 81: Average convergence time comparison between the runs with PCA and without PCA for the 4x4 gridshell.

Therefore, PCA provides a compromise between solution quality and computational efficiency. While the full 17D optimization yields the best structural weight, PCA with 10 components achieves a near-optimal solution at less than half the computational cost. The 6-component case is computationally more efficient but suffers from reduced accuracy and higher variance, indicating that too much dimensionality reduction may oversimplify the design space.

5.4. 3D optimization case: Symmetric 9x9 Grid with size variables

In this subchapter the last case study of a 3D symmetric 9×9 grid optimization case with symmetric boundary conditions and 288 steel elements is analysed to evaluate the performance of the BO framework. The same setup of the BO framework as for the 4x4 gridshell is used. The case study is inspired by the C30 gridshell designed and built by Octatube which serves to create the boundary conditions, set the load combination, focus on a particular type of cross-section and determine the material properties. This case is selected to assess the BO framework on a practical design implemented in reality.

For context, a short description about the design and construction of the C30 shell is provided in the following paragraphs. Afterwards, the optimization problem is defined and results are evaluated.

5.4.1. C30 Gridshell

The aim of constructing this structure was to cover the inner courtyard of an office building which had a monumental character due its resemblance to Dutch architecture of the 16th/17th century even though the foundations were laid in the 1916 (Octatube, 2020). As a result of this status, some design and building challenges emerged. An example was that no horizontal forces were allowed to be transferred perpendicular to the façade. Also, in Figure 82 it can be seen that there are polygonal towers in three of the edges of the enclosed space which makes the construction of continuous straight edge beams unfeasible and in turn causes the loss of stiffness of the structure. To overcome this challenge, pretension cables were used to keep the edges together (see Figure 83). This must be carried out with millimetre-level precision, as even slight deviations can lead to a completely different distribution of forces.

Figure 82: Overview of the C30 structure.
(*Source: Octatube, 2020*)

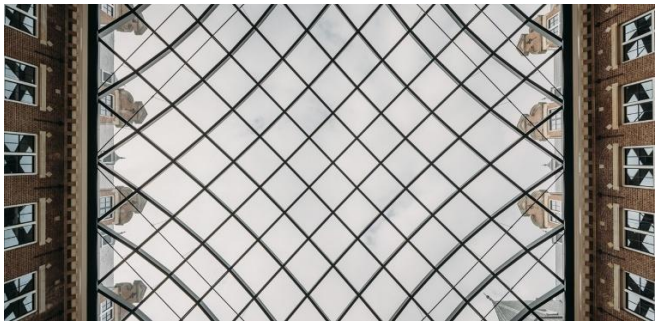


Figure 83: C30 view from below the roof.
Pretension cables in the corner.
(*Source: Octatube, 2020*)

Another challenge during the construction of the grid shell was how to close the middle part of the roof. To do this, the engineers developed an assembly technique based on the principle of reciprocal frames which is a type of self-supporting structure. There are four ladder frames (orange elements in Figure 84) in the middle, that do not rely on any scaffolding and in the end close the

roof. Cranes were used to lift the frames into position (Figure 85). To achieve this, advanced prefabrication techniques were used such as file-to-factory which sends data to a laser cutter (instead of sawing) through a script in order to produce the profiles and connections.

Lastly, there was a challenge in determining the connection stiffness between the different elements which ultimately influences how the stresses are distributed over the roof. To tackle this the engineers developed two separate models that represented the lower and upper limit of the connection stiffnesses. In the former a lot of deformations were found whereas in the latter the deformations are more controlled.

Because of the structure's shape and the varying sizes and angles of the steel elements, a parametric design approach was employed in order to design the shell efficiently.

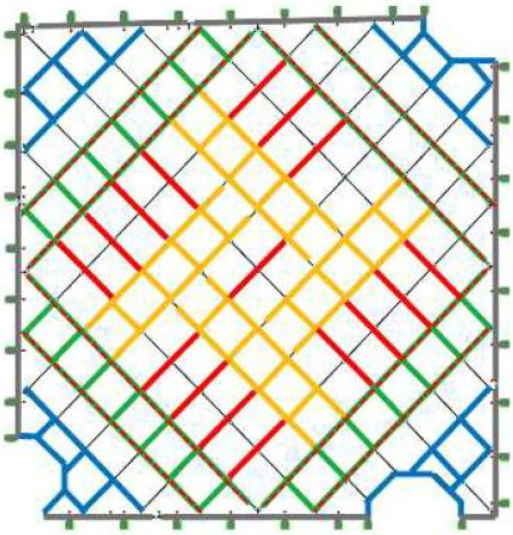


Figure 84: Groups of prefabricated elements in the C30 shell.

Figure 85: Placing of the ladder frames using cranes. (*Source: Octatube, 2020*)

5.4.2. Problem definition & analysis

The structural configuration is defined as a diagonal grid consisting of 9×9 nodes. The plan dimensions are set to a length and width of 28 m, with a height of 3.84 m with a total of 288 elements. The geometry remains fixed throughout the analysis and is illustrated in Figure 86 and Figure 87. The geometry is form-found using the Grasshopper script presented in Figure 163 (Appendix D2. 9x9 Gridshell Additional Figures). This script employs the dynamic relaxation method via the Kangaroo 2 plugin, however, a detailed explanation of the method lies beyond the scope of this thesis. For further information, the reader is referred to *Shell Structures for Architecture: Form Finding and Optimization* by Adriaenssens et al. (2014). The geometry is simplified to a square layout than the original C30 design to make the problem symmetric.

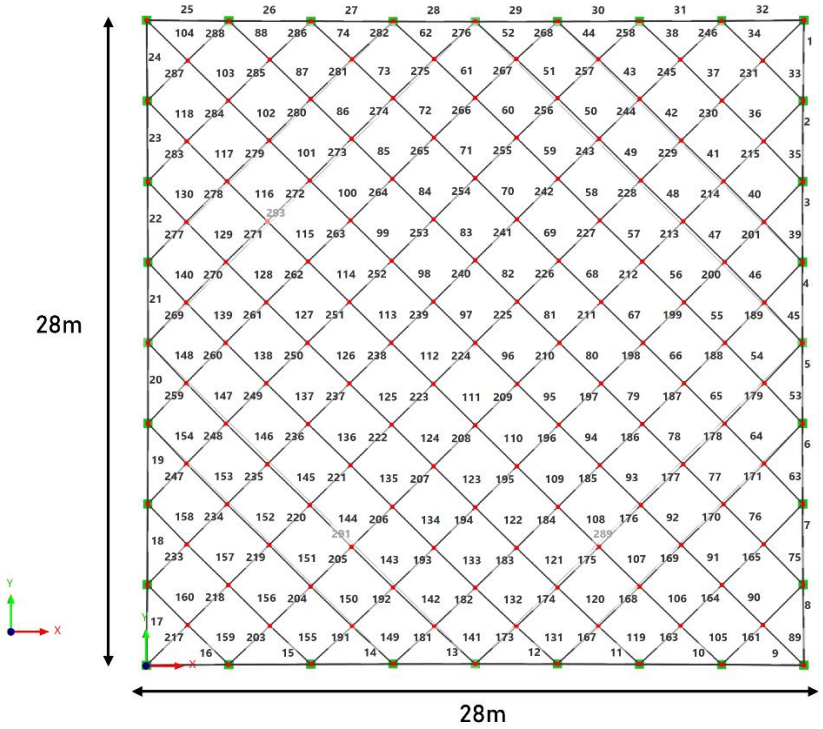


Figure 86: 9x9 gridshell. Layout with member numbers.

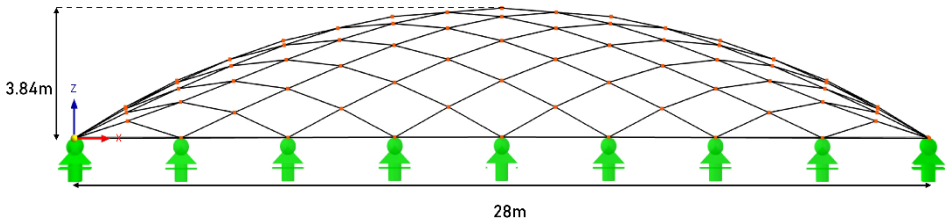


Figure 87: 9x9 gridshell. Side view.

The test load combination is defined at the ultimate limit state (ULS) as $1.20 * self - weight + 1.20 * Glass + 1.50 * Wind$. This formulation has been adapted from the original structural report for the C30 gridshell. The load cases and resulting load combination are shown in Figure 156 to Figure 159 in Appendix D2.

The support conditions of the structure are illustrated in Figure 88 to Figure 90. In these representations, arrows indicate the directions in which the supports are free to move, thereby defining the degrees of freedom at each support location. The boundary conditions are inspired by the C30 gridshell where the structure is allowed to move to accommodate any thermal loads that might occur. There are also additional measures taken to prevent any horizontal forces developing on the façade of the historic buildings, but they are addressed later in the text.

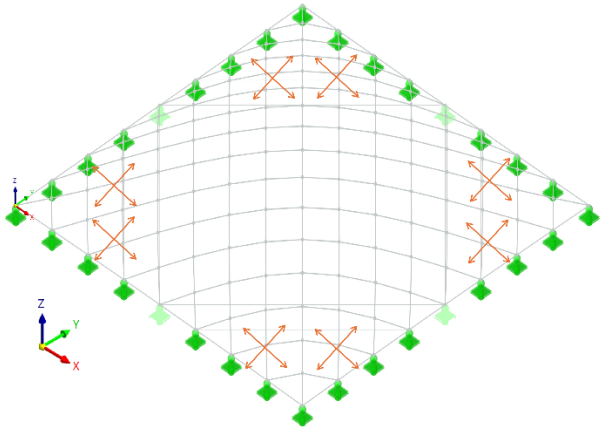


Figure 88: Supports: Translation in x-free, y-free, z-fixed. Rotations in x-,y-,z- directions are free.

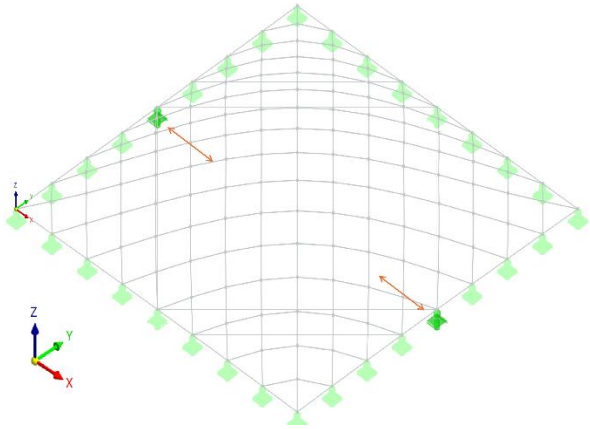


Figure 89: Supports: Translation in x-free, y-fixed, z-fixed. Rotations in x-,y-,z- are free.

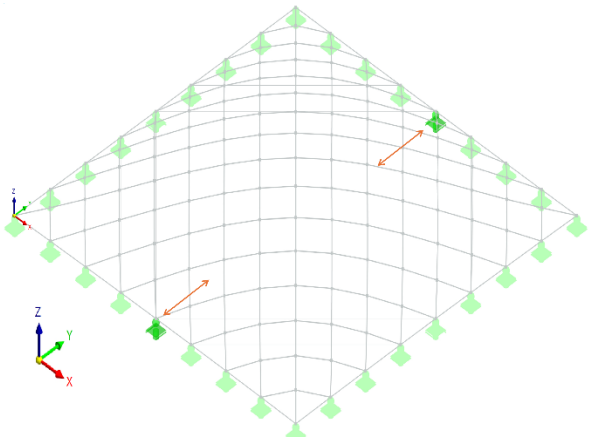


Figure 90: Supports: Translation in x-fixed, y-free, z-fixed. Rotations in x-,y-,z- are free.

The structure is modelled using steel grade S355, characterized by a yield strength of $f_y = \sigma^{max} = 355\ N/mm^2$. This material specification is applied to all structural members similar to the C30 gridshell.

The members are modelled using parametric Rectangular Hollow Sections (RHS). For each element i, the axial cross-sectional area is constrained within the bounds $1723 \leq A_i \leq 10010mm^2$, as illustrated in Figure 91 and Figure 92. These values are derived from setting the thickness of the cross-section $t = 8mm$ and the ratio of $width/height = 1/3$. The choice for these parametric values is made based on the structural report of the C30 gridshell.

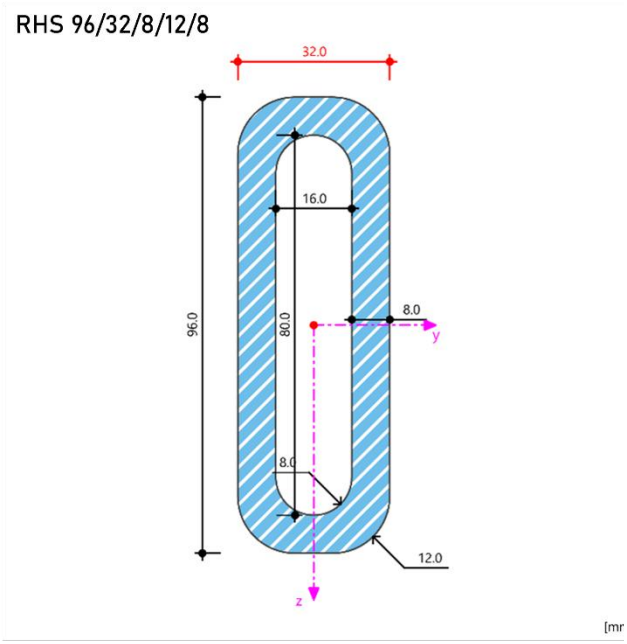


Figure 91: Lower bound for the area of the RHS profiles (A = 1723mm2).

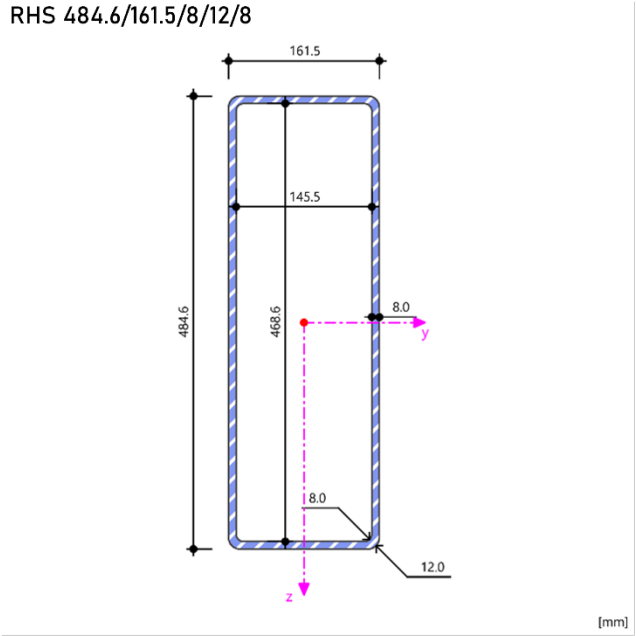


Figure 92: Upper bound for the area of the RHS profiles (A = 10010mm2).

The constraints are outlined below:

- Strength:
  - Stress constraints are selected to be $\sigma^{max} - |\sigma_i| \geq 0$, where the σ^{max} is the maximum allowable yield stress of the material, and σ_i is the stress in the i^{th} element and are calculated as described in Chapter 4.3.2.
- Stability:
  - Local buckling strength of each beam element is calculated according to the EN 1993-1-1, section 6.3.1 as described in Chapter 4.3.3. The effective buckling length factor is chosen as 1 (conservatively) for the pin-pin condition since the

optimization takes place in the preliminary design phase when the rigidity of the joints is unknown.

Based on the results of the previous case study of the 4x4 gridshell, the analysis of the full dimensionality of the 9x9 gridshell (288D) is prohibitively expensive to evaluate with the BO framework.

Therefore, in this case study, two lines of symmetry are introduced in order to reduce the dimensionality of the optimization problem to be below 20D. These symmetry conditions simplify the dimensionality of the optimization problem while preserving the structural behaviour of the system. The applied symmetry are illustrated in Figure 93.

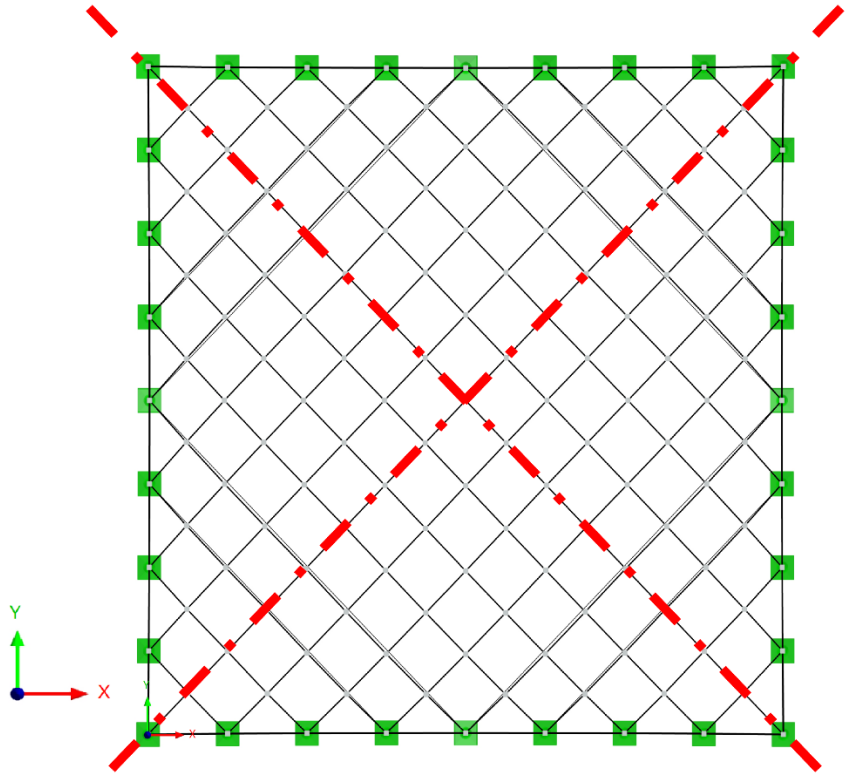


Figure 93: 9x9 gridshell: Lines of symmetry.

The optimization task is formulated as a *size optimization problem*, in which the parametric cross-sections of the structural members are adjusted with the objective of minimizing the overall weight of the structure.

The design shown in Figure 94 illustrates the 17 dimensional input feature space. Each colour represents the cross-section assigned to a group of steel elements. This is different from the 4x4 gridshell case where each individual element has its own cross-section. The edge beams (nr. 1-32) share the same cross-section. Furthermore, 8 rod elements with solid circular cross-section with diameter equal to 50mm are modelled as shown. They can only accommodate tension forces. This addition is made based on the original C30 gridshell to prevent any horizontal forces developing on the façade of the historic buildings that support the structure. These 8 elements are excluded from the optimization problem keeping the task 17D.

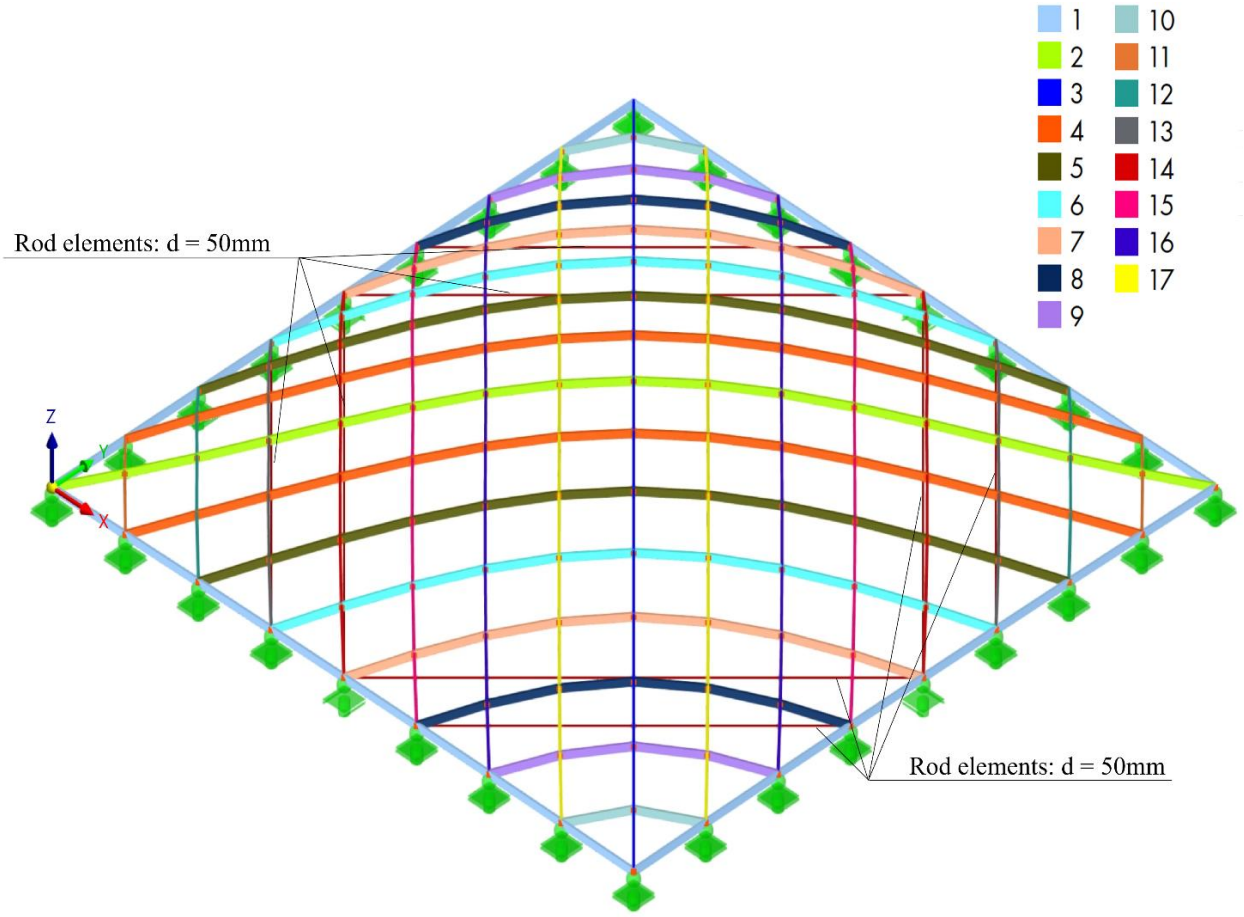


Figure 94: 9x9 Gridshell. 17D design space.

The convergence history of the constrained BO framework compared against random search is shown in Figure 95. Two BO runs are shown - one with 17 initial samples and another with 170 initial samples, both employing a 17-dimensional kernel with 288 Gaussian Process surrogate models to handle the strength and stability constraints discussed in Chapter 4.3. The shaded regions illustrate the variability across independent runs, while the solid lines indicate the median performance.

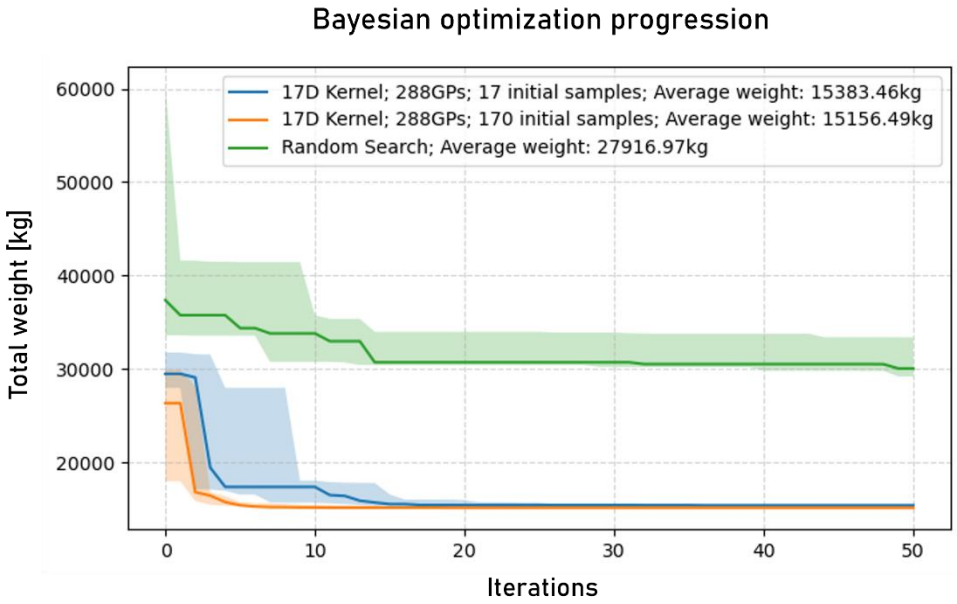


Figure 95: Bayesian optimization progress: 17D 9x9 Gridshell model. Median objective function values and corresponding 95% confidence interval.

It can be observed that both BO configurations significantly outperform random search. The BO runs rapidly reduce the structural weight within the first 10–15 iterations, converging toward feasible lightweight solutions. Increasing the number of initial samples improves convergence stability and leads to a slightly lighter final design (15,156.49 kg compared to 15,383.46 kg on average). In contrast, random search converges slowly and stagnates at higher weight levels with higher variability, highlighting the efficiency of the proposed surrogate-assisted constrained optimization approach.

In the following figures the effect of the optimization process on the gridshell structure is visualized. Figure 96 shows the initial design, based on the uniform C30 gridshell cross-section RHS 300/100/8 for all elements, with a total structural weight of 38.389 tons. The max utilization ratio is 0.30 which leads to excessive material usage.

In contrast, Figure 98 displays the optimized configuration obtained through Bayesian Optimization. The allocation reduces material consumption significantly, yielding a total structural weight of 15.098 tons which corresponds to a *2.54* × lighter structure or a weight reduction of approximately 60%.

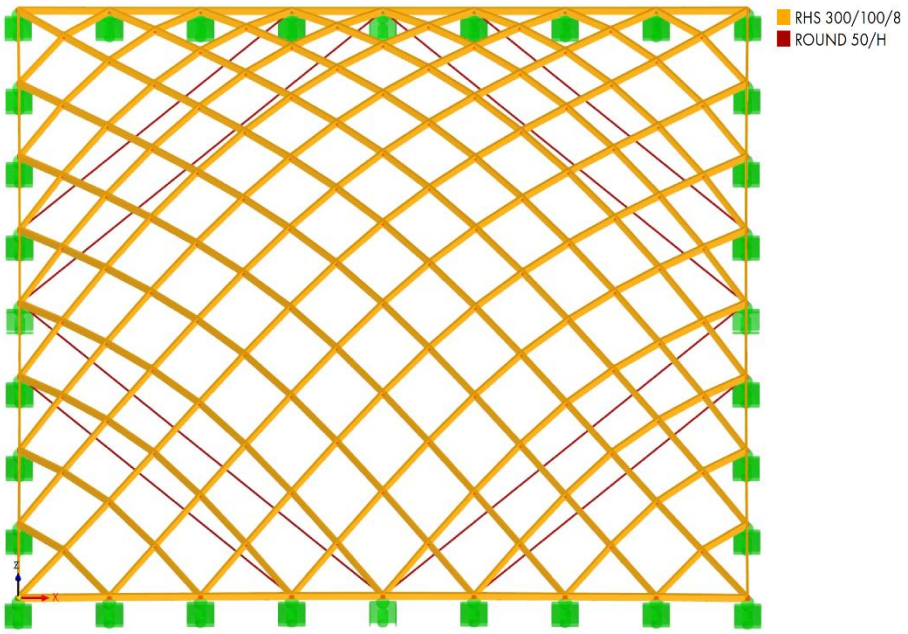


Figure 96: 9x9 Gridshell. Initial design based on the C30 gridshell cross-sections.
Unit of the cross-sections is [mm].

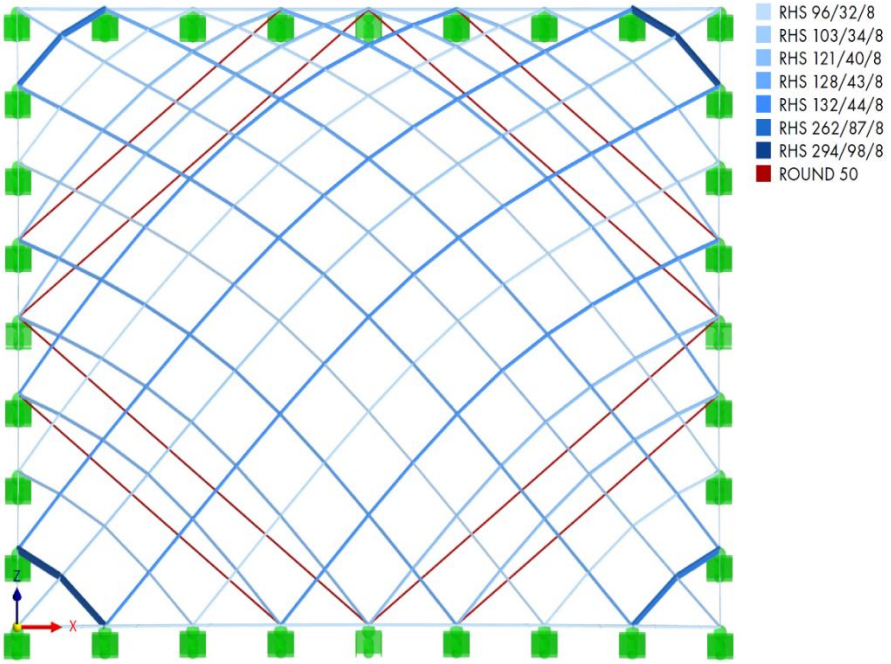


Figure 97: 9x9 Gridshell. Optimized cross-sections.
Unit of the cross-sections is [mm].

The max utilization ratios in the optimized design calculated in RFEM6 per cross-section are:

- RHS 96/32/8 → **0.71**
- RHS 103/34/8 → **0.96**
- RHS 121/40/8 → **0.93**
- RHS 128/43/8 → **0.92**
- RHS 132/44/8 → **0.92**
- RHS 262/87/8 → **0.90**
- RHS 294/98/8 → **0.80**

These values further validate the structural efficiency of the optimized design. Most of the cross-sections operate very close to their capacity limits, demonstrating that the optimization framework successfully exploited their load-bearing potential. The first cross-section RHS 96/32/8 has the lowest utilization ratio but it is also the bottom boundary for the cross-sectional area which means that given the problem definition it is the optimal result.

Furthermore, an observable pattern emerges in the cross-section distribution shown in Figure 97. This mean that PCA can optionally be applied. The same procedure as for the 4x4 gridshell is applied here to evaluate what the effect of the optimization in a reduced latent is in terms of quality of the results and computational time.

Figure 98 shows the explained variance ratio as a function of the number of principal components. The first principal component alone captures over 40% of the variance in the dataset, while the second accounts for approximately 13%. Beyond the first few components, the marginal contribution of additional components diminishes significantly, with components beyond the 10th each contributing less than 2% to the overall variance. This indicates that much of the variability in the 17-dimensional design space can be represented in a reduced subspace of lower dimensionality. This is confirmed in the plot of the cumulative explained variance plot in Figure 99.

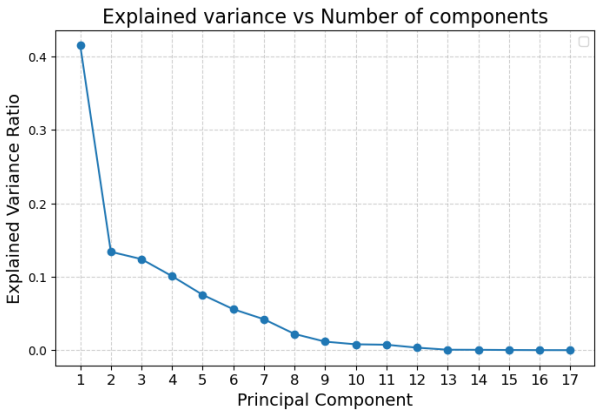


Figure 98: 9×9 Gridshell: Explained variance by each principle component.

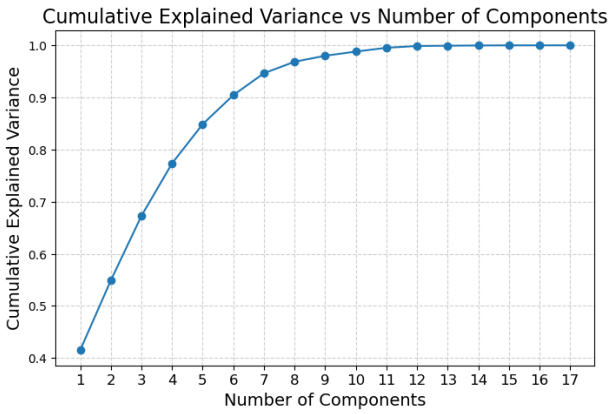


Figure 99: 9×9 Gridshell: Cumulative explained variance by each principle component.

The cases with 11 and 14 components are examined to assess the influence of PCA on the structure, serving as a sensitivity study.

Table 13 presents statistical performance indicators for the optimization of the 9×9 gridshell using PCA-reduced design spaces. The values indicate that increasing the number of principal components not only yields lighter structures but also enhances robustness and consistency across optimization runs.

Table 13: 9×9 Gridshell: Statistics for the PCA with different number of principle components.

PCA number of components	**11 principle components**	**14 principle components**
Best Weight [kg]	20921.38	15827.2
Worst Weight [kg]	24225.74	15906.11
Mean [kg]	22193.87	15865.16
Standard deviation [kg]	1598.03	34.47

Furthermore, Figure 100 depicts the convergence histories for Bayesian Optimization with 11 and 14 PCA components, compared against the best solution obtained without PCA (red dashed line). The optimization with 11 components stagnates around 22,000 kg on average, which is substantially higher than the benchmark solution. On the other hand, the 14-component configuration performs markedly better, steadily reducing the structural weight to an average of 15,865.16 kg, closely approaching the non-PCA best solution.

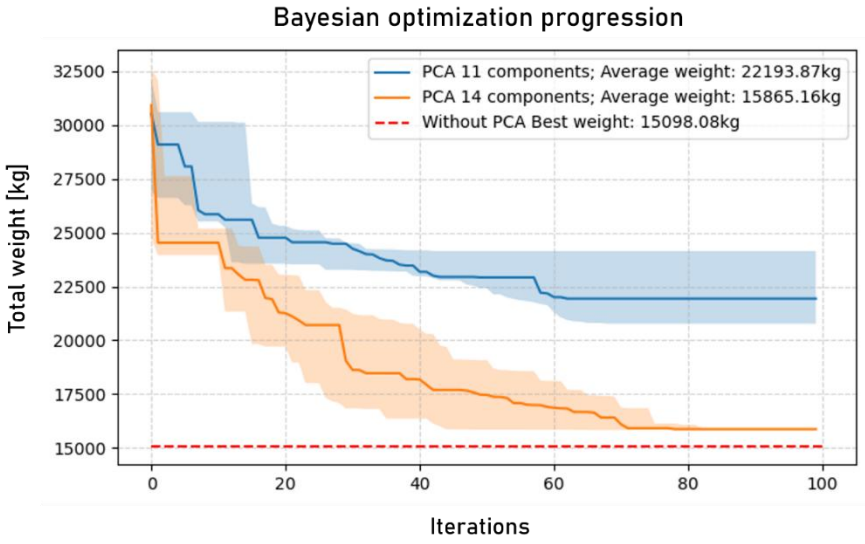


Figure 100: Bayesian optimization progress with PCA for 9×9 gridshell. Median objective function values and corresponding 95% confidence interval for the different number of principal components.

Finally, the convergence time comparison (Figure 101) provides further insight into the trade-off between accuracy and efficiency. The 11-component PCA run converged in approximately 130 minutes, while the 14-component run required around 150 minutes. The non-PCA case took 200 minutes on average. This pattern is consistent with the earlier 4×4 gridshell study, where PCA reduced computational cost but at the expense of solution quality.

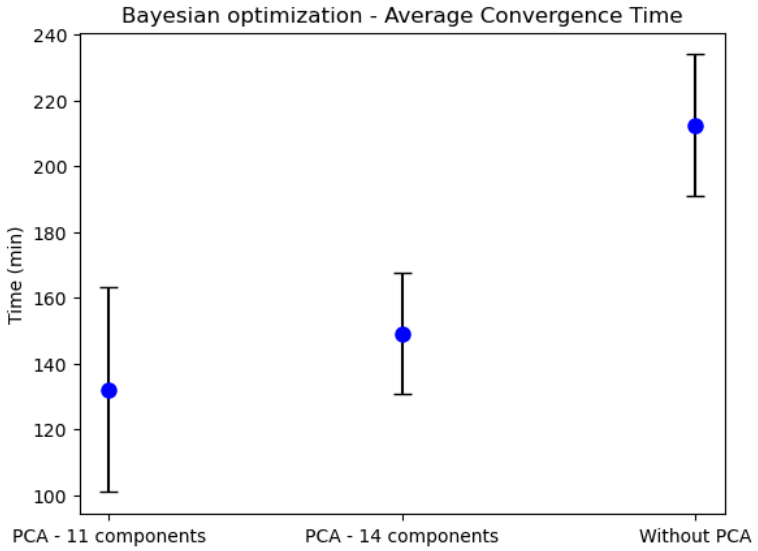


Figure 101: Average convergence time and 95% confidence interval comparison between the runs with PCA and without PCA for the 9×9 gridshell.

The results highlight the trade-off between computational efficiency and accuracy. When fewer components are retained (11 PCA), the optimization achieves faster convergence. However, this comes at the cost of significantly reduced solution quality. Increasing the number of components to 14 improves accuracy while still offering moderate computational savings compared to the full design space. By contrast, conducting the optimization without PCA yields the highest-quality solutions, albeit at the expense of the greatest computational cost.

6. Bayesian Optimization Tool

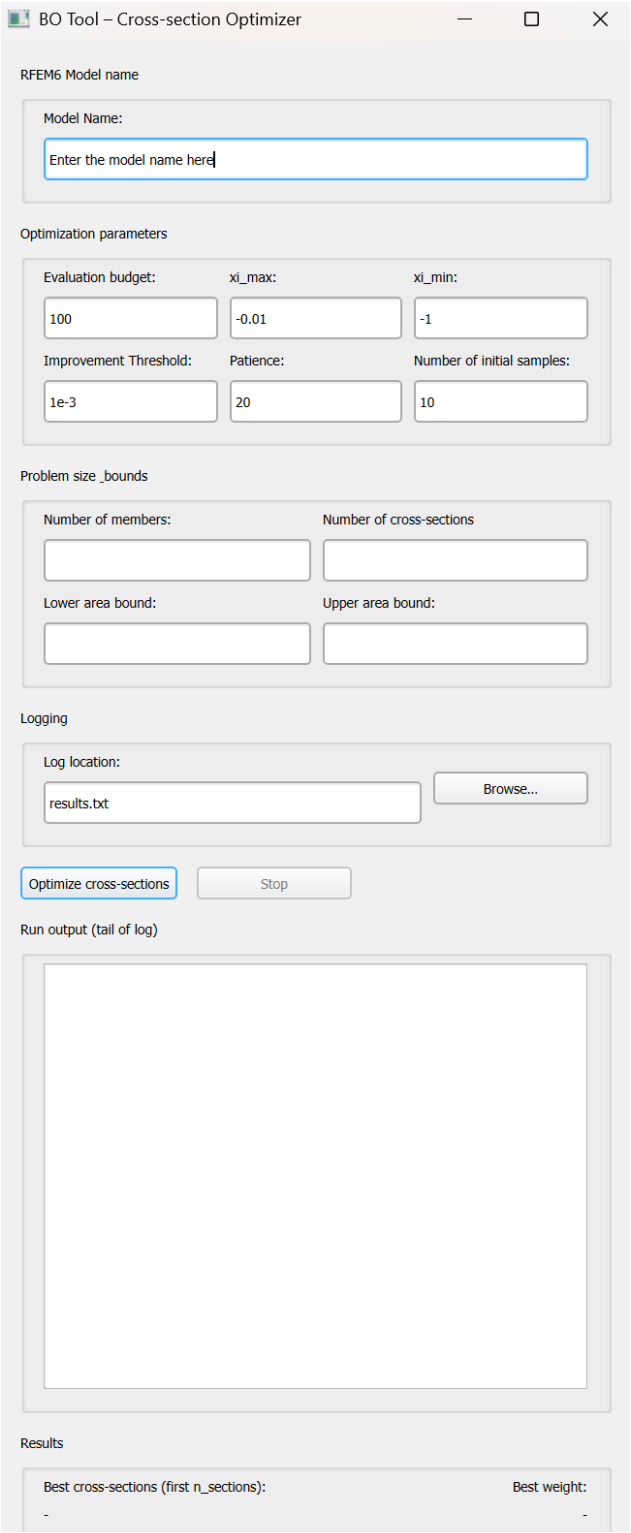


Figure 102: User interface of the BO Tool for cross-section optimisation.

The basic version of the "BO Tool – Cross-section Optimizer" shown in Figure 102 is designed to help structural engineers automate the selection of optimal cross-sections for members in a structure, using the constrained BO algorithm discussed in subchapter 4.1. Its interface is structured in a way that allows engineers to input key project information and optimization parameters without needing to handle the algorithmic details directly in the Python code.

The workflow begins with specifying the "**RFEM6 Model Name**", where the user enters the name of the structural model to be optimized. Next, under "**Optimization parameters**", the user defines the computational budget and tuning criteria. The *Evaluation budget* sets the maximum number of structural analyses the tool will perform. The ξ_{max} and ξ_{min} parameters define the range of acquisition function exploration values as defined in subchapter 3.4 and have default values of -0.01 and -1, while the *Improvement Threshold* sets the minimum improvement required to continue the optimization. The *Patience* value indicates how many iterations the tool will continue if no improvement is found, and *Number of initial samples* determines how many starting cross-section configurations are tested before the Bayesian Optimization loop begins.

The section "**Problem size bounds**" allows engineer to describe the structural system being optimized. Here, the number of members and the available cross-sections must be entered, along with upper and lower bounds for the cross-sectional area. These inputs ensure that the optimizer only searches within feasible engineering limits. In a future version of the tool these bounds can be extended to be per element, instead of being the same for all elements as it is currently.

For record-keeping, the "**Logging"** section specifies where the results are stored, with a default filename provided. Engineers can browse to select a different directory if needed. Once all parameters are set, clicking "**Optimize cross-sections"** starts the optimization process. The "**Run output"** panel then displays the progress of the optimization in real time, showing the tail of the log as the tool iteratively searches for improved solutions.

Finally, the "**Results"** box summarizes the outcome of the optimization by reporting the best-performing cross-sections (for the first nnn members) and the corresponding optimized structural weight. In this way, the tool translates the complex Bayesian Optimization procedure into a user-friendly interface, making it accessible for structural engineers who wish to efficiently minimize weight while satisfying design requirements.

The BO tool's user interface could be enhanced in future versions, and its functionality extended by incorporating parametric cross-sections of various types, such as HEA, IPE, and others.

7. Discussion

7.1. 1D optimization case: Cantilever Beam with size variables

Proof-of-concept integration of Python implementation of constrained BO with RFEM6

The successful implementation of this case study provides a proof-of-concept that the Python script of the Bayesian optimization can be integrated within industry-standard software packages such as RFEM6.

Disadvantages of using diverse cross-sections

A database of 516 diverse standardized cross-sections is used in the analysis of this problem, and the results show that the BO framework is able to reach close to optimal results within reasonable time. However, this diversity introduces a high degree of irregular (non-smooth) behaviour to the Gaussian processes used to model them as evidenced by Figure 106 - Figure 124 in Appendix A. As a result, for the following case studies only a single type of cross-section is used in the analysis, namely circular hollow sections (CHS) or rectangular hollow sections (RHS).

7.2. 2D optimization case: Cantilever Truss with shape and size variables

Advantages of using structural knowledge when shape optimization is considered

Unlike random search strategies, which rely solely on stochastic sampling, constrained BO benefits significantly from incorporating structural knowledge to guide the search process more effectively in tasks where shape (layout) of the structure is part of the optimization. For instance, to create the initial samples, the z-coordinates in the examples analysed in subchapter 5.2 are sampled in increasing order so that the resulting truss geometries are similar to a cantilever's moment diagram, with the truss height growing larger toward the supports. Therefore, incorporating domain-specific knowledge, such as the expected optimal shape of a cantilever truss, into the initial samples, the surrogate model's ability to recognize promising regions of the design space at an early stage of the optimization process is enhanced.

Advantages of using Gaussian processes per element

The results indicate that utilizing 18 GPs to model the constraint function for each steel element consistently produced better and more reliable outcomes compared to the approach with 4 GPs. The former is more effective in navigating the design space and managing constraint satisfaction per member, leading to improved performance across optimization runs.

Disadvantages of using standardized (discrete) cross-sections

It is important to note that the use of a standardized dataset of cross-sections presents several disadvantages in the context of constrained BO. The adoption of discrete area values, as dictated by the available cross-section database mentioned in the cantilever beam and truss case studies (described in subchapter 5.1 and 5.2, respectively) introduces discontinuities into the design space. This lack of smoothness can affect the accuracy of the Gaussian process surrogate model, which relies on the assumption of continuous objective and constraint functions.

As a result, the surrogate's predictions might produce results with higher variability due to the rounding of the cross-sectional area during the optimization process.

The need to map continuous algorithmic suggestions to the nearest available discrete value may introduce additional bias in a given design region, especially if the cross-section database is sparse. Therefore, the use of as many available profiles from the *same* type of cross-section (e.g. CHS, RHS) is recommended to approximate a continuous distribution of the area and avoid the disadvantages of diverse cross-sections mentioned before.

Advantages of using parametric (continuous) cross-sections

Employing parametric cross-sections in RFEM6 with dynamically adjustable areas solves the previous issue as it allows the continuous nature of the Gaussian processes to be more accurately represented.

7.3. 3D optimization case: 4x4 gridshell with size variables

Dimensionality sensitivity analysis

In this case study different dimensionalities of the optimization problem are analysed such as 1D, 17D, 18D and 28D. The findings presented in subchapter 5.3 indicate that the BO framework continues to reduce the structural weight up to 18D, which aligns with the empirically established limit of approximately 20 dimensions as generally reported in literature (e.g. Moriconi et al., 2020).

Number of initial samples sensitivity analysis

Constrained BO achieves much lower average weight of 830.16kg with 170 initial samples and 860.57kg with 17 initial samples. Notably, increasing the number of initial samples accelerates convergence, as reflected by the faster decline in structural weight during the early iterations. While both BO settings eventually stabilize around a similar weight range, the configuration with more initial samples demonstrates slightly better performance and reduced variance. This highlights the effectiveness of incorporating a larger initial design in guiding the surrogate model, thereby enabling the algorithm to explore the design space more efficiently and converge more reliably to near-optimal solutions.

Effect of Dimensionality Reduction via PCA

In addition to exploring dimensionality, Principal Component Analysis (PCA) is investigated as a dimensionality reduction technique to improve the scalability of the BO framework. The solution found for the best final design in the original 17D input space suggests that a pattern of similar cross-sections exists for groups of the structural elements. More details are given in subchapter 5.3.

Therefore, PCA is applied to decrease the complexity of the problem from the original 17D input space to a lower complexity namely to a 6D and 10D latent space. This is due to the fact that the cumulative explained variance analysis reveals that the first six principal components captured approximately 95% of the data variance, while ten components are sufficient to capture nearly 100%.

Based on the findings in subchapter 5.3 & 5.4 the trade-off between efficiency and solution quality is highlighted:

- Fewer PCA components improve computational speed but degrade optimization accuracy.
- More PCA components retain variance and yield closer-to-optimal results but reduce the computational advantage.
- Without PCA, optimization achieves the best weight but requires more computational effort.

The use of the BO framework in the original input feature space is recommended when the underlying pattern of the structure is not well known as suggested also in the paper by Eriksson & Jankowiak (2021).

Moreover, the inherent linearity of PCA may be limiting its effectiveness in supporting the optimization process. Therefore, variational autoencoders can be used as a more advanced method. This model allows mappings to and from the latent space to be arbitrarily non-linear (Bishop, 2006).

7.4. 3D optimization case: 9x9 gridshell with size variables

Scalability of the constrained BO algorithm

One of the key distinctions from the previous gridshell is that steel elements are grouped according to two lines of symmetry, with a single cross-section assigned to each group rather than to individual members. This approach brings the optimization problem closer to practical engineering applications where adjacent elements in an arch share the same cross-section. Also, it reduces the dimensionality from 288D to 17D which is within the ~20D limit for BO. This demonstrates, that by grouping the elements based on their structural function can help in the optimization process.

Overall, the 9×9 gridshell study extends the earlier findings by demonstrating that Bayesian Optimization can handle much larger structural systems, though with increased computational demands. PCA can offer a practical tool for reducing runtime in such cases, but the dimensionality threshold must be chosen carefully to avoid discarding key structural variance.

8. Conclusion

In this chapter the main conclusions are presented. This is done by answering the sub-questions outlined in Chapter 1.4 which help answer the main research question mentioned below:

"To what extent can Bayesian optimization be applied to efficiently optimize the shape and cross-sections of structures in terms of minimizing structural weight, while ensuring structural integrity and integration with industry-standard tools like RFEM6?"

RQ1: How can Bayesian optimization be used in the weight optimization of macrostructures?

Based on the analysed cases in Chapter 5, the constrained Bayesian optimization framework applied in this research is found to be performing well for weight minimization of macrostructures under structural constraints. The total structural weight is treated as the objective function while the structural performance requirements for strength and stability serve as the constraints. The algorithm uses a probabilistic surrogate, namely a Gaussian process to predict both expected improvement of the objective function and the feasibility probability of the sampled point producing designs that have unity checks below the threshold values of the constraints. As a result, this combination enables a sample-efficient search of the design space.

In order to identify the next most likely point the maximization of an acquisition function is a critical step in the framework. Empirically it has been found that the constrained Expected Improvement (cEI) acquisition function offers a balanced exploration–exploitation trade-off by prioritizing candidate designs that promise weight reduction while maintaining a high predicted probability of meeting all structural constraints as described in Chapter 4.

To start the optimization loop, initial training samples have to be generated using a space-filling sampling strategy such as the Latin Hypercube Sampling explained in Chapter 3.6. The number of initial samples can have an effect on the variance of the outcomes as was found during the analysis of the high-dimensional gridshell cases. Larger number of initial samples, e.g. 10 points per input dimension, reduces the variance of the results but increases the computational time because more FE evaluations are necessary by default when compared to only 1 point per input dimension.

The relevant hyperparameters of the model are the length scale values for the weight and constraint kernels and the exploration-exploitation trade-off parameter ξ which is part of the acquisition function. The former is re-calculated every 5^{th} iteration by maximizing the log marginal likelihood of the data using the maximum likelihood estimation (MLE) approach as explained in subchapter 3.5. This frequency has been determined empirically by varying the frequency of the hyperparameter optimization. Notably, performing this operation every iteration sometimes yielded worse results because it caused the model to overfit the data. The computational cost also increases with the frequency of hyperparameter tuning. Similar findings have been reported by (Shende et al., 2021). The range of optimal values for the second hyperparameter ξ used in the adaptive linear function explained in subchapter 3.4 has been found to be between $\xi_{min} = 0.01$ and $\xi_{max} = 1$ but with a negative sign since the objective is the minimization of the total structural weight.

Regarding the stopping criteria, usually in applications of the Bayesian optimization framework there is a given evaluation budget that limits the number of experiments the practitioners or scientists can make. Since in this thesis the framework is used for the optimization of macrostructures there is no such requirement. The heuristic used to determine when the algorithm should stop is defined by a "patience" and an "improvement threshold" variables. The first one determines how many iterations the algorithm is allowed to continue without seeing any significant improvement of the weight of the structure. This value is set empirically to 30 iterations to allow for more exploration of the design space before stopping the algorithm while keeping the computational time within a reasonable limit. The improvement threshold is set empirically to a small value of 0.001, which ensures that any promising regions of the design space are not skipped by the algorithm but can be explored further before returning the previous design region. Both values can be adjusted by the user depending on the available time for the optimization task. Some general conclusion points that apply to all of the analysed cases with the standard constrained BO (subchapter 4.1) are discussed below.

First, all runs converged to constraint-satisfying designs, confirming the reliability of the BO setup. An intermediate kernel dimensionality (e.g. 17D) yields clearly better solutions and stability (i.e. lower variance of results) than both an overly compressed (e.g. 1D) and the full-dimensional kernel (e.g. 28D for the 4x4 gridshell).

Second, a larger number of initial designs improves convergence quality and narrows variability compared with a small initial set and with random search. This is in contrast to the findings of Shende et al. (2021) who state that a larger initial training set does not seem to guarantee a better solution or faster convergence. Therefore, the sensitivity of BO to the number of initial samples can be considered as problem-specific and has to be checked depending on the data.

Third, evaluation cost rises with kernel dimensionality while the quality of the final designs improves.

When compared with literature case studies as done in subchapter 5.2, the Bayesian optimization framework delivers near-optimal results at dramatically lower FEM evaluation cost. For the cantilever truss case the convergence rate is *18 times* faster than the reference solution found in literature with only a 0.5% increase in the total weight of the structure.

All of the case studies analysed in the thesis demonstrate the broad applicability of the proposed method for structures with 1D, 2D and 3D geometry and its potential to solve complex design problems. The achieved results via the constrained BO algorithm for the 4x4 gridshell show a *1.67*× lighter structure than the reference design and *2.54* × lighter structure than the reference design for the 9x9 gridshell.

Furthermore, a specific dimensionality reduction technique has been explored for the input feature space to determine their effectiveness in terms of computational time and obtained results. As discussed in subchapter 3.8 the linear deterministic Principle Component Analysis (PCA) has been used. It proves effective in reducing dimensionality for the 4x4 gridshell and the 9x9 gridshell cases analysed in subchapter 5.3 and subchapter 5.4, though its benefits depend strongly on the

number of components retained. A small number of components leads to faster runtimes but may introduce randomness and compromise both accuracy and consistency, particularly if the structural patterns underlying the cross-section distribution are not very clear. Retaining more components improves solution quality and stability but increases computational cost. Nevertheless, PCA still converges faster compared to working in the full design space, albeit at the expense of obtaining heavier designs. Therefore, it can be considered as an optional approach compared to the standard constrained BO.

In addition, one limitation of the present thesis must be acknowledged. The constrained BO has been evaluated only under a single load case/combination for all of the analysed case studies. In practical structural engineering applications, members are typically subjected to a variety of load combinations representing different design situations and service conditions. The extension of the current approach to handle multiple or combined load combinations remains to be investigated, and its effectiveness in such scenarios is yet to be validated.

In conclusion, the Bayesian optimization framework holds a large potential in the optimization of macrostructures designed with materials different from steel such as wood, concrete, glass etc. To achieve this all of the design checks that serve as the constraints for the strength and stability have to be adapted to the country specific code requirements for the given material. In addition, the library of cross-sections can be expanded to include open parametric profiles such as HEA, HEB, IPE etc. Furthermore, in the present thesis the global stability has been excluded from the constraints of the design due to implementation challenges, but in a future version of the BO framework it can be included as a separate Gaussian process that models this global constraint.

Also, it is concluded that the effectiveness of the algorithm is dependent only on the distribution of the internal forces and the type of finite element used. Therefore, from a theoretical point of view, the proposed framework can be applied to any structure independent of the boundary conditions as long as the internal forces and the design checks can be calculated accurately by the FEM package. Further research is needed to evaluate the performance of the Bayesian optimization on various structural typologies such as moment frames and even bridges.

In addition, the analysed cases can serve as a benchmark for researchers to compare different optimization approaches.

RQ2: How can a gridshell be modelled in RFEM6 and what is the obtained structural behaviour?

The geometry of the two gridshells is generated by employing form-finding methods that align the structural shape with the expected load-bearing behaviour. The Kangaroo2 plugin for Grasshopper is used, which implements one of the most popular form-finding techniques based on the dynamic relaxation principle. This approach iteratively adjusts the geometry by simulating the equilibrium of forces within the system, allowing the structure to naturally settle into a form that efficiently carries the applied loads through membrane actions.

The structural model in RFEM6 is developed using beam elements, which provide a good representation of the gridshell's members while capturing their axial, bending, and shear behaviour as explained in subchapter 4.3. The governing load combination considered in the analysis of the

larger 9×9 gridshell is defined in reference to the C30 gridshell project described briefly in subchapter 5.4.1. This example served as a guideline to establish realistic loading scenarios, ensuring that the model reflects comparable structural demands and environmental influences. The boundary conditions of the model are likewise derived from the C30 gridshell, allowing the numerical representation to reproduce the support conditions observed in practice. By adopting these boundary conditions, the FEM analysis ensures consistency with a realistic case study. Furthermore, rectangular hollow sections (RHS) are selected as the cross-sections for the analysis. These closed profiles are chosen due to their lower susceptibility to lateral torsional buckling when compared to open cross-sections and their suitability for gridshell applications as evidenced in the C30 case study.

Finally, the strength and stability checks are calculated in RFEM6 and used in the Bayesian optimization algorithm according to the EN 19931-1-1 recommendations as described in Chapter 4.3.2 & 4.3.3 which ensures the structural feasibility of the gridshells and their compliance with the design codes.

9. Recommendations for future research

Given the efficiency of the Bayesian optimization framework for optimizing the design of macrostructures, some directions for future research are provided in this chapter. They focus on two goals: (i) scalability and computational acceleration of the BO framework in higher dimensional feature space and (ii) extending the framework from single-objective optimization to multi-objective optimization. Throughout the thesis, various recommendations for future research are also suggested.

Scalability and computational acceleration of the BO framework in higher dimensional feature space

For high-dimensional settings, scalability can be improved by methods that reduce the effective search space while preserving expressiveness. Sparse Axis-Aligned Subspace BO (SAASBO) can be adopted to infer sparse relevance patterns and prevent overfitting from fully-ARD kernels as discussed by (Eriksson & Jankowiak, 2021). They have used it for the hyperparameter tuning of a support vector machine (SVM) using the Matern kernel with 3 regularization parameters and 385 length scales making the dimensionality D = 388. Similarly, the authors have benchmarked the approach against other BO variants that rely on low dimensional embedding of the high dimensional feature space such as REMBO and HeSBO on a real-world vehicle design problem in a crash test simulation called MOPTA08 shown in Figure 104 for weight optimization (pSeven, 2018). The problem consists of 124 design variables normalized to [0,1] and 68 performance constraints. They report that SAASBO converges faster to a close value to the known optimum for both problems as shown by the blue line in Figure 103.

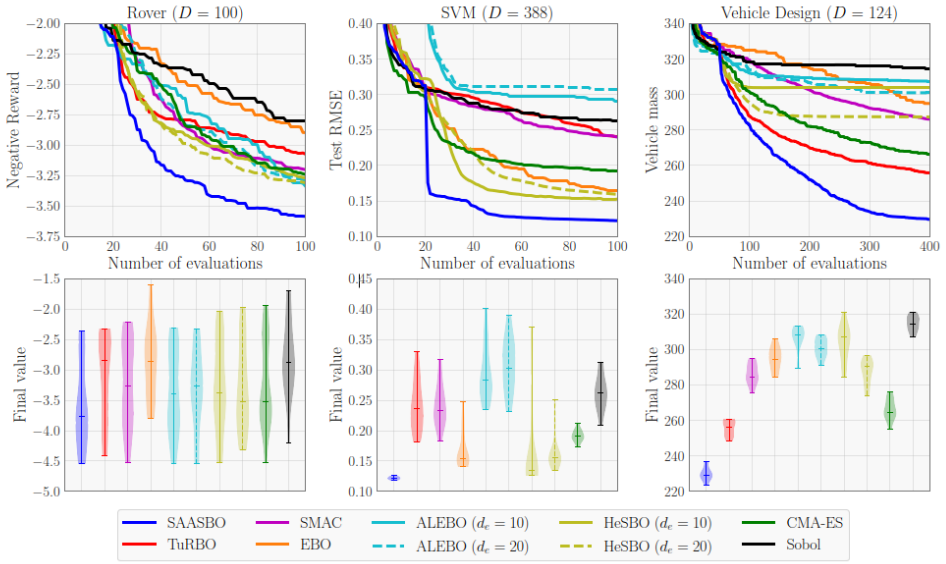


Figure 103: SAASBO performance compared to other Bayesian optimization variants. (*Source: Eriksson & Jankowiak, 2021*)

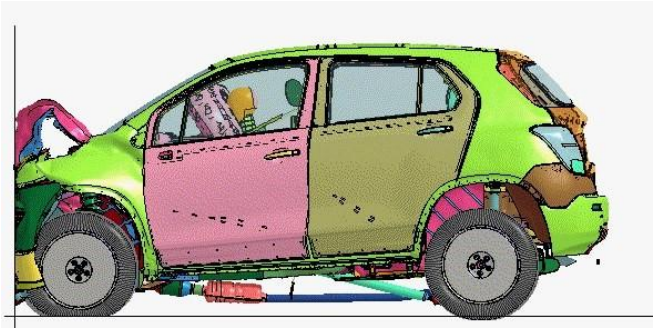


Figure 104: MOPTA08 vehicle design problem. (*Source: pSeven, 2018*)

Another direction for further investigation is the application of the Blackbox Matrix-Matrix Inference introduced by Gardner et al., 2018. The authors claim that their method reduces the computational complexity from $O(N^3)$ to $O(N^2)$ which can reduce the execution time significantly. They do this by calculating the marginal log likelihood mentioned in subchapter 3.5 in a stochastic manner instead of performing an exact calculation. By using the developed GPyTorch package and coupled with GPU acceleration, the authors claim it can scale the GP inference to thousands of data points and in the documentation of the package even millions of data points is mentioned.

A final third recommendation is the use of trust regions (i.e. TuRBO) in the design space as proposed by Eriksson et al., 2019. Instead of relying on a single global GP model, TuRBO employs multiple local probabilistic models that focus search within trust regions of the objective function design space. These local searches can quickly uncover high-quality solutions, while a global sampling strategy decides how to distribute samples among the regions, ensuring an effective balance between exploration and exploitation.

Multi-objective Bayesian optimization

To further extend this research a modular Bayesian optimization (BO) framework can be developed. The different "modules" could focus on optimizing an objective function that is selected by the user. A few example functions are listed below:

- Minimizing the weight of the structure based on the ULS combinations which is already demonstrated in this thesis.
- Minimizing the total deflection of the structure based on the SLS combinations.
- Minimizing the CO_2 emissions of the structure.
- Minimizing the construction cost of the structure by including the design of the connections.
- Minimizing/Maximizing a particular geometry parameter such as height or width of a truss.

The corresponding adaptations of the BO framework can be either single-objective or multi-objective optimization routines by combining multiple of the above-mentioned functions. In the latter case, there is no single solution that represents the best design, but the result is a list of options to be considered and the trade-offs to be evaluated by the engineer. This is the so-called Pareto front shown in Figure 105.

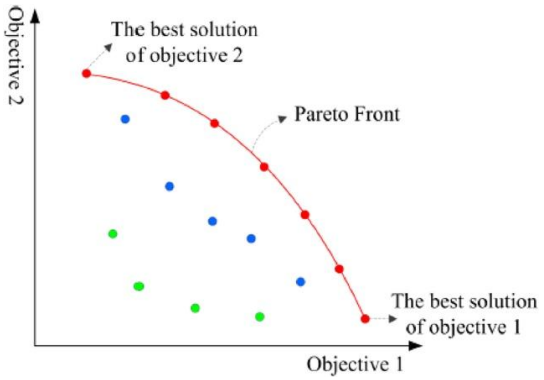


Figure 105: Pareto front between objective 1 and objective 2. (*Source:Rahman & Szabó, 2021)*

In the context of the BO framework the multi-objective optimization can be achieved by multi-task GPs with multiple outputs or a combination of single-task GPs. This setup enables the Expected Hypervolume Improvement (EHVI) acquisition function which is the multi-objective analogue of Expected Improvement (Coelho, 2025).

An example of such trade-offs can happen when the CO_2 emissions of the building have to be minimized alongside the cost of construction. The main structural elements can be designed with less material if moment resistant connections are included in the design instead of the simpler pin connections. However, these are often more expensive to both design and produce leading to increased construction costs. Another example is when the weight of the structure is optimized with different profiles for the members as is done in Chapter 5 but often this increases the material costs because ordering many custom profiles is usually more expensive than ordering single profiles in bulk quantities.

Therefore, a balance must be found by the structural engineer and the purpose of the BO optimization tool is to facilitate this decision-making process.

10. Bibliography

Allwood, J. M., Cullen, J. M., & Milford, R. L. (2010). Options for Achieving a 50% Cut in Industrial Carbon Emissions by 2050. Environmental Science & Technology, 44(6), 1888–1894. https://doi.org/10.1021/es902909k

Fisher, B.; Nakicenovic, N. (2007) Issues related to mitigation in the long-term context. In Climate Change 2007: Mitigation: Contribution of Working Group III to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change; Metz, B.,Davidson, O. R., Bosch, P. R., Dave, R., Meyer, L. A., Eds.;Cambridge University Press: Cambridge, UK.

UNFCCC. Adoption of the Paris Agreement. Report No. FCCC/CP/2015/L.9/Rev.1, http://unfccc.int/resource/docs/2015/cop21/eng/l09r01.pdf

Chen, Y., Fang, Y., Feng, W., Liu, S., & Zhao, G. (2022). How to minimise the carbon emission of steel building products from a cradle-to-site perspective: A systematic review of recent global research. Journal of Cleaner Production, 368, 133156. https://doi.org/10.1016/j.jclepro.2022.133156

WSA (2024). World Steel in Figures 2024 - Worldsteel.org. worldsteel.org. https://worldsteel.org/data/world-steel-in-figures-2024/

Somers, J. (2021). Technologies to decarbonise the EU steel industry, EUR 30982 EN, Publications Office of the European Union, Luxembourg, ISBN 978-92-76-47147-9 (online), doi:10.2760/069150 (online), JRC127468

Klimaatwet. (2019). (No. 253 of 2019). In Government of the Netherlands. Staatsblad van het Koninkrijk der Nederlanden.

Dyvik, S. H., Manum, B., & Rønnquist, A. (2021). Gridshells in Recent Research—A Systematic Mapping Study. *Applied Sciences*, *11*(24), 11731. https://doi.org/10.3390/app112411731

Grande, E., Imbimbo, M., & Tomei, V. (2017a). Structural Optimization of grid shells: design parameters and combined strategies. Journal of Architectural Engineering, 24(1). https://doi.org/10.1061/(asce)ae.1943-5568.0000286

Grande, E., Imbimbo, M., & Tomei, V. (2017b). Role of global buckling in the optimization process of grid shells: Design strategies. *Engineering Structures*, *156*, 260–270. https://doi.org/10.1016/j.engstruct.2017.11.049

Adriaenssens, S., Block, P., Veenendaal, D., & Williams, C. J. K. (2014). *Shell structures for architecture: Form Finding and Optimization*. Routledge.

Mesnil, R., Douthe, C., & Baverel, O. (2018). Non-Standard patterns for gridshell structures: fabrication and structural optimization. *Journal of the International Association for Shell and Spatial Structures*, *58*(4), 277–286. https://doi.org/10.20898/j.iass.2017.194.893

Borgart, A. (2024a). The relationship between geometric and mechanical properties of shell structures. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:00ab51d6-55db-453a-9a62-23dfbd439b7b

Borgart, A. (2024b). Lecture on gridshells. *CIEM5250: Building Engineering course*. Delft University of Technology.

Jose Martinez Castro, Wolf Song (2024). Generative Form Finding with Kangaroo Physics. In W.S. Elkhuizen (Ed.), Computational Design for (Industrial) Designers, using Rhino Grasshopper. Delft University of Technology. https://doi.org/10.59490/tb.87

Ravisetti, M. (2023, January 9). Scientists reveal why Rome's pantheon has yet to crumble. *CNET*. https://www.cnet.com/science/biology/scientists-reveal-why-romes-pantheon-has-yet-to-crumble/

Structurae. (n.d.). Palazzetto dello Sport (Rome, 1957) | Structurae. *Structurae*. https://structurae.net/en/structures/palazzetto-dello-sport

Carlos Lázaro. (2023). *Félix Candela's last shell | Institut für Baustatik und Baudynamik | Universität Stuttgart*. (n.d.). Universität Stuttgart. https://www.ibb.uni-stuttgart.de/aktuelles/vortragsreihe-tragwerke/2023-lazaro/

Ahm, P., & Perry, E. J. (1965). DESIGN OF THE DOME SHELL ROOF FOR SMITHFIELD POULTRY MARKET. *ICE Proceedings*, *30*(1), 79–130. https://doi.org/10.1680/iicep.1965.9567

Julian Harrap Architects. (n.d.). *Museum of London Smithfield Market*. https://julianharraparchitects.co.uk/projects/museum-of-london

P.C.J. Hoogenboom. (2024). Inextensional deformation. https://www.phoogenboom.nl/b17_handout_5.pdf

Williams, C. (2014). What is a shell? In S. Adriaenssens, P. Block, D. Veenendaal, & C. Williams (Eds.), Shell structures for architecture (pp. 21–32). Abingdon: Routledge.

Zexin, S., & Mei, H. (2017). Robotic Form-Finding and construction based on the architectural projection logic. *IOP Conference Series Materials Science and Engineering*, *216*, 012058. https://doi.org/10.1088/1757-899x/216/1/012058

Tamplin, R & Iuorio, Ornella. (2018). Challenges in designing and fabrication of a thin concrete shell.

Chiang, Y. (2022). *Design and fabrication of shell structures. aided by radial basis functions and reconfigurable mechanisms*. Delft University of Technology. https://doi.org/10.7480/abe.2022.03.6269

Buro Happold. (2019). *Above and beyond: bringing our structural expertise to airport roofs*. Buro Happold. https://www.burohappold.com/news/above-and-beyond-bringing-our-structural-expertise-to-airport-roofs/

Ermias Y., Shabtai B. and Szerzo P. (2013). *Museum of Hamburg History - Evolution of German Shells: Efficiency in form*. Princeton University. https://shells.princeton.edu/Ham.html

Ney & Partners (2011). *Glass Roof Dutch Maritime Museum*. https://ney.partners/project/glass-roof-dutch-maritime-museum/

Sigrid Adriaenssens. (2019). *Structures in the low countries: Cupola over the Dutch Maritime Museum*. Form Finding Lab. https://formfindinglab.wordpress.com/2017/11/15/structures-in-the-low-countries-cupola-over-the-dutch-maritime-museum/

Gythiel, W., & Schevenels, M. (2022). Gradient-based size, shape, and topology optimization of single-layer reticulated shells subject to distributed loads. *Structural and Multidisciplinary Optimization*, *65*(5). https://doi.org/10.1007/s00158-022-03225-w

European Committee for Standardization (2005). EN 1993-1-1: Eurocode 3: design of steel structures—Part 1-1: general rules and rules for buildings. European Committee for Standardization

Kawamura, H., Ohmori, H., & Kito, N. (2002). Truss topology optimization by a modified genetic algorithm. *Structural and Multidisciplinary Optimization*, *23*(6), 467–473. https://doi.org/10.1007/s00158-002-0208-0

Richardson, J. N., Adriaenssens, S., Coelho, R. F., & Bouillard, P. (2013). Coupled form-finding and grid optimization approach for single layer grid shells. *Engineering Structures*, *52*, 230–239. https://doi.org/10.1016/j.engstruct.2013.02.017

EurocodeApplied. (2025a). Table of design properties for steel tubes - Circular Hollow Sections (CHS). EurocodeApplied.com. Accessed April 2025: https://eurocodeapplied.com/design/en1993/chs-design-properties

EurocodeApplied. (2025b). Table of design properties for flanged steel profiles (IPE, HEA, HEB, HEM, UB, UC, UBP) EurocodeApplied.com. Accessed April 2025: https://eurocodeapplied.com/design/en1993/ipe-hea-heb-hem-design-properties

EurocodeApplied. (2025c). Table of design properties for Square Hollow Sections (SHS). EurocodeApplied.com. Accessed April 2025: https://eurocodeapplied.com/design/en1993/shs-design-properties

EurocodeApplied. (2025d). *Table of design properties for rectangular steel profiles - Rectangular Hollow Sections (RHS).* EurocodeApplied.com. Accessed April 2025: https://eurocodeapplied.com/design/en1993/rhs-design-properties

EurocodeApplied. (2025e). *Table of design properties for rectangular steel profiles - Rectangular Hollow Sections (RHS).* EurocodeApplied.com. Accessed April 2025: https://eurocodeapplied.com/design/en1993/rhs-design-properties

Bouwen met staal. *(2025).* Bouwen met staal | Profiel vinden. Accessed April 2025: https://www.bouwenmetstaal.nl/tools/staalprofielen/profiel-vinden

Table of design properties for Rectangular Hollow Sections RHS

EN 1993-1-1:2005. (2005). CEN - European Committee for Standardization. Eurocode 3: Design of steel structures - Part 1- 1: General rules and rules for buildings.

Tapeh, A.T.G., Naser, M.Z. (2023).Artificial Intelligence, Machine Learning, and Deep Learning in Structural Engineering: A Scientometrics Review of Trends and Best Practices. *Arch Computat Methods Eng* **30**, 115–159 . https://doi.org/10.1007/s11831-022-09793-w

Atul. (2025, February 15). AI vs Machine Learning vs Deep Learning. Edureka. https://www.edureka.co/blog/ai-vs-machine-learning-vs-deep-learning/

Kapoor, T., Wang, H., Núnez, A., & Dollevoet, R. (2023). Physics-Informed neural networks for solving forward and inverse problems in complex beam systems. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(5), 5981–5995. https://doi.org/10.1109/tnnls.2023.3310585

Blum, Christian & Roli, Andrea. (2001). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. ACM Comput. Surv.. 35. 268-308. 10.1145/937503.937505.

Zavala, G. R., Nebro, A. J., Luna, F., & Coello, C. a. C. (2013). A survey of multi-objective metaheuristics applied to structural optimization. *Structural and Multidisciplinary Optimization*, *49*(4), 537–558. https://doi.org/10.1007/s00158-013-0996-4

Tran, A., Tran, M., & Wang, Y. (2019). Constrained mixed-integer Gaussian mixture Bayesian optimization and its applications in designing fractal and auxetic metamaterials. *Structural and Multidisciplinary Optimization*, *59*(6), 2131–2154. https://doi.org/10.1007/s00158-018-2182-1

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1206.2944

Huang, C., Lee, J., Zhang, Y., Zhou, S., & Tang, J. (2022). Mixed-Input Bayesian Optimization Method for Structural damage diagnosis. *IEEE Transactions on Reliability*, *72*(2), 678–691. https://doi.org/10.1109/tr.2022.3179602

Saka, M. (2009). Optimum Design of Steel Frames using Stochastic Search Techniques Based on Natural Phenomena: A Review. *Computational Science, Engineering and Technology Series*, 105–147. https://doi.org/10.4203/csets.16.6

Wei, L., Tang, T., Xie, X., & Shen, W. (2011). Truss optimization on shape and sizing with frequency constraints based on parallel genetic algorithm. *Structural and Multidisciplinary Optimization*, *43*(5), 665–682. https://doi.org/10.1007/s00158-010-0600-0

Tang, W., Tong, L., & Gu, Y. (2005). Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables. *International Journal for Numerical Methods in Engineering*, *62*(13), 1737–1762. https://doi.org/10.1002/nme.1244

Rajasekaran S. (2001). Optimization of Large Scale Three Dimensional Reticulated Structures Using Cellular Genetics and Neural Networks. *International Journal of Space Structures*. ;16(4):315-324. doi:10.1260/026635101760832244

Gholizadeh, S. (2013). Layout optimization of truss structures by hybridizing cellular automata and particle swarm optimization. *Computers & Structures*, *125*, 86–99. https://doi.org/10.1016/j.compstruc.2013.04.024

Mortazavi, A., & Toğan, V. (2016). Simultaneous size, shape, and topology optimization of truss structures using integrated particle swarm optimizer. *Structural and Multidisciplinary Optimization*, *54*(4), 715–736. https://doi.org/10.1007/s00158-016-1449-7

Gomes, H. M. (2011). Truss optimization with dynamic constraints using a particle swarm algorithm. *Expert Systems With Applications*, *38*(1), 957–968. https://doi.org/10.1016/j.eswa.2010.07.086

Luh, G., & Lin, C. (2011). Optimal design of truss-structures using particle swarm optimization. *Computers & Structures*, *89*(23–24), 2221–2232. https://doi.org/10.1016/j.compstruc.2011.08.013

Li, L., Huang, Z., Liu, F., & Wu, Q. (2007). A heuristic particle swarm optimizer for optimization of pin connected structures. Computers & Structures, 85(7–8), 340–349. https://doi.org/10.1016/j.compstruc.2006.11.020

Camp, C. V., & Bichon, B. J. (2004). Design of Space Trusses Using Ant Colony Optimization. *Journal of Structural Engineering*, *130*(5), 741-751. https://doi.org/10.1061/(asce)0733-9445(2004)130:5(741)

Aydoğdu, İ., & Saka, M. (2011). Ant colony optimization of irregular steel frames including elemental warping effect. *Advances in Engineering Software*, *44*(1), 150–169. https://doi.org/10.1016/j.advengsoft.2011.05.029

Degertekin, S. (2012). Improved harmony search algorithms for sizing optimization of truss structures. *Computers & Structures*, *92–93*, 229–241. https://doi.org/10.1016/j.compstruc.2011.10.022

Cheng, M., Prayogo, D., Wu, Y., & Lukito, M. M. (2016). A Hybrid Harmony Search algorithm for discrete sizing optimization of truss structure. *Automation in Construction*, *69*, 21–33. https://doi.org/10.1016/j.autcon.2016.05.023

Miguel, L. F. F., & Miguel, L. F. F. (2012). Shape and size optimization of truss structures considering dynamic constraints through modern metaheuristic algorithms. *Expert Systems With Applications*, *39*(10), 9458–9467. https://doi.org/10.1016/j.eswa.2012.02.113

Pham, H. A. (2016). Truss optimization with frequency constraints using enhanced differential evolution based on adaptive directional mutation and nearest neighbor comparison. Advances in Engineering Software, 102, 142–154. https://doi.org/10.1016/j.advengsoft.2016.10.004

Ho-Huu, V., Vo-Duy, T., Luu-Van, T., Le-Anh, L., & Nguyen-Thoi, T. (2016b). Optimal design of truss structures with frequency constraints using improved differential evolution algorithm based on an adaptive mutation scheme. *Automation in Construction*, *68*, 81–94. https://doi.org/10.1016/j.autcon.2016.05.004

Nguyen-Van, S., Nguyen, K. T., Luong, V. H., Lee, S., & Lieu, Q. X. (2021). A novel hybrid differential evolution and symbiotic organisms search algorithm for size and shape optimization of truss structures under multiple frequency constraints. *Expert Systems With Applications*, *184*, 115534. https://doi.org/10.1016/j.eswa.2021.115534

Ho-Huu, V., Nguyen-Thoi, T., Vo-Duy, T., & Nguyen-Trang, T. (2016). An adaptive elitist differential evolution for optimization of truss structures with discrete design variables. *Computers & Structures*, *165*, 59–75. https://doi.org/10.1016/j.compstruc.2015.11.014

Hasançebi, O., Çarbaş, S., Doğan, E., Erdal, F., & Saka, M. (2009). Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures. *Computers & Structures*, *87*(5–6), 284–302. https://doi.org/10.1016/j.compstruc.2009.01.002

Shende, S., Gillman, A., Yoo, D., Buskohl, P., & Vemaganti, K. (2021). Bayesian topology optimization for efficient design of origami folding structures. *Structural and Multidisciplinary Optimization*, *63*(4), 1907–1926. https://doi.org/10.1007/s00158-020-02787-x

Mathern, A., Steinholtz, O. S., Sjöberg, A., Önnheim, M., Ek, K., Rempling, R., Gustavsson, E., & Jirstrand, M. (2020). Multi-objective constrained Bayesian optimization for structural design.

Structural and Multidisciplinary Optimization, *63*(2), 689–701. https://doi.org/10.1007/s00158-020-02720-2

Moriconi, R., Deisenroth, M. P., & Kumar, K. S. S. (2020). High-dimensional Bayesian optimization using low-dimensional feature spaces. *Machine Learning*, *109*(9–10), 1925–1943. https://doi.org/10.1007/s10994-020-05899-z

Coelho, R. C., Alves, A. F. C., Pires, T. N., & Pires, F. A. (2025). A composite Bayesian optimisation framework for material and structural design. *Computer Methods in Applied Mechanics and Engineering*, *434*, 117516. https://doi.org/10.1016/j.cma.2024.117516

Rasmussen, C. E., & Williams, C. K. I. (2005). *Gaussian processes for machine learning*. MIT Press.

DSAIE (2024). Data Science and Artificial Intelligence for Engineers Course given at Delft University of Technology.

Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing, 16(5), 1190–1208. https://doi.org/10.1137/0916069

Lizotte, D. (2008). Practical Bayesian Optimization. PhD thesis, University of Alberta, Edmonton, Alberta, Canada. https://doi.org/10.7939/r3-379f-gw15

Greif, L., Hübschle, N., Kimmig, A., Kreuzwieser, S., Martenne, A., & Ovtcharova, J. (2025). Structured sampling strategies in Bayesian optimization: evaluation in mathematical and real-world scenarios. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-025-02597-2

Eriksson, D., & Jankowiak, M. (2021). High-Dimensional Bayesian Optimization with Sparse Axis-Aligned Subspaces. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2103.00349

Gardner, J. R., Pleiss, G., Bindel, D., Weinberger, K. Q., & Wilson, A. G. (2018). GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1809.11165

Eriksson, D., Pearce, M., Gardner, J. R., Turner, R., & Poloczek, M. (2019). Scalable global optimization via local Bayesian optimization. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1910.01739

Rahman, M. M., & Szabó, G. (2021). Multi-objective urban land use optimization using spatial data: A systematic review. Sustainable Cities and Society, 74, 103214. https://doi.org/10.1016/j.scs.2021.103214

pSeven. (2018, April 9). *pSeven Desktop beats MOPTA08 automotive benchmark*. pSeven Blog. https://www.pseven.io/blog/use-cases/pseven-beats-mopta08-automotive-benchmark.html

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer Verlag.

Rasmussen, C. E., & Williams, C. K. I. (2005). *Gaussian processes for machine learning*. MIT Press.

RFEM6 API. Accessed April 2025: https://www.dlubal.com/en/downloads-and-information/documents/online-manuals/rfem-6-rstab-9-webservice-api

Brochu, E., Cora, V. M., & De Freitas, N. (2010). A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1012.2599

Ament, S., S. Daulton, D. Eriksson, M. Balandat, and E. Bakshy (2024). "Unexpected Improvements to Expected Improvement for Bayesian Optimization" (cit. on pp. 76, 82, 91, 130).

Coelho, R. (2025). Composite Bayesian Optimisation for Data-Driven Design and Analysis of Polycrystalline Materials. Phd thesis. University of Porto

Debney, P. (2021). Computational engineering. The Institution of Structural Engineers. ISBN 978-1-906335-45-8

Appendix A: Literature review of optimization algorithms

Table 14: Papers on metaheuristic optimization algorithms used for structural optimization problems.

Paper	Central Theme	Main concept	Theories & Methods	Results	Most important insight	Pros	Cons
(Wei et al. , 2011)	Truss optimization on shape and sizing with frequency constraints based on parallel genetic algorithm	The paper addresses the challenges in truss shape and sizing optimization under frequency constraints, which can lead to convergence issues and complex sensitivity analyses. It proposes a Niche Hybrid Parallel Genetic Algorithm (NHPGA) as an effective solution to these problems.	The NHPGA combines the strengths of parallel computing, simplex search, and genetic algorithms with niche techniques to improve computational efficiency and solution quality. Unlike traditional methods, this approach uses global probabilistic population search and avoids reliance on gradient information.	The NHPGA significantly reduces computational time and enhances the quality of solutions in truss optimization examples. It demonstrates the potential to effectively integrate genetic algorithm capabilities, simplex search exploitation, and the computational speedup offered by parallel computing.	The algorithm's flexibility and adaptability allow it to address various structural optimization challenges with minimal prior knowledge, making it a promising architecture for high-performance parallel genetic algorithms.	During the exploration, simplex synchronous searches first are applied to the potential niches. Then another sim-plex search is performed asynchronously for quickly discovering the global optimum in the located promising zones. The NHPGA performs the computationally expensive operation steps in parallel which reduces the total computational time of the method.	The master process and slave processes in NHPGA need to wait each other in synchronous parallel stage, and this will have impact on the efficiency of NHPGA. Asynchronous parallel genetic algorithm development is needed to avoid the waiting process. The fitness function evaluation involved finite-element analysis usually consumes more than 95% of total computational time. Better constraints handling methods are needed.
(Tang et al., 2005)	Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables	The paper introduces an enhanced genetic algorithm (GA) designed to minimize the weight of truss structures by optimizing sizing, shape, and topology variables, utilizing a combination of discrete and continuous variables.	The algorithm employs mixed coding schemes such as binary with float and integer with float coding. A surrogate function is used to consolidate constraints into a single form, and surrogate reproduction is utilized to select candidates for the mating pool based on constraint satisfaction and fitness. A novel strategy a competition between parent and offspring populations, based on their constraint adherence and fitness, enhancing gene longevity.	Through various examples, the improved GA demonstrates feasible and effective results, showing significant advancements in numerical outcomes.	This paper introduces a novel approach for generating the next population by having parent and offspring populations compete based on their constraint adherence and fitness values, thereby extending the lifespan of superior genes.	Integer and mixed coding is used to represent the different variable types for topology, size and shape optimization. A surrogate function is applied to impose a penalty on all constraints. A surrogate reproduction, which considers both fitness value and degree of constraint violation, keeps the potential gene and keeps selective pressure.	Large scale problems should be analyzed to evaluate the performance of the algorithm for higher dimensions.

Table 14: Papers on metaheuristic optimization algorithms used for structural optimization problems. *(Continued)*

Paper	Central Theme	Main concept	Theories & Methods	Results	Most important insight	Pros	Cons
(Rajasekaran, 2001)	Optimization of Large Scale Three Dimensional Reticulated Structures Using Cellular Genetics and Neural Networks	The paper introduces the Cellular Genetic Algorithm (CGA), a novel optimization method combining Cellular Automata (CA) and Genetic Algorithm (GA) for optimizing large space structures by treating member areas as discrete variables.	The CGA utilizes a structural analysis package (such as FEAST, ANSYS, or SAP) or neural networks to define objective functions for CA cells. Neural networks are trained with data from these packages to expedite the typically time-consuming analysis. A multilevel optimization strategy is applied to progressively narrow the search space for design variables.	Numerical tests reveal the computational efficiency of the CGA, especially when combined with neural networks, making it particularly effective for large-scale optimization challenges. The study also examines the non-linear load deflection characteristics of optimized structures, highlighting the method's practical advantages.	Multilevel optimization approach is implemented by reducing the size of the search space for individual design variables in each successive level of optimization process including the non-linear load deflection behaviour.	It effectively manages discrete variables, which is crucial for optimizing the areas of members in space structures. Efficient with large scale problems.	The integration of Cellular Automata, Genetic Algorithms, and neural networks may complicate the implementation and require careful tuning and understanding. Initial data training data is required for the neural networks.
(Gholizadeh, 2013)	Layout optimization of truss structures by hybridizing cellular automata and particle swarm optimization	The paper introduces an efficient hybrid optimization algorithm called Sequential Cellular Particle Swarm Optimization (SCPSO) for the layout optimization of truss structures. The algorithm combines the strengths of cellular automata (CA) and particle swarm optimization (PSO) to enhance performance.	SCPSO integrates a CA-based mechanism into the PSO framework by using it for velocity updating of the particles. This integration occurs within the context of sequential unconstrained minimization techniques. The hybrid approach leverages the collective computation strengths of CA and the global search capabilities of PSO to optimize truss layouts more effectively.	The numerical results show that SCPSO achieves superior solutions and faster convergence rates compared to other optimization algorithms. This demonstrates the effectiveness of the hybrid approach in improving both the solution quality and computational efficiency for truss structure optimization.	The proposed Sequential Cellular Particle Swarm Optimization (SCPSO) algorithm effectively balances exploration and exploitation through its novel CA-based PSO scheme and sequential framework.	In order to evaluate the efficiency of the SCPSO algorithm, 4 classical layout optimization problems of truss structures are solved. SCPSO is a powerful optimization algorithm with high global search ability at low computational cost and fast convergence rate.	The paper used the exterior penalty function method (EPFM) for handling design constraints which can be computationally slow. The areas of the structural elements are optimized based on a selected discrete set.

Table 14: Papers on metaheuristic optimization algorithms used for structural optimization problems. *(Continued)*

Paper	Central Theme	Main concept	Theories & Methods	Results	Most important insight	Pros	Cons
(Gholizadeh, 2013)	Layout optimization of truss structures by hybridizing cellular automata and particle swarm optimization	The paper introduces an efficient hybrid optimization algorithm called Sequential Cellular Particle Swarm Optimization (SCPSO) for the layout optimization of truss structures. The algorithm combines the strengths of cellular automata (CA) and particle swarm optimization (PSO) to enhance performance.	SCPSO integrates a CA-based mechanism into the PSO framework by using it for velocity updating of the particles. This integration occurs within the context of sequential unconstrained minimization techniques. The hybrid approach leverages the collective computation strengths of CA and the global search capabilities of PSO to optimize truss layouts more effectively.	The numerical results show that SCPSO achieves superior solutions and faster convergence rates compared to other optimization algorithms. This demonstrates the effectiveness of the hybrid approach in improving both the solution quality and computational efficiency for truss structure optimization.	The proposed Sequential Cellular Particle Swarm Optimization (SCPSO) algorithm effectively balances exploration and exploitation through its novel CA-based PSO scheme and sequential framework.	In order to evaluate the efficiency of the SCPSO algorithm, 4 classical layout optimization problems of truss structures are solved. SCPSO is a powerful optimization algorithm with high global search ability at low computational cost and fast convergence rate.	The paper used the exterior penalty function method (EPFM) for handling design constraints which can be computationally slow. The areas of the structural elements are optimized based on a selected discrete set.
(Mortavazi et al., 2016)	Simultaneous size, shape, and topology optimization of truss structures using integrated particle swarm optimizer	The study focuses on minimizing the weight of truss structures by simultaneously optimizing their shape, size, and topology. To achieve this, an algorithm called the integrated particle swarm optimizer (iPSO) is introduced as an effective optimization tool.	The iPSO method enhances the standard particle swarm optimizer (PSO) by incorporating the concept of 'weighted particles' to boost performance. Additionally, an 'improved fly-back' technique is used to effectively manage problem constraints. These innovations aim to streamline the optimization process while maintaining efficiency and effectiveness.	The iPSO methodology was tested on various benchmark problems and demonstrated competitive results when compared to existing techniques for truss structure optimization. Its formulation is noted for its simplicity, making it an appealing option due to its effective performance and ease of implementation.	The most important insight from this investigation is that the integrated Particle Swarm Optimizer (iPSO) is an effective and versatile algorithm for optimizing truss structures, capable of handling discrete, continuous, and combined search spaces, as well as multiple load conditions and design constraints.	Simplicity of the method, because it builds upon the standard PSO, weighted particle and improved fly-back technique.	Genetic Algorithms converge faster for certain benchmark problems.

Table 14: Papers on metaheuristic optimization algorithms used for structural optimization problems. *(Continued)*

Paper	Central Theme	Main concept	Theories & Methods	Results	Most important insight	Pros	Cons
(Gomes, 2011)	Truss optimization with dynamic constraints using a particle swarm algorithm	The paper explores the use of Particle Swarm Optimization (PSO) for optimizing the mass of structural trusses in terms of size and shape, while considering frequency constraints. Due to the inherent non-linear and dynamic nature of these optimization problems, the study focuses on the PSO algorithm as a suitable approach.	The PSO algorithm is chosen for its proven effectiveness as a global optimizer in various fields and its capability to handle non-linear problems without relying on gradient-based methods. The paper briefly reviews the key features of PSO that make it suitable for addressing such optimization challenges.	Through four benchmark examples of truss optimization with shape and size frequency constraints, it was demonstrated that the PSO algorithm performed comparably to other methods, and in certain cases, it outperformed them. These results highlight the potential of PSO in effectively solving complex structural optimization problems.	The algorithm does not require the gradients of the objective function, but only the objective function itself which enables the method to deal with symmetrical trusses without any modifications.	The algorithm used in the paper has lower number of parameters necessary and is able to optimize in the continuous design space of the variables. It works with population and random parameters that balance between exploration and exploitation to escape local minima/maxima in the optimization process.	The constraints were implemented with penalty functions.
(Luh & Lin, 2011)	Optimal design of truss-structures using particle swarm optimization	The study applies a two-stage Particle Swarm Optimization (PSO) approach to optimize truss structures with the aim of minimizing weight while considering constraints like stress, deflection, and kinematic stability.	Initially, the truss topology is optimized using a modified Binary Particle Swarm Optimization (BPSO). Following this, the size and shape of the truss members are further refined using the Attractive and Repulsive Particle Swarm Optimization (ARPSO) technique.	The methodology was tested on a two-tier, 39-member, 12-node ground structure and demonstrated the capability to identify truss structures that are more optimal than those previously documented in the literature.	Two-stage PSO based optimization scheme for truss-structure is developed. Better truss structures are found in less calculation time compared to one-stage genetic algorithm and two-stage ant algorithms.	Introduces more diversity in the optimized shapes due the first BPSO algorithm that optimized the topology of the algorithm. Symmetry about the central vertical axis is employed to reduce the number of variables.	Relatively large number of calculations depending on the case (ca. 260000).

Table 14: Papers on metaheuristic optimization algorithms used for structural optimization problems. *(Continued)*

Paper	Central Theme	Main concept	Theories & Methods	Results	Most important insight	Pros	Cons
(Li et al., 2007)	A heuristic particle swarm optimizer for optimization of pin connected structures	The paper introduces a heuristic particle swarm optimizer (HPSO) designed to optimize the design of pin-connected structures by integrating elements of particle swarm optimization and harmony search.	The HPSO algorithm incorporates the particle swarm optimizer with passive congregation (PSOPC) alongside a harmony search scheme. It employs a 'fly-back mechanism' to address problem-specific constraints and utilizes the harmony search for managing variable constraints.	The effectiveness of the HPSO algorithm was validated against PSO and PSOPC algorithms across five planar and spatial truss design cases. The findings demonstrate that HPSO significantly improves convergence rates and achieves optimal designs more swiftly than the other algorithms.	The HPSO algorithm manages variable constraints through a combination of the harmony search approach and the 'fly-back mechanism' for addressing problem-specific constraints. Unlike the PSO and PSOPC algorithms, HPSO ensures that particles remain within the variable boundaries, fully utilizing the particle's movement capabilities throughout the optimization process.	HPSO algorithm converge more quickly than the PSO and the PSOPC algorithms.	Convergence rate of the HPSO algorithm will slow down when the number of iterations increase.
(Camp & Bichon, 2004)	Design of Space Trusses Using Ant Colony Optimization	The paper introduces a design procedure using Ant Colony Optimization (ACO) for the discrete optimization of space trusses. The primary aim is to minimize the total weight (or cost) of the structure while adhering to material and performance constraints such as stress and deflection limits.	The design translates the optimization of space trusses into a modified Traveling Salesman Problem (TSP), where the TSP network mirrors the structural topology, and the TSP tour length represents the structure's weight. The ACO algorithm is employed for this purpose, utilizing discrete variables, a flexible constraint format, and a penalty function to ensure constraints are met, accommodating multiple loading scenarios.	The study compares the ACO-based truss design method with designs achieved through a genetic algorithm and classical continuous optimization techniques, illustrating the efficacy of the ACO procedure in optimizing truss structures under the specified constraints	While both ACO and Genetic Algorithms (GAs) utilize a population of agents to represent solutions, ACO maintains information through artificial pheromone trails, capturing the memory of the entire colony over generations. In contrast, GAs focus on information from just the current generation. This allows ACO to develop solutions in each new search cycle using the accumulated collective information, providing an advantage over GAs.	ACO for space trusses is less influenced by poor initial solutions compared to the genetic algorithms (GAs).ACO circumvents this issue by integrating information gathered by the colony with a nearest-neighbour heuristic, which prioritizes the shortest path or, in this case, the smallest cross-sectional area. The reliability of the method is good less than 1% on average for all examples.	The algorithm focuses on optimization in the discrete design space instead of the continuous design space which might lead to suboptimal results depending on the selected discrete sets.

Table 14: Papers on metaheuristic optimization algorithms used for structural optimization problems. *(Continued)*

Paper	Central Theme	Main concept	Theories & Methods	Results	Most important insight	Pros	Cons
(Aydoğdu & Saka, 2011)	Ant colony optimization of irregular steel frames including elemental warping effect	The paper addresses the significant impact of warping on the design of steel space frames with thin-walled steel sections. It aims to optimize the design of these frames by considering warping effects according to the LRFD-AISC guidelines.	The study utilizes the Ant Colony Optimization (ACO) technique to solve the design problem, taking into account the provision for warping in the optimization process.	Several space frame examples are optimized using the developed algorithm, effectively illustrating the influence of warping on achieving optimal designs.	Warping causes a significant amount of increase in the minimum weight of the designs for symmetrical and unsymmetrical space frames alike.	Sensitivity analysis is performed to select the appropriate values for the parameters of the ant colony optimization technique.	The problem is solved in the discrete design space considering a list of predefined cross-sections.
(Degertekin, 2012)	Improved harmony search algorithms for sizing optimization of truss structures	The paper focuses on improving the Harmony Search (HS) algorithm, which is inspired by music improvisation, to enhance its application in the optimization of truss structures. The original HS is known for its sensitivity to tuning parameters, prompting the development of new variants to lessen this dependency.	Two enhanced variants, Efficient Harmony Search (EHS) and Self-Adaptive Harmony Search (SAHS), are introduced for the sizing optimization of truss structures. These algorithms aim to maintain robustness while mitigating the HS's reliance on parameter tuning.	The performance of EHS and SAHS is evaluated through four classical truss structure weight minimization problems. The study compares their results with those of the standard HS and other recent meta-heuristic algorithms, demonstrating the robustness and improved efficiency of the proposed variants.	Improved designs can be achieved by progressively decreasing the pitch adjustment parameter as the optimization process advances. Constraints have to be normalized. EHS and SAHS dynamically update their internal parameters during the search process.	Both HS methods presented in this paper possess the inherent ability of converging to a nearly global optimum design due to low standard deviation for the optimized weight.	Computational cost of EHS and SAHS in terms of structural analyses is significantly higher when compared to other meta-heuristic algorithms.
(Cheng et al., 2016)	A Hybrid Harmony Search algorithm for discrete sizing optimization of truss structure	The paper introduces a novel variant of the Harmony Search (HS) algorithm, called the Hybrid Harmony Search (HHS) algorithm, designed to improve optimization outcomes by integrating different search strategies.	The HHS algorithm retains the Harmony Memory and pitch adjustment functions of the original HS algorithm but replaces its randomization function with Global-best Particle Swarm Optimization (PSO) and neighbourhood search techniques.	The performance of the HHS algorithm is evaluated on six discrete truss structure optimization problems across various loading conditions. The results indicate that the HHS algorithm excels in delivering optimal solutions.	The HHS algorithm uses memory consideration and pitch adjustment to focus on global search in the early stage and Global-best PSO search and neighbourhood search to focus on local search in the later stage.	The HHS algorithm effectively achieves an ideal balance between exploration and exploitation while also reaching the optimal solution much more quickly than other methods tested.	Analysis of HHS effectiveness in dealing with large-scale optimization problems in high-dimensional feature space is needed.

Table 14: Papers on metaheuristic optimization algorithms used for structural optimization problems. *(Continued)*

Paper	Central Theme	Main concept	Theories & Methods	Results	Most important insight	Pros	Cons
(Miguel & Miguel, 2012)	Shape and size optimization of truss structures considering dynamic constraints through modern metaheuristic algorithms	The paper focuses on addressing the complex issue of mass optimization in truss structures, considering both shape and sizing under multiple natural frequency constraints, using novel metaheuristic algorithms.	It leverages Harmony Search (HS) and Firefly Algorithm (FA), both of which are non-gradient-based methods, to circumvent the challenges typically associated with dynamic sensitivity analysis and the convergence issues of traditional gradient-based approaches.	The application of these algorithms to four benchmark problems demonstrated that both HS and FA delivered superior results within a relatively short computational time in three cases and performed on par with the best solutions found in literature for the fourth case, highlighting their robust optimization capabilities.	HS finds optimal solutions quicker, while FA tends to achieve slightly better solution quality, highlighting their respective strengths in solving complex, nonlinear optimization problems.	HS and FA demonstrate superior or comparable results to existing methods, and they perform efficiently without the need for extensive parameter tuning.	In 1 out of 4 cases the HS and FA reached a reached worse result that the ones available in literature.
(Pham, 2016)	Truss optimization with frequency constraints using enhanced differential evolution based on adaptive directional mutation and nearest neighbour comparison	The article introduces a novel differential evolution algorithm, ANDE, designed to address the challenging problem of truss optimization with dynamic frequency constraints, specifically focusing on optimizing shape and size.	ANDE incorporates three modifications to the conventional differential evolution approach: an adaptive p-best strategy for balancing global exploration and local exploitation, a directional mutation rule to enhance solution improvement chances, and a nearest neighbour comparison method for pre-emptively skipping unlikely solutions. These modifications streamline the process without needing additional parameter adjustments.	Testing ANDE on five benchmark examples demonstrates that the algorithm delivers good and stable results, maintaining compliance with frequency constraints. ANDE's optimal designs are generally comparable to or better than those achieved by other advanced metaheuristics, with the added advantage of requiring fewer structural analyses.	By incorporating three straightforward modifications into ANDE, it achieves a balance between global exploration and local exploitation while also reducing computational costs.	ANDE provides a gradual transition from global exploration to local exploitation during the search of the design space. It incorporates prejudgment of a solution which reduces the number of FE evaluations. No additional parameters are needed.	Application of ANDE to large-scale problems and different types of structures should be further studied.

Table 14: Papers on metaheuristic optimization algorithms used for structural optimization problems. *(Continued)*

Paper	Central Theme	Main concept	Theories & Methods	Results	Most important insight	Pros	Cons
(Ho-Huu et al., 2016)	Optimal design of truss structures with frequency constraints using improved differential evolution algorithm based on an adaptive mutation scheme	The paper discusses an improved Differential Evolution (IDE) algorithm designed for optimizing the shape and size of truss structures under frequency constraints, with enhancements primarily in the mutation and selection phases.	The IDE introduces a new scheme in the mutation phase that adaptively uses multiple popular mutation strategies ("rand/1," "rand/2," "best/1," and "best/2") to balance global exploration and local exploitation. Additionally, the selection phase incorporates an elitist selection technique to preserve better individuals for subsequent generations, thereby improving convergence rates.	Tests on five benchmark problems demonstrate the IDE's efficiency and robustness, achieving optimal designs similar to those obtained by the standard DE but with significantly reduced computational costs. The IDE outperforms DE and other known methods in terms of efficiency.	The improvements are carried on mutation and selection phases but mainly focused on the mutation phase.	The new scheme effectively maintains a balance between global exploration and local exploitation during the search process of the Differential Evolution (DE) algorithm. In almost all problems, the number of structural analyses of the IDE is only approximately a haft of that of the original DE.	The stability of the IDE method requires further improvements. Further work can investigate the performance of IDE for other types of structures such as shell structures.
(Nguyen-Van et al., 2021)	A novel hybrid differential evolution and symbiotic organisms search algorithm for size and shape optimization of truss structures under multiple frequency constraints	The article introduces a novel optimization algorithm called Hybrid Differential-SOS (HDS), which combines elements of Differential Evolution (DE) and Symbiotic Organisms Search (SOS) to enhance the quality of solutions and speed of convergence in the optimization of truss structures with multiple frequency constraints.	The HDS algorithm leverages newly developed operators from DE and SOS to boost both global and local search capabilities. It incorporates an automatically adapted parameter for balancing these search aspects.	The algorithm's effectiveness is validated through the examination of 26 benchmark mathematical functions and 5 numerical examples of truss structure optimization. The findings reveal that HDS achieves high-quality optimal solutions with less computational effort compared to the original DE and SOS methods and other existing optimization paradigms.	The approach employs an elitist scheme in the selection phase to retain the best solutions, facilitating improved solution quality and reduced computational effort.	The results have shown that the HDS provides superior performance in terms of convergence speed and solution accuracy compared to DE, SOS, and many other methodologies. HDS is very promising in applications related to composite laminated and functionally graded beams, plates, shells and structural health monitoring.	Further study in the discrete design space for different structures might be beneficial.

Table 14: Papers on metaheuristic optimization algorithms used for structural optimization problems. *(Continued)*

Paper	Central Theme	Main concept	Theories & Methods	Results	Most important insight	Pros	Cons
(Ho-Huu, Nguyen-Thoi, et al., 2016)	An adaptive elitist differential evolution for optimization of truss structures with discrete design variables	The paper introduces an adaptive elitist differential evolution (aeDE) algorithm aimed at optimizing truss structures with discrete design variables, featuring enhancements over the traditional differential evolution (DE) method.	The aeDE algorithm incorporates three key modifications: an adaptive mutation technique that selects a mutation operator based on the deviation of the objective function among previous generations to maintain search balance, an elitist selection strategy to accelerate convergence by preserving top-performing individuals, and a rounding technique to handle discrete design variables effectively.	The efficiency and dependability of aeDE are validated through six truss structure optimization problems, showing that aeDE generally outperforms standard DE and several other methods found in the literature, offering superior solution quality and faster convergence.	The adaptive elitist differential evolution (aeDE) algorithm significantly enhances the optimization of truss structures with discrete design variables. By incorporating three key modifications—adaptive mutation for search balance, elitist selection for faster convergence, and a rounding technique for discrete variables—the aeDE reliably achieves optimal solutions more efficiently than the standard differential evolution (DE) and other methods	Its effectiveness is particularly evident in large-scale problems, and its simplicity in design allows for easy extension to other engineering optimization challenges, making it a robust and versatile tool in computational optimization.	Although it claims to be simple to implement, the introduction of adaptive techniques and elitist strategies could still require careful tuning and understanding for best performance in diverse scenarios.

Table 15: Papers on Bayesian optimization.

Paper	Central Theme	Main concept	Theories & Methods	Results	Most important insight	Pros	Cons
(Shende et al., 2021)	Bayesian topology optimization for efficient design of origami folding structures	The paper explores the use of Bayesian optimization (BO) for solving structural optimization problems, specifically focusing on origami-inspired design spaces, which are characterized by complex and non-convex design possibilities. BO is proposed as an efficient method for optimizing these complex design spaces using fewer evaluations of expensive finite element objective function.	Bayesian optimization is utilized with a Gaussian process (GP) surrogate model to mimic expensive objective function. Then it is compared to traditional optimization methods like gradient-based techniques and genetic algorithms. The study also looks into hyperparameter tuning, sensitivity to the initial training set, and proposes heuristic methods to reduce overall computational costs.	Bayesian optimization requires fewer finite element solutions compared to traditional methods, making it a promising choice for such non-convex optimization problems (presence of multiple local optima). Consistently outperforms the gradient-based method, delivering previously undiscovered designs for the structure.	Bayesian optimization is less sensitive to the initial training set than the gradient-based approach. Based on the studies conducted, the Gaussian process (GP) surrogate model works well for origami optimization problems. The squared exponential covariance function as well as the Matern kernel are able to find good solutions to the optimization problems analysed in the paper.	Good sensitivity analysis of the hyperparameter tuning and influence of the initial training set.	Design space is relatively small. The kernels are stationary and isotropic. The use of nonstationary and anisotropic kernels might lead to new designs but increase the number of hyperparameters. Nonstationary kernels are good when modelling processes in which different regions of the input space show different characteristics e.g. change in smoothness or variability. Anisotropic kernels are useful when the input dimensions have different level of impact which can be modelled by using different length scale parameters for each input dimension. Example is the Automatic Relevance Determination (ARD) kernel which takes the square exponential kernel and assigns a single length scale per input dimension. However, there is a risk of overfitting.

Table 15: Papers on Bayesian optimization. *(Continued)*

Paper	Central Theme	Main concept	Theories & Methods	Results	Most important insight	Pros	Cons
(Coelho et al., 2025)	A composite Bayesian optimization framework for material and structural design	The paper presents a design framework using Bayesian optimization to improve efficiency and quality in material and structural design processes, focusing on reducing computational costs and effectively managing uncertainty.	The framework involves an initial step of efficient design space exploration and a subsequent composite Bayesian optimization strategy to evaluate the objective function. It employs a surrogate model and techniques like Principal Component Analysis for dimensionality reduction, utilizing a dynamic, adaptive sampling strategy instead of traditional random sampling methods.	The framework shows significant improvements in performance and quality, especially in nonlinear settings, through various design scenarios. It reduces the number of required experiments and demonstrates its potential to enhance design methodologies in both material and structural engineering compared to established data-driven approaches.	This method allows the optimizer to (i) fully assess the numerical simulation's response, (ii) improve the surrogate model's predictive precision, and (iii) effortlessly incorporate the objective function's structure into the optimization framework.	The BO framework effectively integrates elements such as reparameterization into a latent response space, surrogate model selection, and gradient-enabled Monte Carlo acquisition functions. The framework's benefits are particularly notable in complex scenarios with geometric or material non-linearity, where it reduces the number of experiments needed to achieve target objectives.	The method is not applied for larger structural engineering problems such as topology, shape and size truss optimization.
(Moriconi et al., 2020)	High-dimensional Bayesian optimization using low-dimensional feature spaces	The paper addresses the challenge of scaling Bayesian Optimization (BO) for high-dimensional problems by introducing a method to optimize within a low-dimensional feature space.	The approach involves learning a low-dimensional feature space alongside joint optimization of the response surface and reconstruction mapping. This allows the optimization of BO's acquisition function in a simplified, lower-dimensional subspace. The method involves reconstructing the original parameter space from this subspace to evaluate the black-box function, while managing exploration through constrained optimization.	This method effectively reduces the complexity of high-dimensional optimization problems, enabling the use of BO with a smaller evaluation budget, although the paper does not specify experimental outcomes.	The framework enables efficient Bayesian optimization of intrinsically low-dimensional black-box functions by leveraging nonlinear embeddings through a manifold Gaussian Process (GP), which provides low-dimensional feature representations and reconstructs high-dimensional data.	The use of a nonlinear constraint based on Lipschitz continuity ensures exploration remains close to the training data, thus improving the reliability of predictions and maintaining optimization focus.	The nonlinear constraint might cause the algorithm to get stuck in local optima.

Table 15: Papers on Bayesian optimization. *(Continued)*

Paper	Central Theme	Main concept	Theories & Methods	Results	Most important insight	Pros	Cons
(Mathern et al., 2021)	Multi-objective constrained Bayesian optimization for structural design	The study aims to address the complexity in planning and designing concrete structures by applying a Bayesian optimization framework to exploit multi-objective strategies, balancing sustainability, buildability, and performance within the constraints of structural design.	The work develops a Bayesian optimization approach to tackle expensive, constrained structural design problems. This approach evaluates trade-offs between cost-effective objectives and expensive constraints, benchmarking its effectiveness against the Non-dominated Sorting Genetic Algorithm II (NSGA-II) and random search methods, focusing on a reinforced concrete beam's design.	The Bayesian optimization framework demonstrated superior performance over NSGA-II and random search by showing improved rates of advancement, higher solution quality, and reduced variance in outcomes, indicating its suitability for multi-objective constrained optimization issues in structural design.	Unlike earlier methods, this work's Bayesian algorithm explicitly leverages the fact that evaluating objective functions is inexpensive, while constraint functions involve costly evaluations due to the extensive numerical computations typical in structural engineering design challenges. Therefore, the constraints are modelled using Gaussian Processes (GPs) which significantly reduce the total number of finite element evaluations by sampling strategic points where the expected improvement is maximum.	Design constraints were established to guarantee that the beam's configuration was feasible for construction and met the necessary bending and shear capacity as outlined by structural design codes.	The sensitivity of Bayesian optimization framework to higher dimensional problems should be explored further. They are sensitive to the curse-of-dimensionality. How the algorithm performs with categorical variables should be examined.

Appendix B: 1D optimization problem: Cantilever Beam

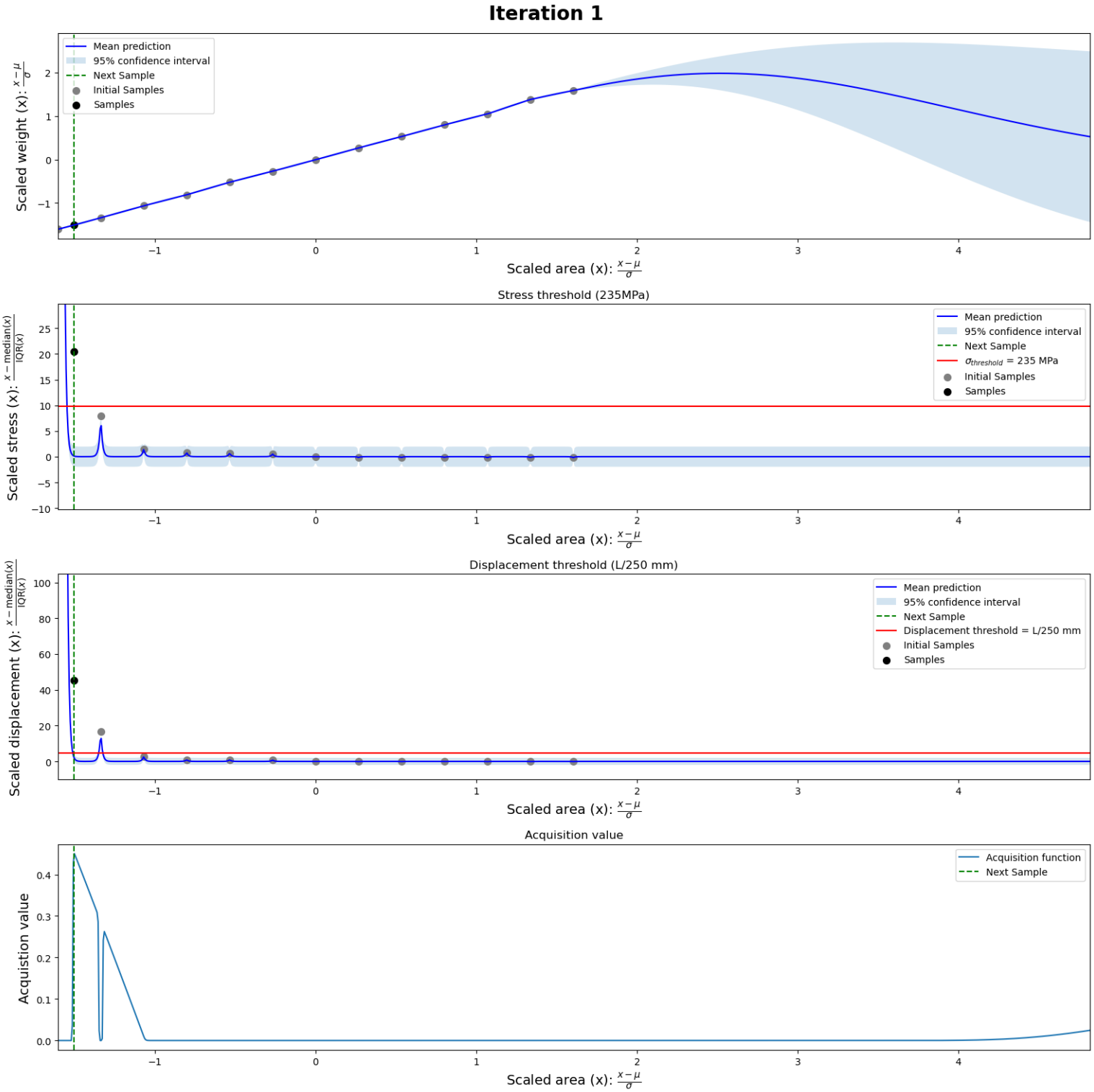


Figure 106: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 1.

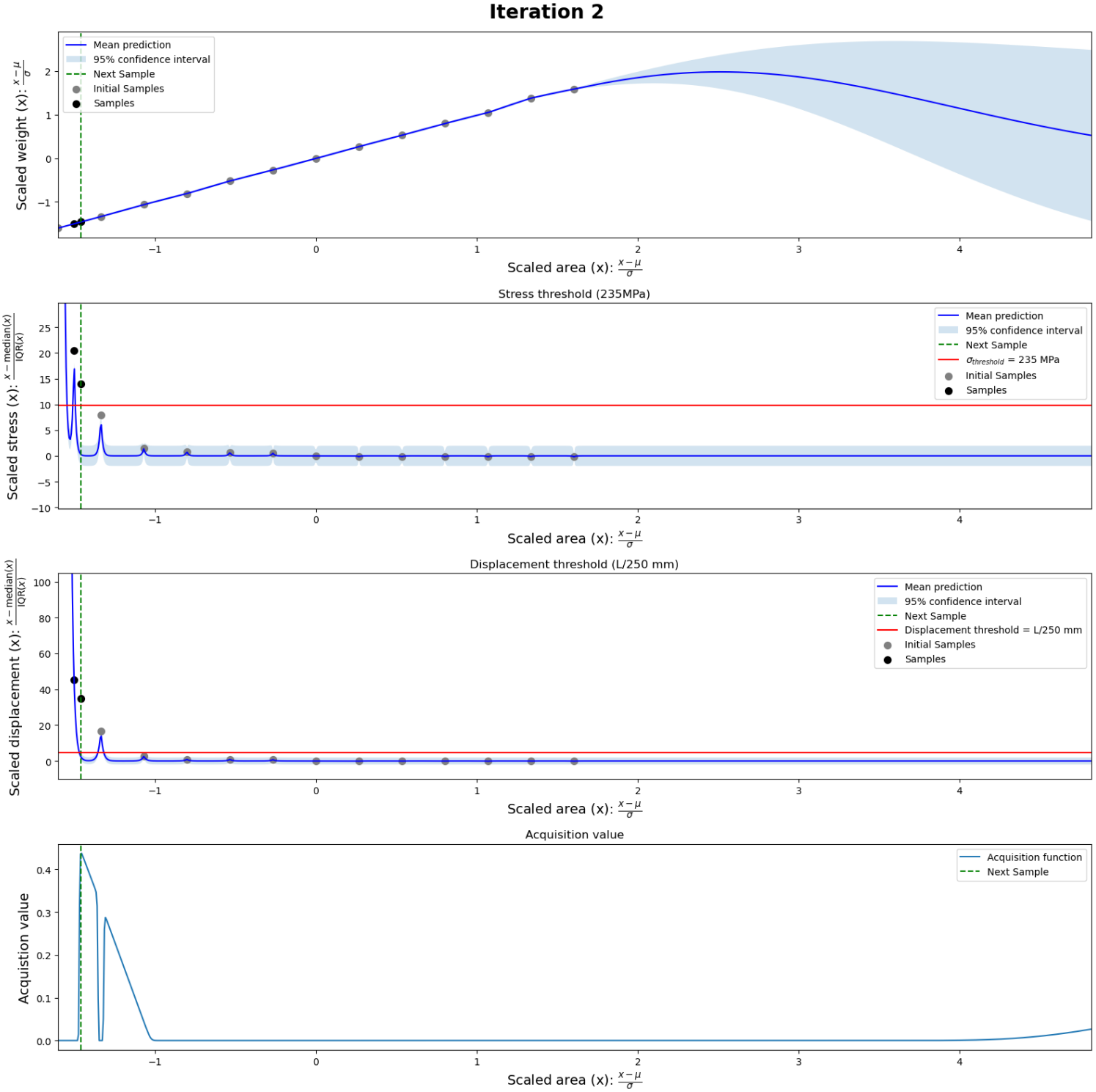


Figure 107: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 2.

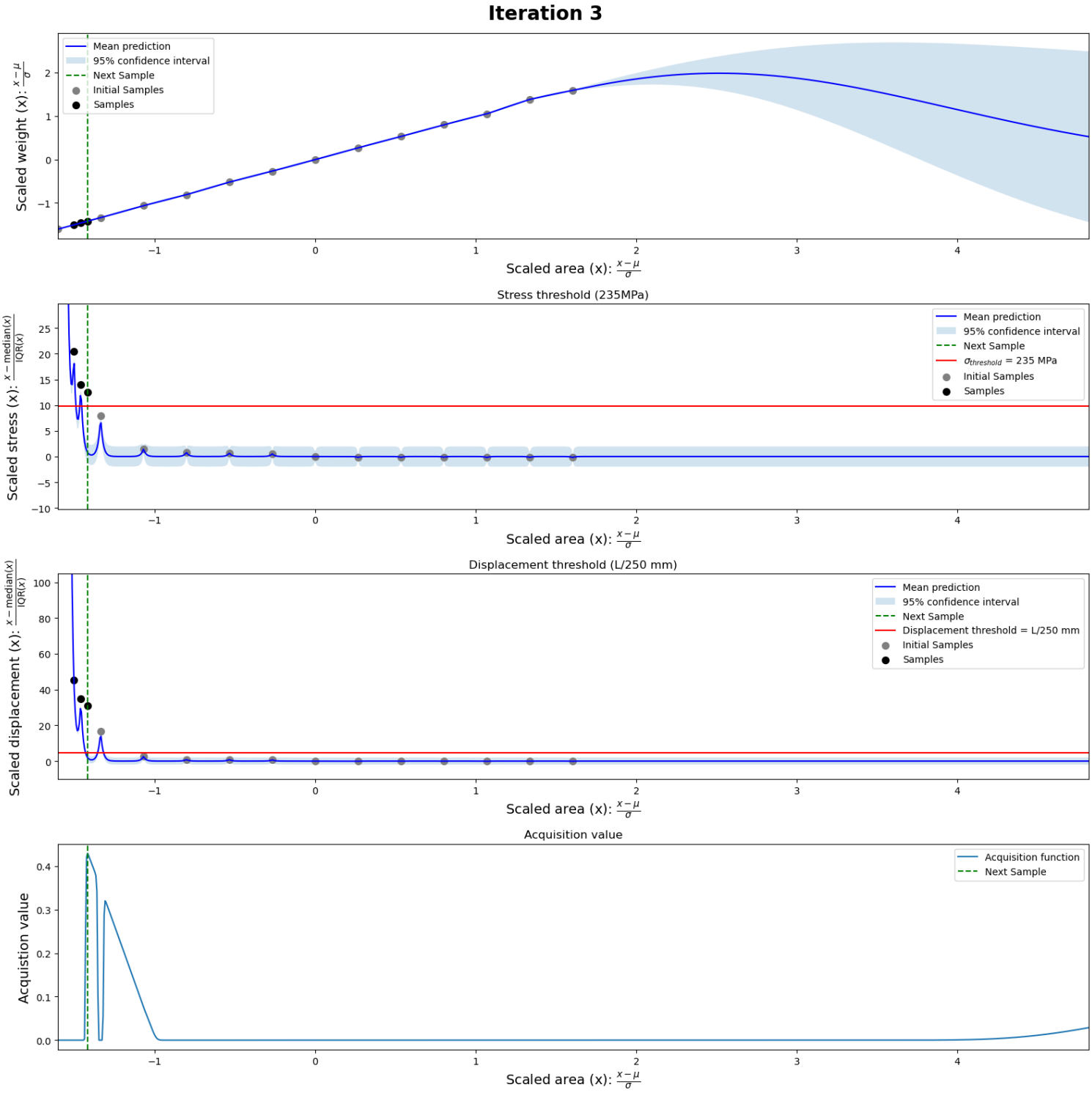


Figure 108: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 3.

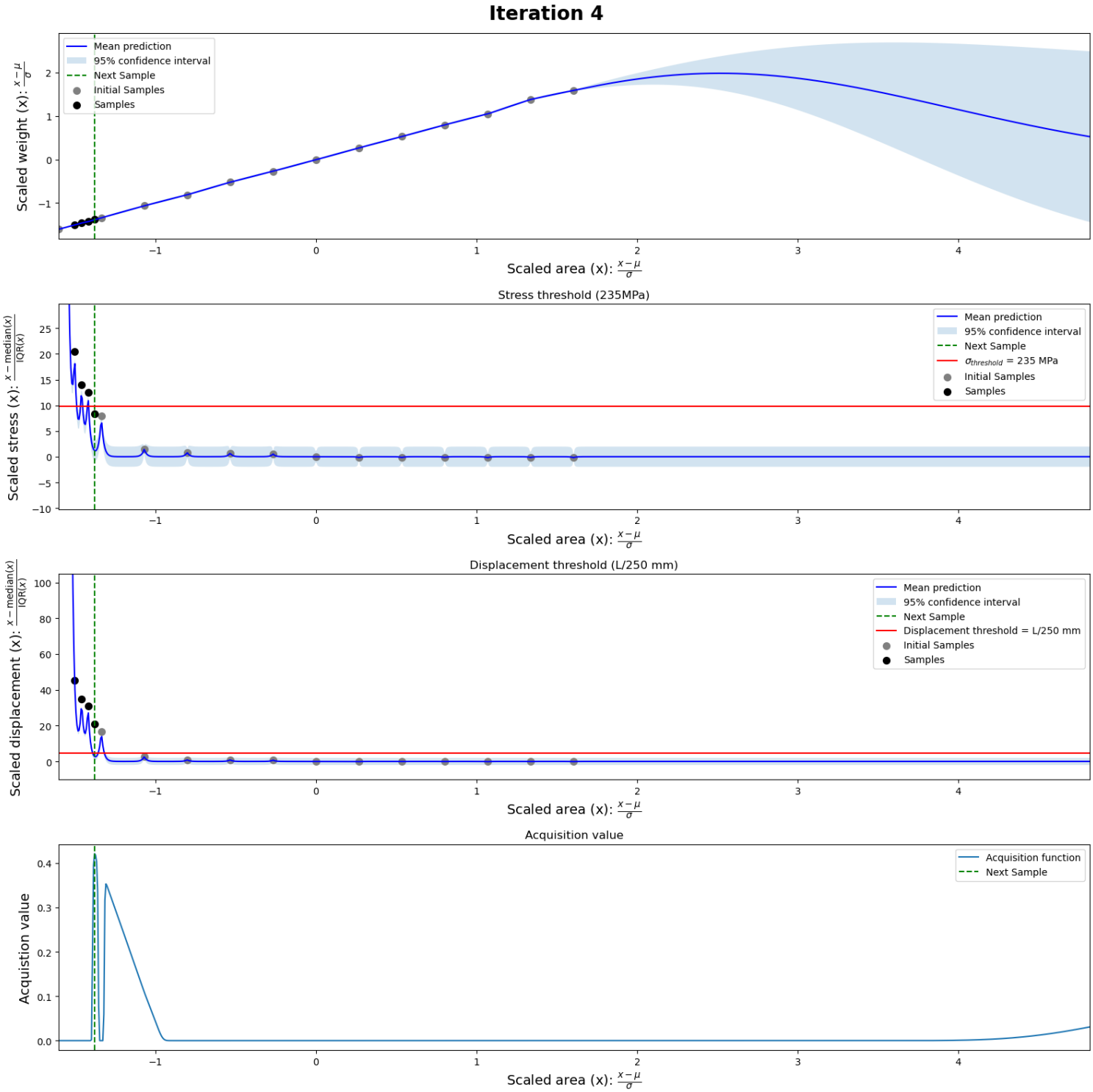


Figure 109: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 4.

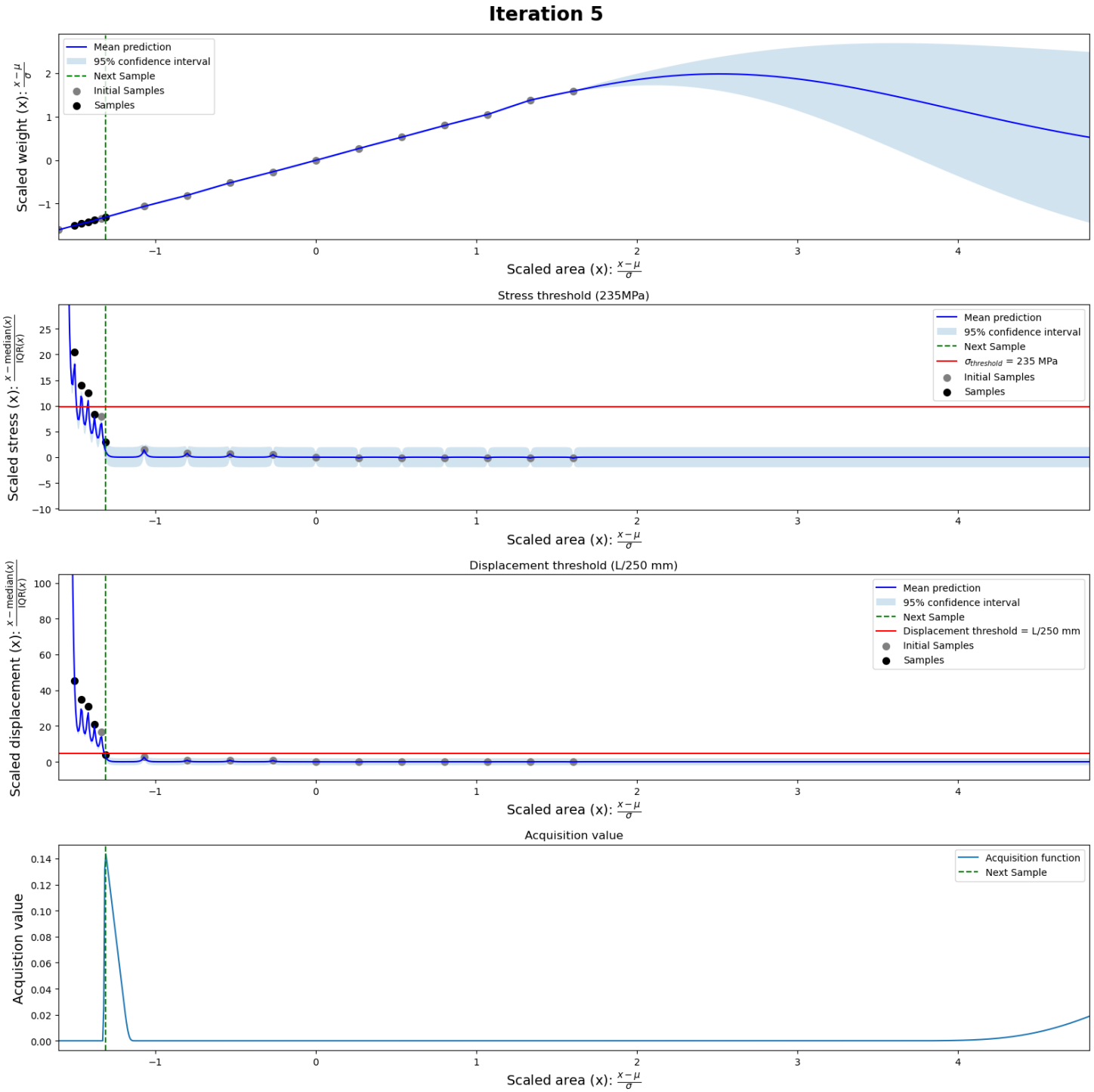


Figure 110: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 5.

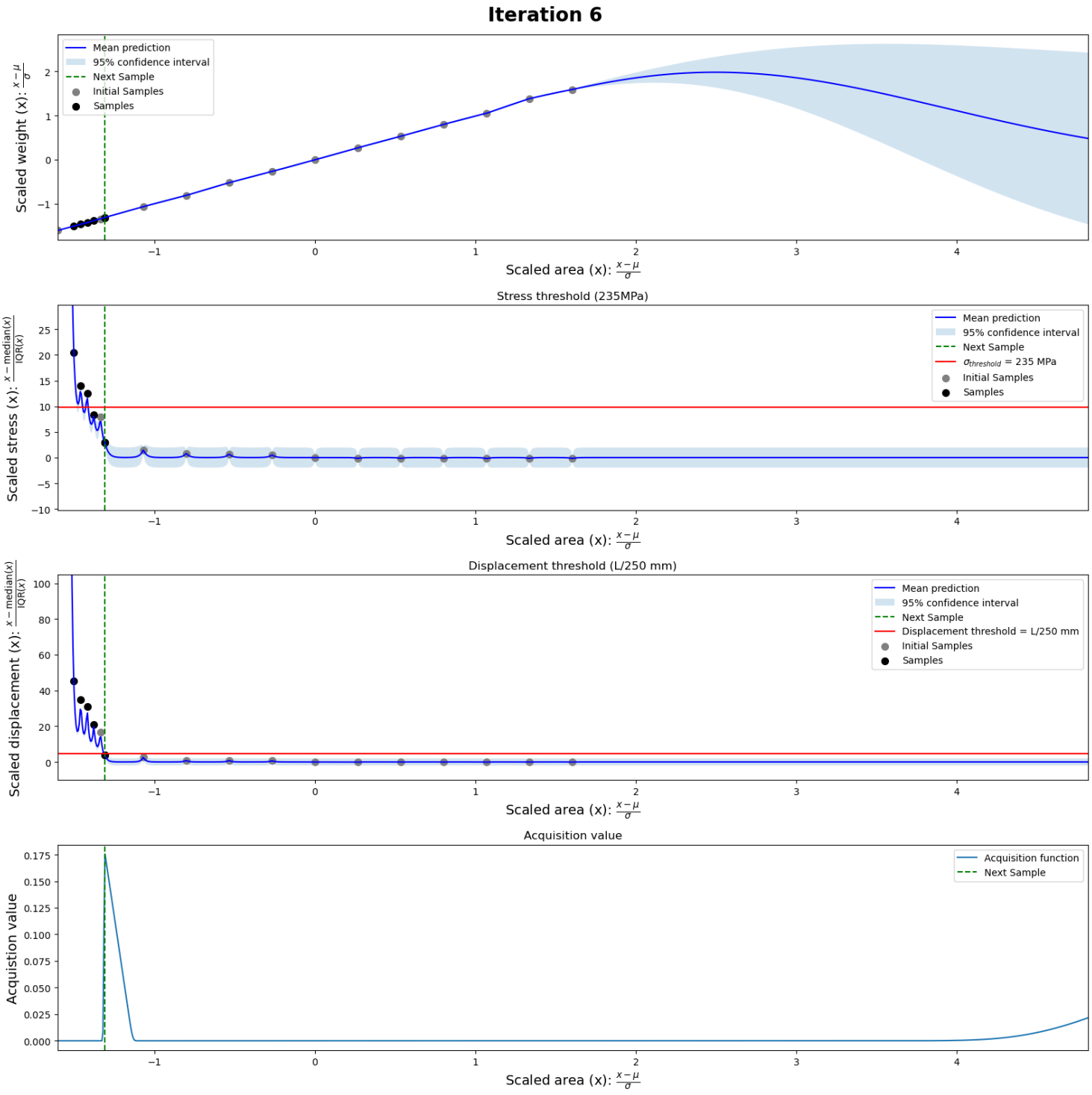


Figure 111: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 6.

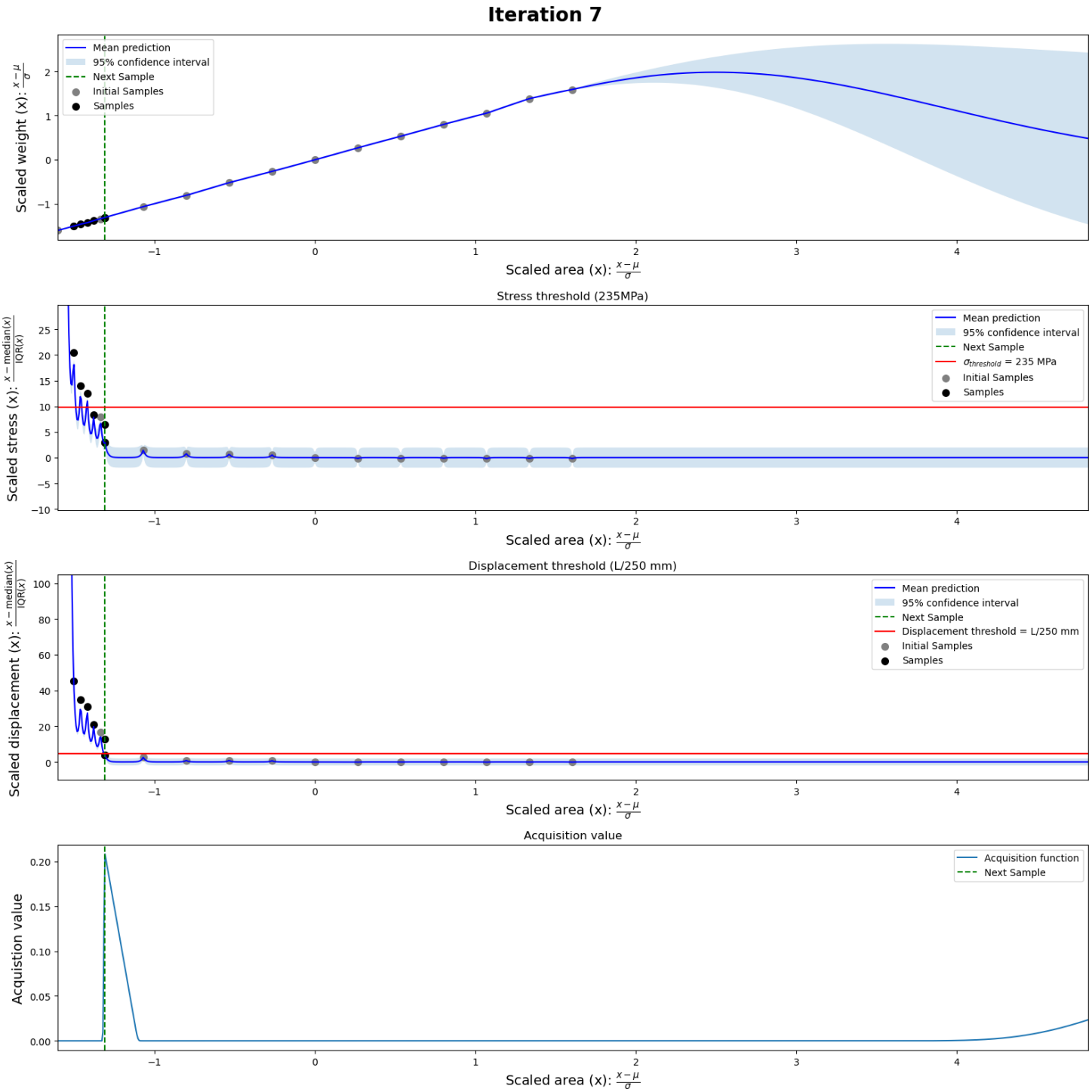


Figure 112: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 7.

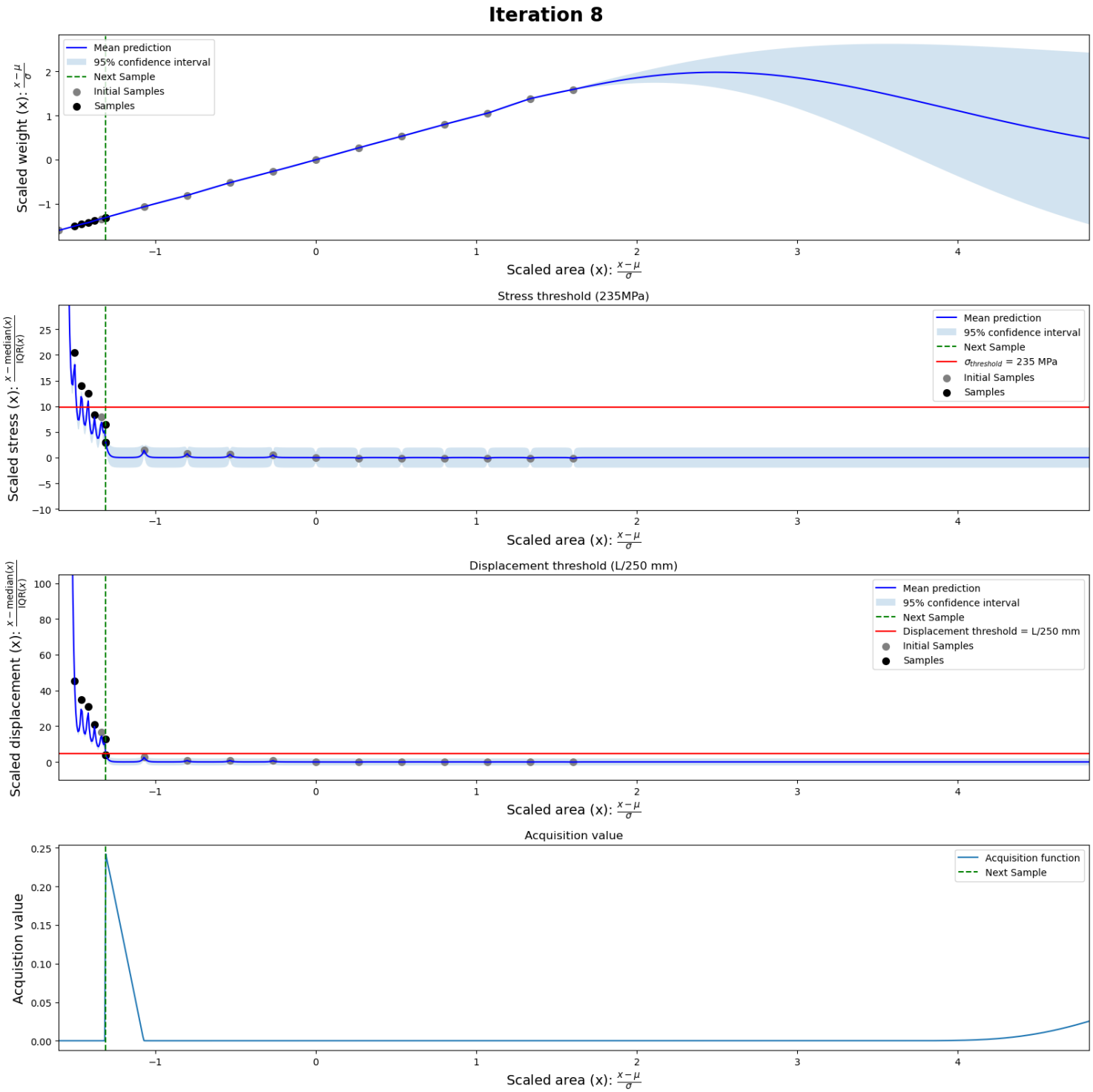


Figure 113: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 8.

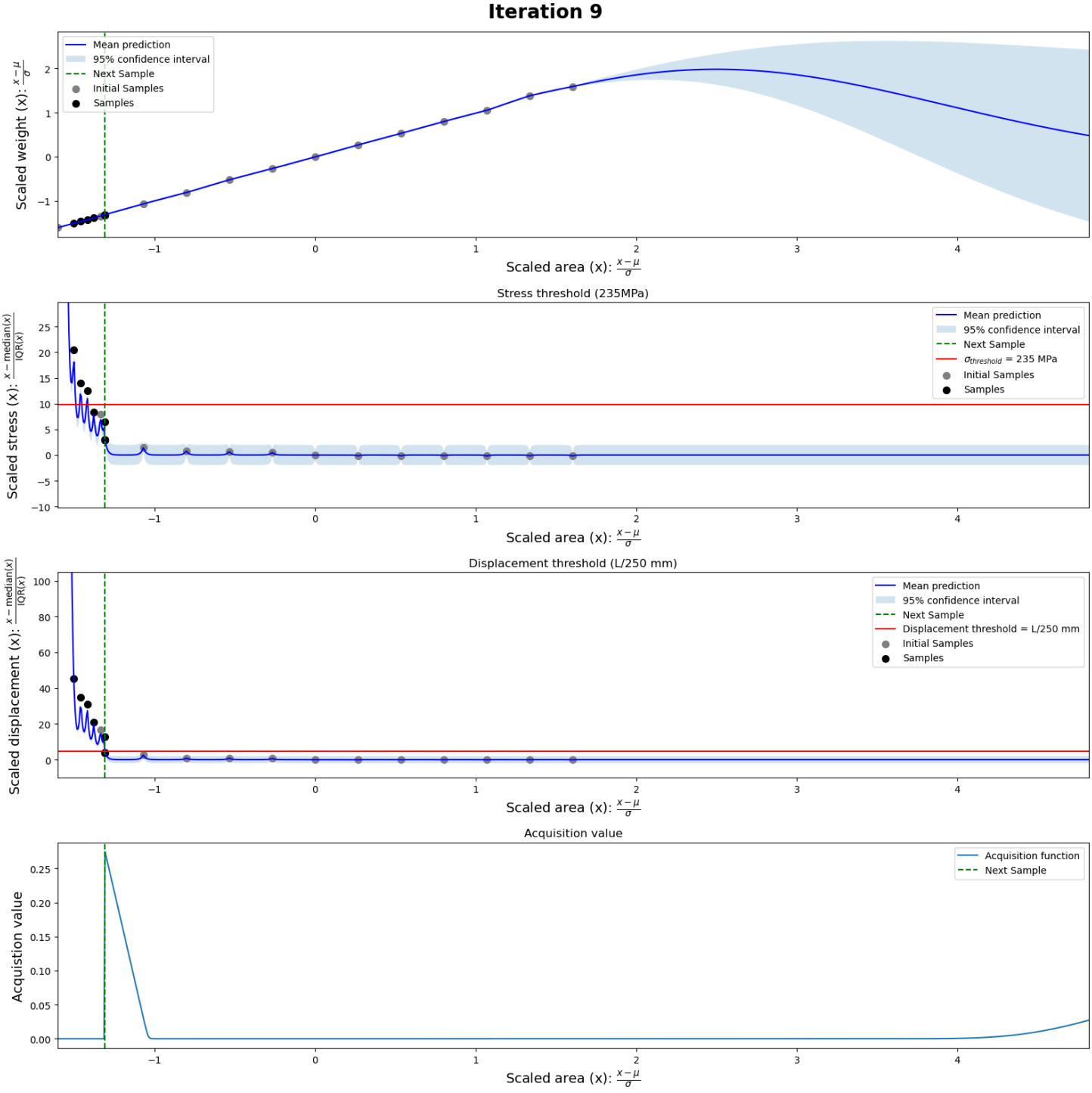


Figure 114: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 9.

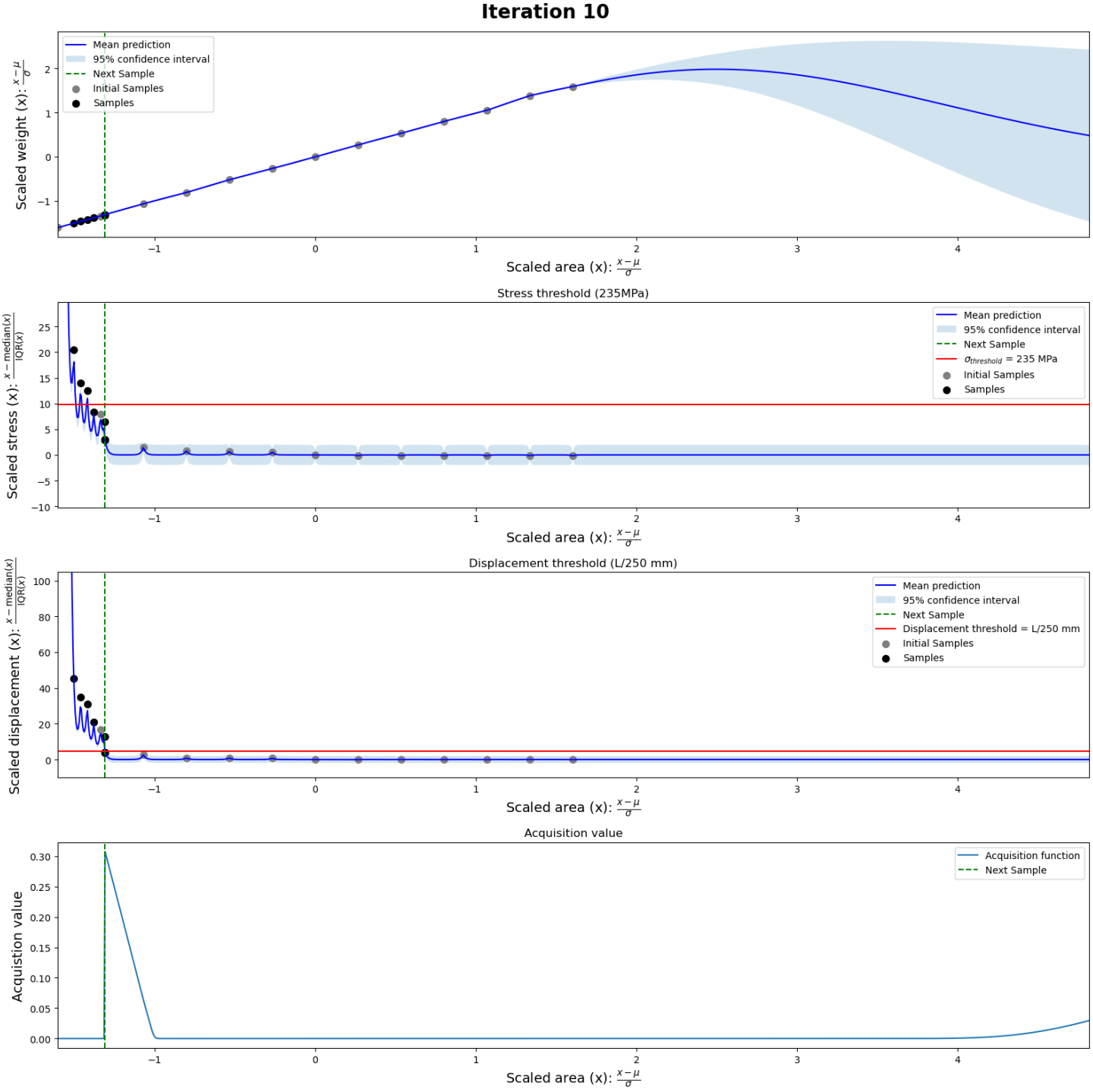


Figure 115: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 10.

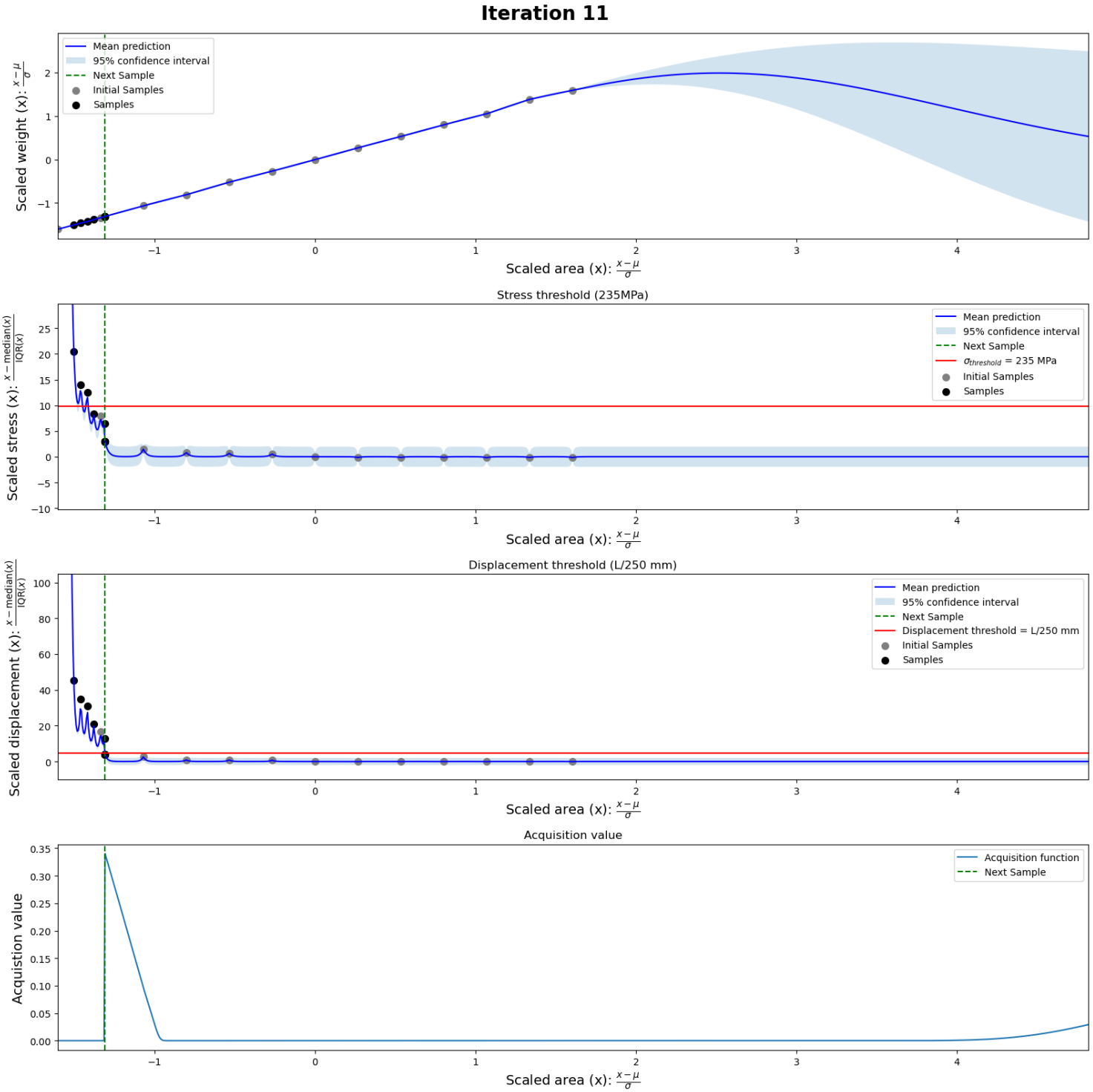


Figure 116: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 11.

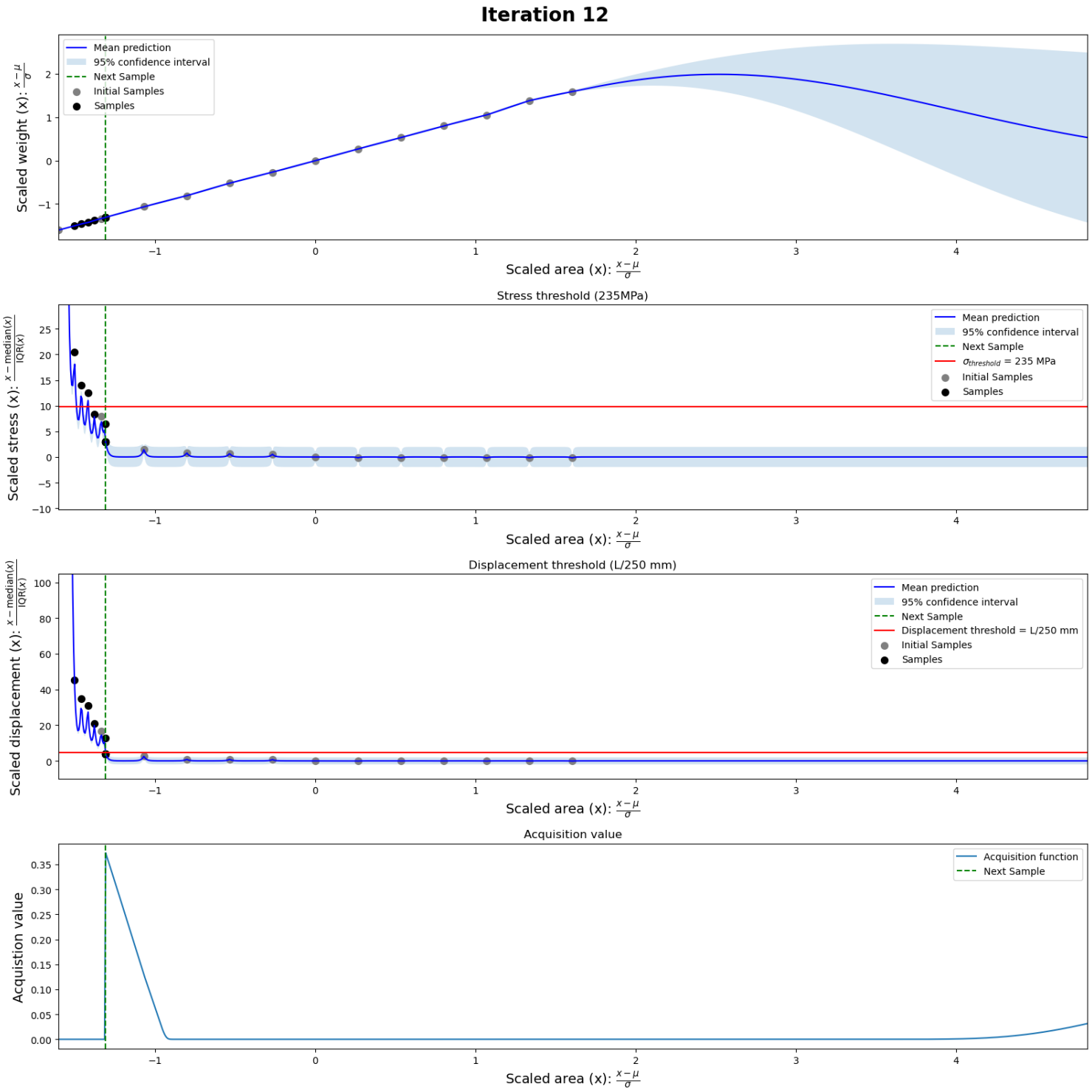


Figure 117: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 12.

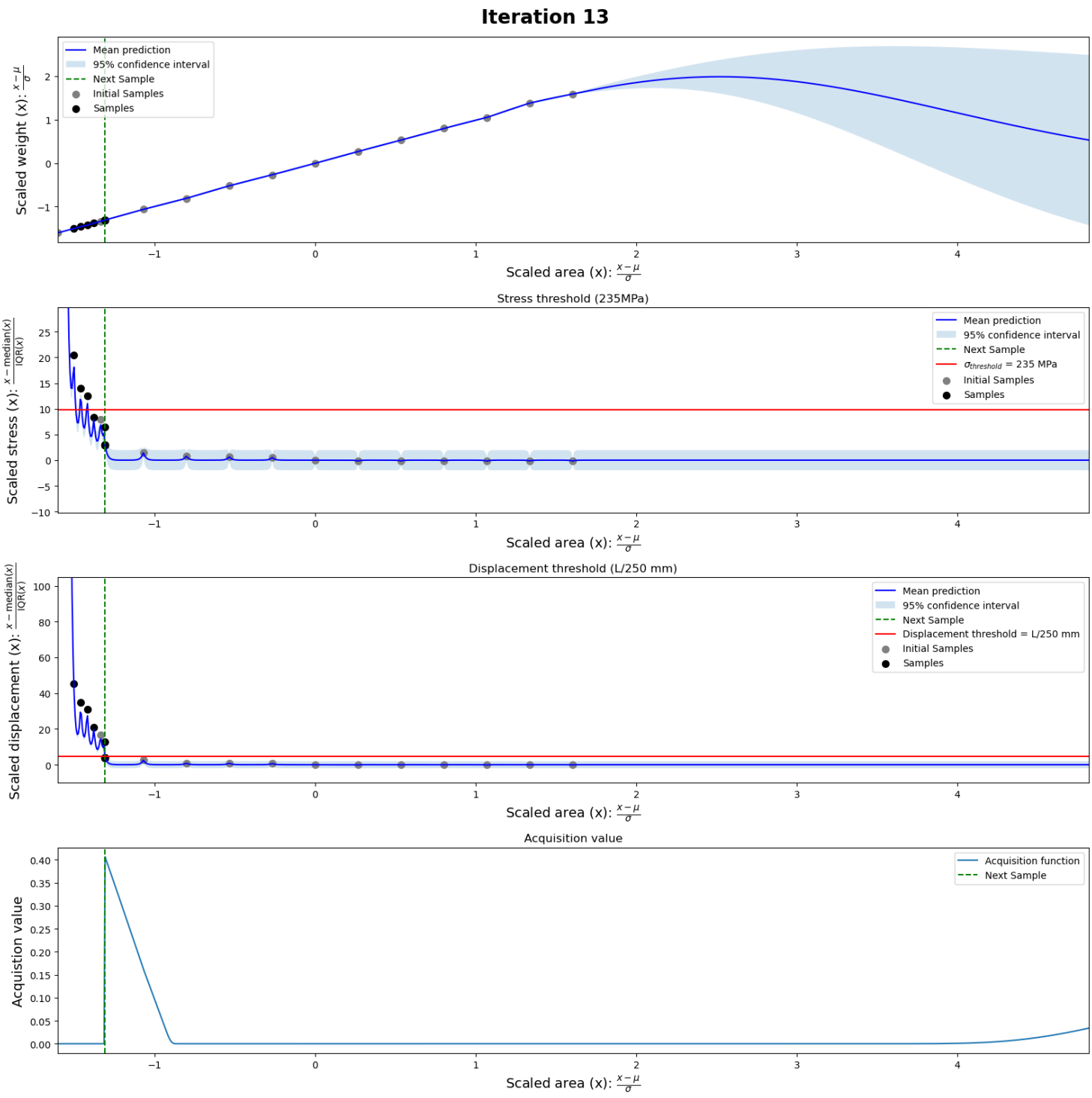


Figure 118: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 13.

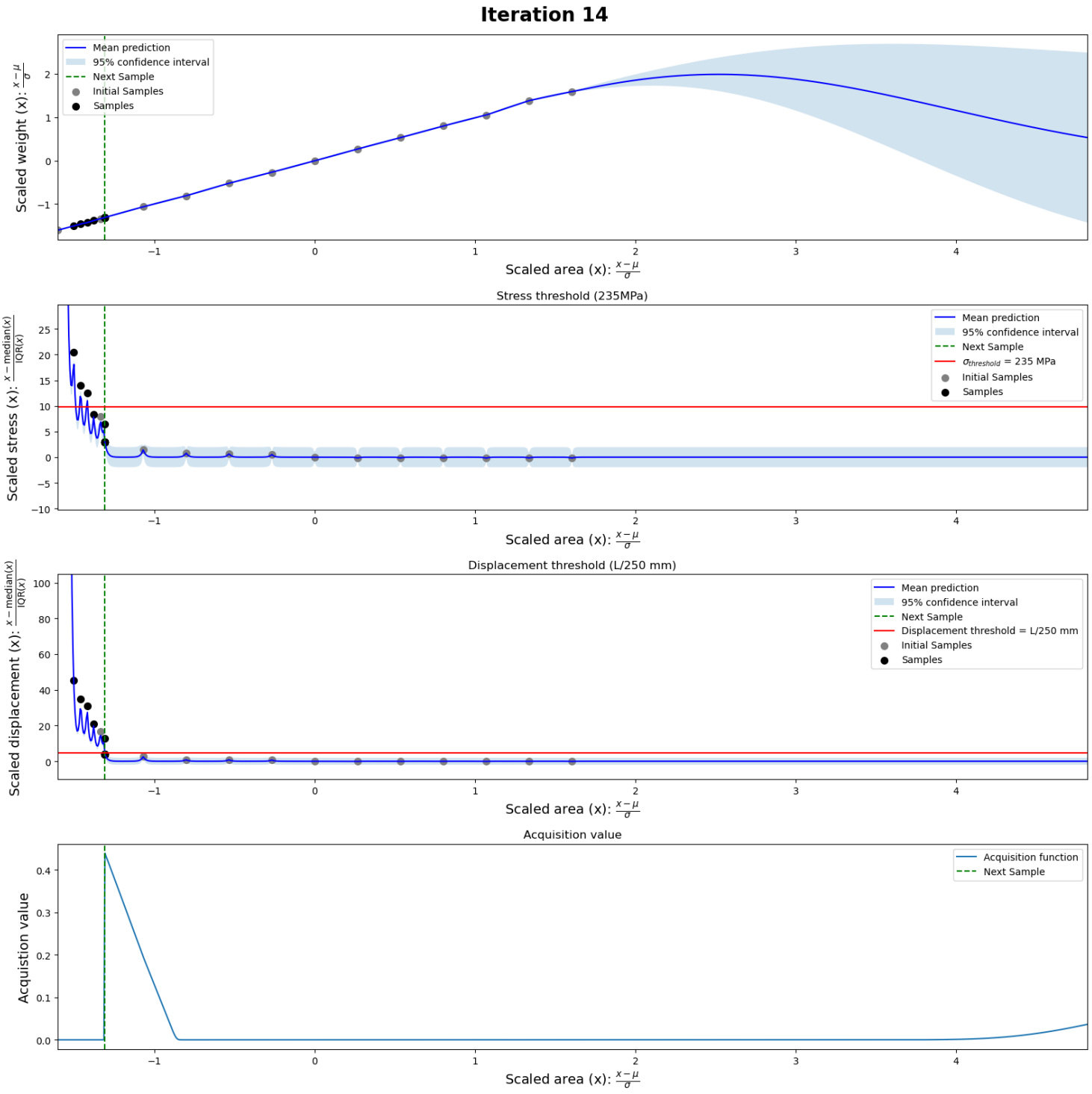


Figure 119: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 14.

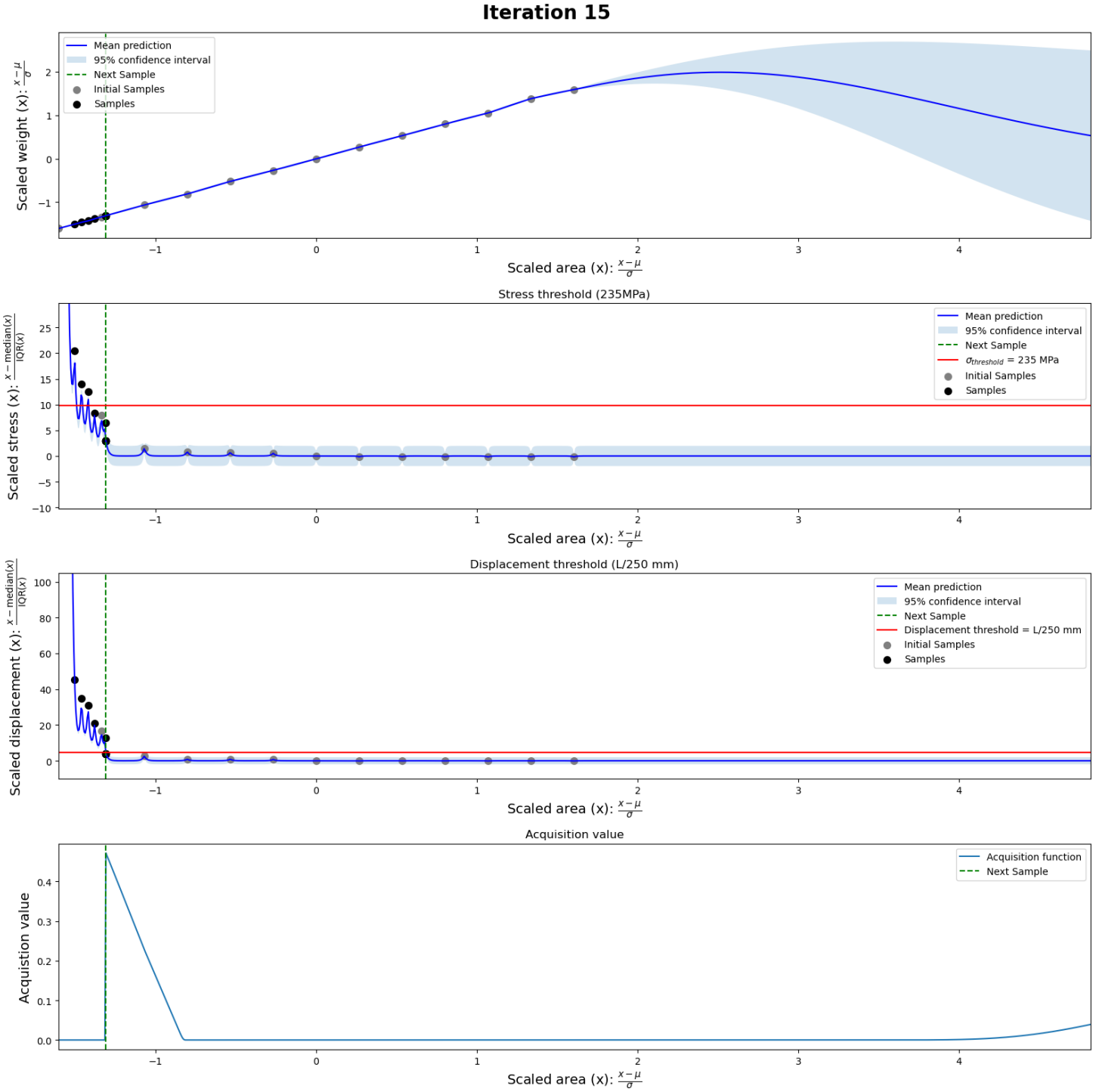


Figure 120: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 15.

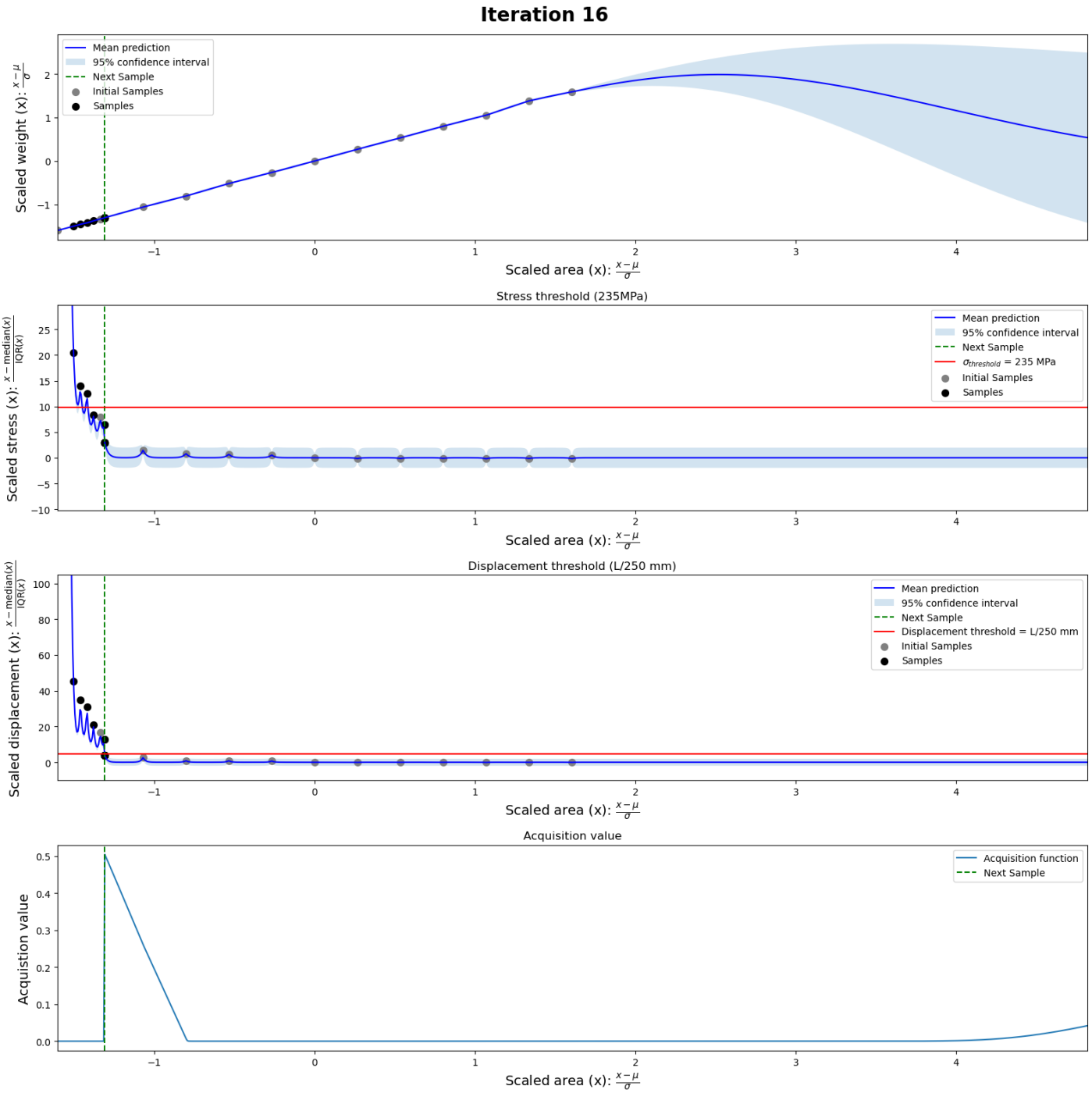


Figure 121: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 16.

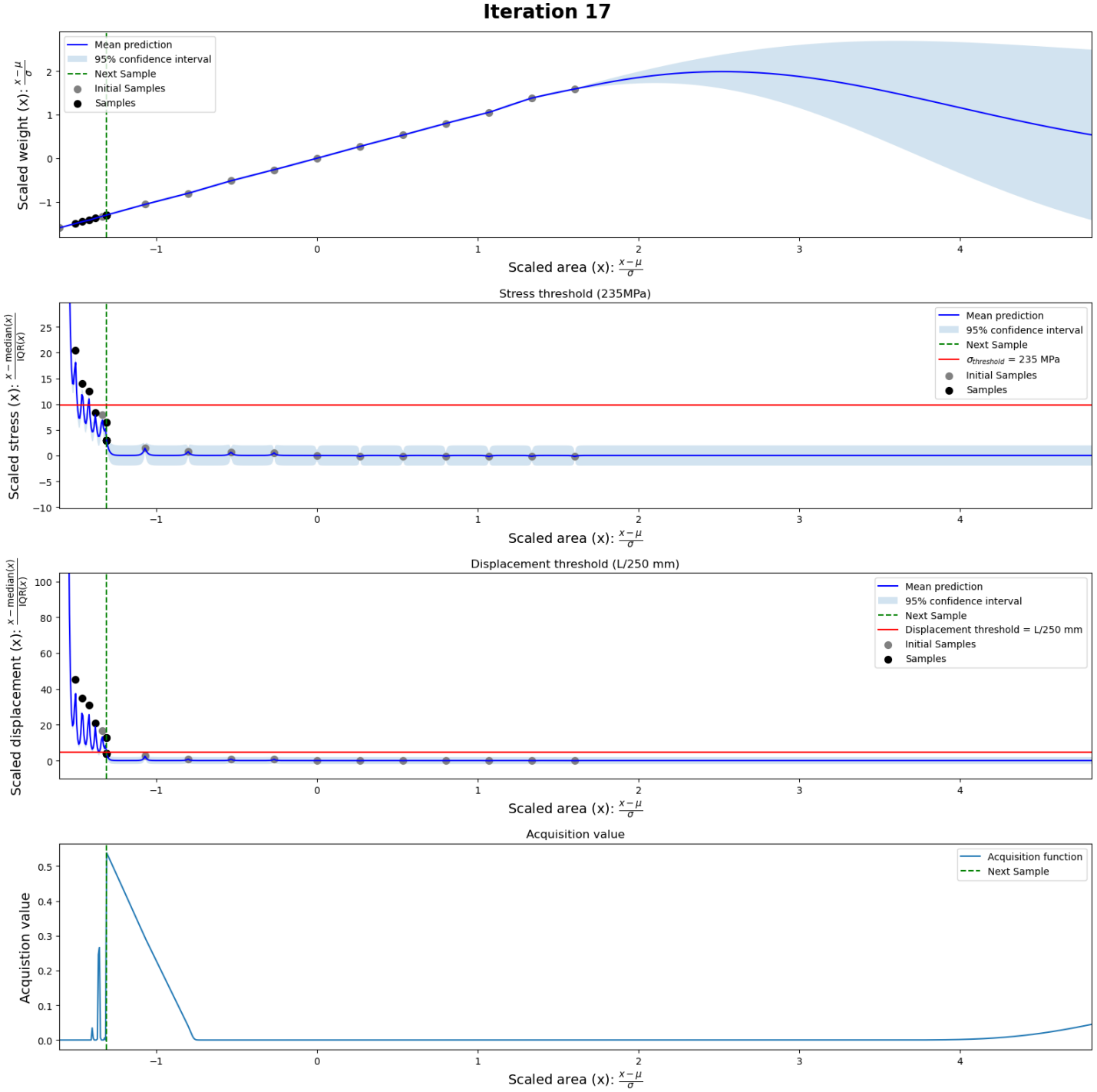


Figure 122: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 17.

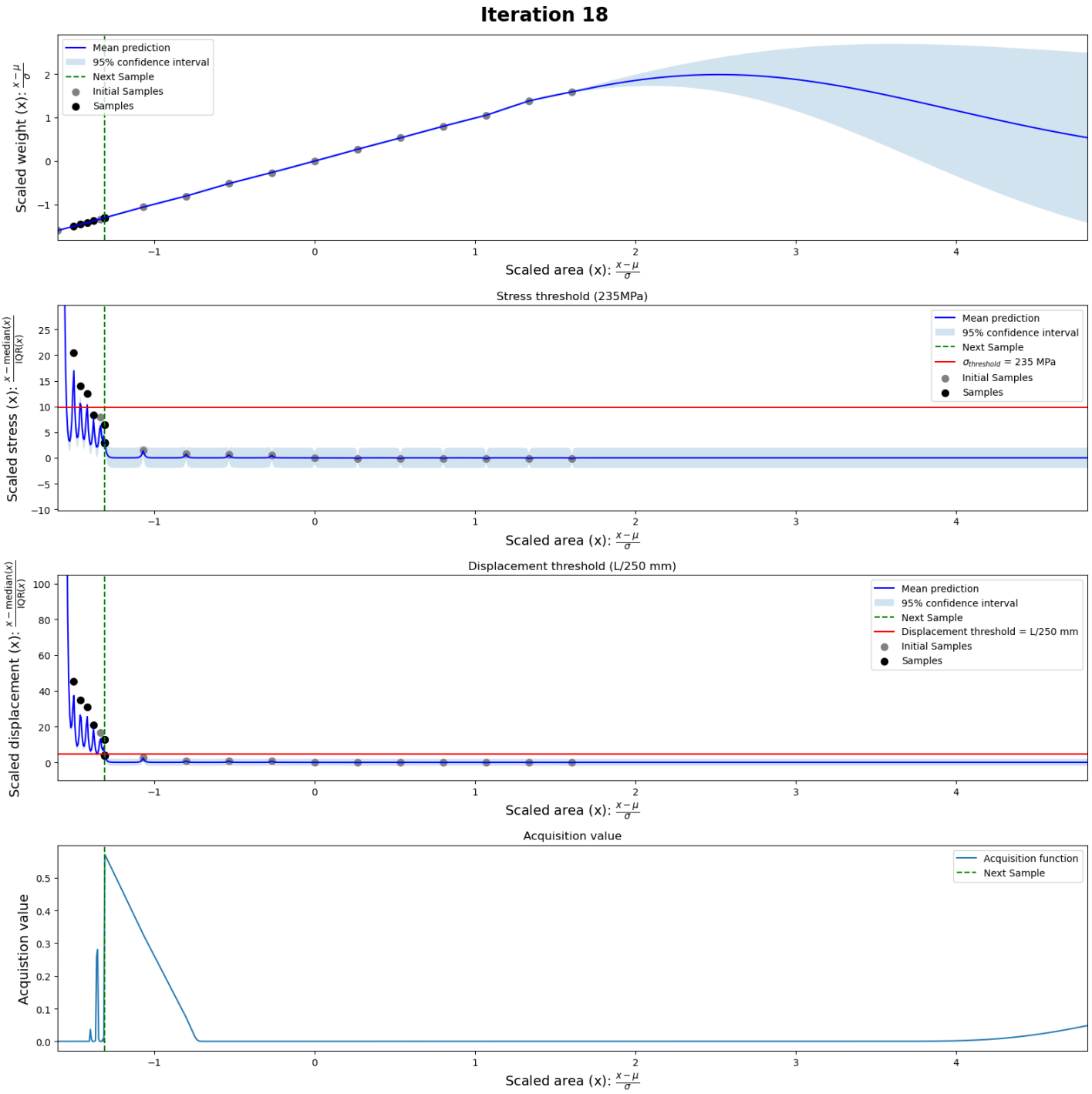


Figure 123: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 18.

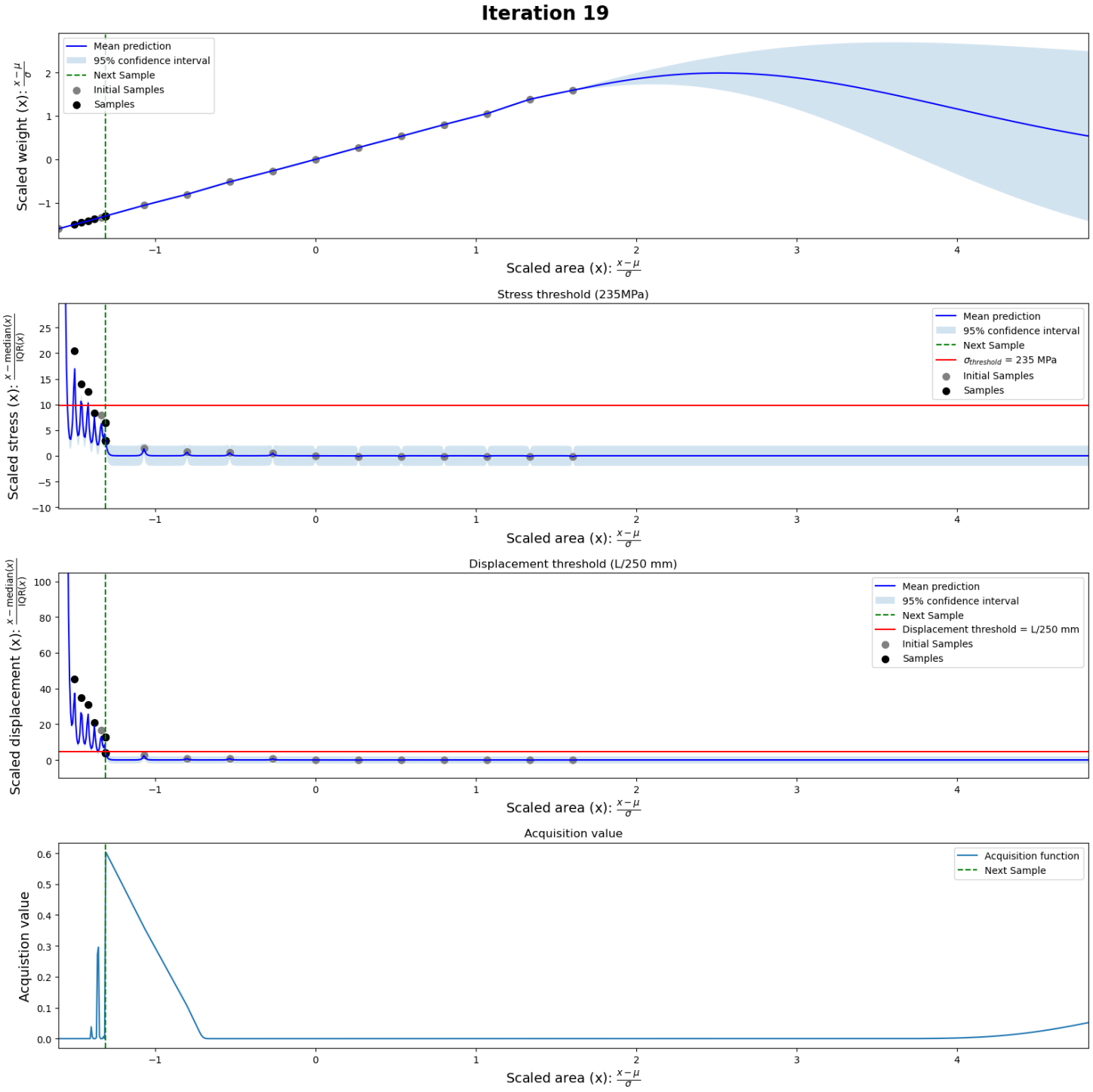


Figure 124: Bayesian optimization progression: 1D Cantilever Beam; Full Dataset; 13 initial samples. Iteration 19.

Table 16: Full profile database sorted by area in ascending order. (*Source: EurocodeApplied & Bouwen met Staal*).

ID	Typology	RFEM Member	Area [mm2]	Weight [kg]
1	CHS	CHS 21.3x2.3	137,0	4,31
2	CHS	CHS 21.3x2.6	153,0	4,80
3	CHS	CHS 26.9x2.3	178,0	5,58
4	CHS	CHS 21.3x3.2	182,0	5,71
5	CHS	CHS 26.9x2.6	198,0	6,22
6	CHS	CHS 26.9x3.2	238,0	7,47
7	CHS	CHS 33.7x2.6	254,0	7,98
8	CHS	CHS 33.7x3.2	307,0	9,64
9	CHS	CHS 42.4x2.6	325,0	10,21
10	CHS	CHS 33.7x4	373,0	11,72
11	CHS	CHS 48.3x2.6	373,0	11,71
12	SHS	SHS 40x2.6	382,0	11,99
13	RHS	RHS 50x30x2.6	382,0	11,99
14	CHS	CHS 42.4x3.2	394,0	12,37
15	CHS	CHS 48.3x3.2	453,0	14,22
16	SHS	SHS 40x3.2	460,0	14,44
17	RHS	RHS 50x30x3.2	460,0	14,44
18	CHS	CHS 60.3x2.6	471,0	14,79
19	CHS	CHS 42.4x4	483,0	15,17
20	SHS	SHS 50x2.6	486,0	15,26
21	RHS	RHS 60x40x2.6	486,0	15,26
22	CHS	CHS 48.3x4	557,0	17,49
23	SHS	SHS 40x4	559,0	17,55
24	RHS	RHS 50x30x4	559,0	16,80
25	CHS	CHS 60.3x3.2	574,0	18,02
26	SHS	SHS 50x3.2	588,0	18,46
27	RHS	RHS 60x40x3.2	588,0	18,46
28	SHS	SHS 60x2.6	590,0	18,53
29	CHS	CHS 76.1x2.6	600,0	18,85
30	SHS	SHS 40x5	673,0	21,13
31	RHS	RHS 50x30x5	673,0	21,13
32	CHS	CHS 48.3x5	680,0	21,36
33	CHS	CHS 60.3x4	707,0	22,20
34	SHS	SHS 60x3.2	716,0	22,48
35	RHS	RHS 80x40x3.2	716,0	22,48
36	SHS	SHS 50x4	719,0	22,58
37	RHS	RHS 60x40x4	719,0	21,82
38	CHS	CHS 76.1x3.2	733,0	23,02
39	IPE	IPE 80	764,0	24,00

40	SHS	SHS 70x3.2	844,0	26,38
41	RHS	RHS 90x50x3.2	844,0	26,50
42	CHS	CHS 88.9x3.2	862,0	27,07
43	CHS	CHS 60.3x5	869,0	27,29
44	SHS	SHS 50x5	873,0	27,41
45	RHS	RHS 60x40x5	873,0	26,25
46	SHS	SHS 60x4	879,0	27,60
47	RHS	RHS 80x40x4	879,0	26,85
48	UNP	UNP 65	903,0	28,35
49	CHS	CHS 76.1x4	906,0	28,45
50	RHS	RHS 100x50x3.2	908,0	28,51
51	SHS	SHS 80x3.2	972,0	30,52
52	RHS	RHS 100x60x3.2	972,0	30,52
53	CHS	CHS 101.6x3.2	989,0	28,23
54	IPE	IPE 100	1032,0	32,34
55	SHS	SHS 70x4	1039,0	32,66
56	RHS	RHS 90x50x4	1039,0	31,87
57	SHS	SHS 50x6.3	1059,0	33,28
58	RHS	RHS 60x40x6.3	1059,0	33,28
59	CHS	CHS 88.9x4	1067,0	33,60
60	SHS	SHS 60x5	1073,0	33,60
61	RHS	RHS 80x40x5	1073,0	32,53
62	UNP	UNP 80	1100,0	34,54
63	CHS	CHS 76.1x5	1117,0	35,07
64	CHS	CHS 114.3x3.2	1117,0	31,71
65	RHS	RHS 100x50x4	1119,0	34,23
66	SHS	SHS 80x4	1199,0	37,68
67	RHS	RHS 100x60x4	1199,0	36,90
68	CHS	CHS 101.6x4	1226,0	38,62
69	SHS	SHS 70x5	1273,0	39,88
70	RHS	RHS 90x50x5	1273,0	38,81
71	SHS	SHS 60x6.3	1311,0	41,13
72	RHS	RHS 80x40x6.3	1311,0	41,13
73	CHS	CHS 88.9x5	1318,0	41,45
74	IPE	IPE 120	1321,0	41,45
75	UNP	UNP 100	1350,0	42,39
76	SHS	SHS 90x4	1359,0	42,70
77	RHS	RHS 120x60x4	1359,0	41,76
78	RHS	RHS 100x50x5	1373,0	42,08
79	CHS	CHS 114.3x4	1386,0	43,65
80	SHS	SHS 80x5	1473,0	46,16
81	RHS	RHS 100x60x5	1473,0	45,09
82	CHS	CHS 101.6x5	1517,0	47,73

83	SHS	SHS 100x4	1519,0	47,73
84	RHS	RHS 120x80x4	1519,0	46,94
85	SHS	SHS 70x6.3	1563,0	48,98
86	RHS	RHS 90x50x6.3	1563,0	48,98
87	SHS	SHS 60x8	1595,0	50,24
88	RHS	RHS 80x40x8	1595,0	50,24
89	CHS	CHS 88.9x6.3	1635,0	51,34
90	IPE	IPE 140	1643,0	51,50
91	SHS	SHS 90x5	1673,0	52,44
92	RHS	RHS 120x60x5	1673,0	51,50
93	RHS	RHS 140x80x4	1679,0	51,97
94	RHS	RHS 100x50x6.3	1689,0	53,07
95	UNP	UNP 120	1700,0	53,38
96	CHS	CHS 139.7x4	1705,0	53,69
97	CHS	CHS 114.3x5	1717,0	54,01
98	SHS	SHS 80x6.3	1815,0	56,83
99	RHS	RHS 100x60x6.3	1815,0	56,83
100	RHS	RHS 160x80x4	1839,0	56,83
101	SHS	SHS 100x5	1873,0	58,72
102	RHS	RHS 120x80x5	1873,0	57,65
103	CHS	CHS 101.6x6.3	1886,0	59,22
104	SHS	SHS 70x8	1915,0	60,29
105	RHS	RHS 90x50x8	1915,0	60,29
106	RHS	RHS 150x100x4	1919,0	59,35
107	IPE	IPE 160	2009,0	63,11
108	UNP	UNP 140	2040,0	64,06
109	CHS	CHS 168.3x4	2065,0	64,84
110	SHS	SHS 90x6.3	2067,0	65,00
111	RHS	RHS 120x60x6.3	2067,0	65,00
112	RHS	RHS 140x80x5	2073,0	63,93
113	RHS	RHS 100x50x8	2075,0	65,31
114	CHS	CHS 139.7x5	2116,0	66,57
115	HEA	HEA 100	2124,0	66,69
116	CHS	CHS 114.3x6.3	2138,0	67,13
117	RHS	RHS 180x100x4	2159,0	67,04
118	SHS	SHS 80x8	2235,0	70,34
119	RHS	RHS 100x60x8	2235,0	70,34
120	SHS	SHS 120x5	2273,0	71,28
121	RHS	RHS 160x80x5	2273,0	70,34
122	SHS	SHS 100x6.3	2319,0	72,85
123	RHS	RHS 120x80x6.3	2319,0	72,85
124	RHS	RHS 200x100x4	2319,0	71,91
125	CHS	CHS 101.6x8	2352,0	68,77

126	RHS	RHS 150x100x5	2373,0	73,48
127	IPE	IPE 180	2395,0	75,05
128	UNP	UNP 160	2400,0	75,36
129	HEA	HEA 120	2534,0	79,57
130	SHS	SHS 90x8	2555,0	80,38
131	RHS	RHS 120x60x8	2555,0	80,38
132	CHS	CHS 168.3x5	2565,0	80,54
133	RHS	RHS 140x80x6.3	2571,0	80,70
134	HEB	HEB 100	2604,0	81,77
135	CHS	CHS 139.7x6.3	2640,0	82,90
136	CHS	CHS 114.3x8	2672,0	83,89
137	SHS	SHS 140x5	2673,0	83,84
138	RHS	RHS 180x100x5	2673,0	82,77
139	CHS	CHS 177.8x5	2714,0	85,09
140	UNP	UNP 180	2800,0	87,92
141	SHS	SHS 120x6.3	2823,0	88,55
142	RHS	RHS 160x80x6.3	2823,0	88,55
143	IPE	IPE 200	2848,0	89,49
144	SHS	SHS 150x5	2873,0	90,12
145	RHS	RHS 200x100x5	2873,0	89,05
146	SHS	SHS 100x8	2875,0	90,43
147	RHS	RHS 120x80x8	2875,0	85,53
148	CHS	CHS 101.6x10	2878,0	90,36
149	RHS	RHS 150x100x6.3	2949,0	92,63
150	CHS	CHS 193.7x5	2964,0	93,07
151	SHS	SHS 160x5	3073,0	96,40
152	RHS	RHS 120x60x10	3093,0	97,03
153	HEA	HEA 140	3142,0	98,66
154	RHS	RHS 140x80x8	3195,0	100,48
155	CHS	CHS 168.3x6.3	3206,0	100,67
156	UNP	UNP 200	3220,0	101,11
157	CHS	CHS 114.3x10	3277,0	102,89
158	CHS	CHS 139.7x8	3310,0	103,93
159	SHS	SHS 140x6.3	3327,0	104,56
160	RHS	RHS 180x100x6.3	3327,0	104,56
161	IPE	IPE 220	3337,0	104,88
162	CHS	CHS 219.1x5	3363,0	105,60
163	CHS	CHS 177.8x6.3	3394,0	106,58
164	HEB	HEB 120	3401,0	106,79
165	SHS	SHS 180x5	3473,0	108,96
166	SHS	SHS 100x10	3493,0	109,59

167	RHS	RHS 120x80x10	3493,0	102,27
168	SHS	SHS 120x8	3515,0	110,53
169	RHS	RHS 160x80x8	3515,0	105,63
170	SHS	SHS 150x6.3	3579,0	112,41
171	RHS	RHS 200x100x6.3	3579,0	112,41
172	RHS	RHS 150x100x8	3675,0	110,65
173	CHS	CHS 193.7x6.3	3709,0	116,46
174	UNP	UNP 220	3740,0	117,44
175	CHS	CHS 244.5x5	3762,0	118,13
176	SHS	SHS 160x6.3	3831,0	120,26
177	RHS	RHS 200x120x6.3	3831,0	120,26
178	SHS	SHS 200x5	3873,0	121,52
179	HEA	HEA 160	3877,0	121,74
180	RHS	RHS 140x80x10	3893,0	122,15
181	IPE	IPE 240	3912,0	122,77
182	CHS	CHS 168.3x8	4029,0	126,51
183	CHS	CHS 139.7x10	4075,0	127,96
184	SHS	SHS 140x8	4155,0	130,62
185	RHS	RHS 180x100x8	4155,0	125,73
186	CHS	CHS 273x5	4210,0	132,19
187	CHS	CHS 219.1x6.3	4212,0	132,26
188	UNP	UNP 240	4230,0	132,82
189	CHS	CHS 177.8x8	4268,0	134,00
190	SHS	SHS 120x10	4293,0	134,71
191	RHS	RHS 160x80x10	4293,0	127,39
192	HEB	HEB 140	4296,0	134,89
193	SHS	SHS 180x6.3	4335,0	135,96
194	SHS	SHS 150x8	4475,0	140,67
195	RHS	RHS 200x100x8	4475,0	135,65
196	RHS	RHS 150x100x10	4493,0	133,67
197	HEA	HEA 180	4525,0	142,09
198	IPE	IPE 270	4595,0	144,13
199	CHS	CHS 193.7x8	4667,0	146,54
200	CHS	CHS 244.5x6.3	4714,0	147,89
201	SHS	SHS 160x8	4795,0	150,72
202	RHS	RHS 200x120x8	4795,0	145,82
203	UNP	UNP 260	4830,0	151,66
204	SHS	SHS 200x6.3	4839,0	151,98
205	RHS	RHS 250x150x6.3	4839,0	151,98
206	CHS	CHS 168.3x10	4973,0	156,16

207	CHS	CHS 139.7x12.5	4995,0	156,85
208	CHS	CHS 323.9x5	5009,0	157,28
209	SHS	SHS 140x10	5093,0	159,83
210	RHS	RHS 180x100x10	5093,0	152,51
211	SHS	SHS 120x12.5	5207,0	163,59
212	RHS	RHS 160x80x12.5	5207,0	163,59
213	CHS	CHS 177.8x10	5272,0	165,53
214	CHS	CHS 273x6.3	5279,0	165,76
215	CHS	CHS 219.1x8	5306,0	166,73
216	UNP	UNP 280	5330,0	167,36
217	SHS	SHS 220x6.3	5343,0	167,68
218	RHS	RHS 260x180x6.3	5343,0	167,68
219	IPE	IPE 300	5381,0	168,93
220	HEA	HEA 200	5383,0	169,03
221	HEB	HEB 160	5425,0	170,35
222	SHS	SHS 180x8	5435,0	170,82
223	RHS	RHS 150x100x12.5	5457,0	171,44
224	SHS	SHS 150x10	5493,0	172,39
225	RHS	RHS 200x100x10	5493,0	165,16
226	CHS	CHS 193.7x10	5771,0	181,21
227	UNP	UNP 300	5880,0	184,63
228	SHS	SHS 160x10	5893,0	184,95
229	RHS	RHS 200x120x10	5893,0	177,63
230	CHS	CHS 244.5x8	5944,0	186,64
231	SHS	SHS 200x8	6075,0	190,91
232	RHS	RHS 250x150x8	6075,0	185,89
233	SHS	SHS 250x6.3	6099,0	191,54
234	RHS	RHS 300x200x6.3	6099,0	191,54
235	CHS	CHS 168.3x12.5	6118,0	192,17
236	SHS	SHS 140x12.5	6207,0	194,99
237	RHS	RHS 180x100x12.5	6207,0	194,99
238	IPE	IPE 330	6261,0	198,76
239	CHS	CHS 323.9x6.3	6286,0	197,38
240	SHS	SHS 260x6.3	6351,0	199,39
241	HEA	HEA 220	6434,0	202,03
242	CHS	CHS 177.8x12.5	6491,0	203,83
243	HEB	HEB 180	6525,0	204,89
244	CHS	CHS 219.1x10	6569,0	206,30
245	CHS	CHS 273x8	6660,0	209,12

246	SHS	SHS 180x10	6693,0	210,07
247	SHS	SHS 150x12.5	6707,0	210,69
248	RHS	RHS 200x100x12.5	6707,0	194,81
249	SHS	SHS 220x8	6715,0	211,01
250	RHS	RHS 260x180x8	6715,0	206,11
251	CHS	CHS 355.6x6.3	6913,0	216,97
252	CHS	CHS 193.7x12.5	7116,0	223,44
253	SHS	SHS 160x12.5	7207,0	226,39
254	RHS	RHS 200x120x12.5	7207,0	210,38
255	IPE	IPE 360	7273,0	226,39
256	SHS	SHS 300x6.3	7359,0	232,36
257	RHS	RHS 350x250x6.3	7359,0	231,10
258	CHS	CHS 244.5x10	7367,0	231,32
259	SHS	SHS 200x10	7493,0	235,19
260	RHS	RHS 250x150x10	7493,0	227,96
261	SHS	SHS 150x14.2	7497,0	235,50
262	UNP	UNP 320	7580,0	238,01
263	SHS	SHS 250x8	7675,0	241,15
264	RHS	RHS 300x200x8	7675,0	236,25
265	HEA	HEA 240	7684,0	241,28
266	UNP	UNP 350	7730,0	242,72
267	HEB	HEB 200	7808,0	245,17
268	CHS	CHS 406.4x6.3	7919,0	248,69
269	CHS	CHS 323.9x8	7939,0	249,32
270	SHS	SHS 260x8	7995,0	251,20
271	CHS	CHS 193.7x14.2	8008,0	251,51
272	UNP	UNP 380	8040,0	252,46
273	SHS	SHS 160x14.2	8065,0	253,40
274	CHS	CHS 219.1x12.5	8113,0	254,75
275	SHS	SHS 180x12.5	8207,0	257,79
276	CHS	CHS 273x10	8262,0	259,43
277	SHS	SHS 220x10	8293,0	260,31
278	RHS	RHS 260x180x10	8293,0	252,99
279	SHS	SHS 150x16	8301,0	260,62
280	RHS	RHS 200x100x16	8301,0	260,62
281	IPE	IPE 400	8446,0	264,07
282	HEA	HEA 260	8682,0	272,61
283	CHS	CHS 355.6x8	8736,0	274,44
284	CHS	CHS 457x6.3	8920,0	280,10
285	CHS	CHS 193.7x16	8932,0	280,47

286	SHS	SHS 160x16	8941,0	280,72
287	HEB	HEB 220	9104,0	285,87
288	CHS	CHS 244.5x12.5	9111,0	286,09
289	CHS	CHS 219.1x14.2	9141,0	287,00
290	UNP	UNP 400	9150,0	287,31
291	SHS	SHS 180x14.2	9201,0	288,88
292	SHS	SHS 200x12.5	9207,0	289,19
293	RHS	RHS 250x150x12.5	9207,0	273,31
294	SHS	SHS 300x8	9275,0	292,02
295	RHS	RHS 350x250x8	9275,0	291,39
296	RHS	RHS 400x200x8	9275,0	286,49
297	SHS	SHS 250x10	9493,0	297,99
298	RHS	RHS 300x200x10	9493,0	290,67
299	HEA	HEA 280	9726,0	305,40
300	CHS	CHS 323.9x10	9861,0	309,60
301	IPE	IPE 450	9882,0	309,29
302	SHS	SHS 260x10	9893,0	310,55
303	CHS	CHS 508x6.3	9930,0	311,79
304	CHS	CHS 406.4x8	10013,0	314,00
305	SHS	SHS 220x12.5	10207,0	320,28
306	RHS	RHS 260x180x12.5	10207,0	320,28
307	CHS	CHS 219.1x16	10209,0	320,56
308	SHS	SHS 180x16	10221,0	320,28
309	CHS	CHS 273x12.5	10230,0	321,22
310	CHS	CHS 244.5x14.2	10274,0	323,42
311	SHS	SHS 200x14.2	10337,0	323,42
312	RHS	RHS 250x150x14.2	10337,0	323,42
313	HEB	HEB 240	10599,0	332,84
314	CHS	CHS 355.6x10	10857,0	342,26
315	SHS	SHS 350x8	10875,0	342,26
316	RHS	RHS 450x250x8	10875,0	342,26
317	HEA	HEA 300	11253,0	353,25
318	CHS	CHS 457x8	11285,0	354,82
319	SHS	SHS 220x14.2	11473,0	361,10
320	RHS	RHS 260x180x14.2	11473,0	361,10
321	CHS	CHS 244.5x16	11486,0	360,65
322	SHS	SHS 300x10	11493,0	361,10
323	RHS	RHS 350x250x10	11493,0	361,10
324	RHS	RHS 400x200x10	11493,0	353,47

325	SHS	SHS 200x16	11501,0	361,10
326	RHS	RHS 250x150x16	11501,0	361,10
327	CHS	CHS 273x14.2	11545,0	362,52
328	IPE	IPE 500	11552,0	361,10
329	SHS	SHS 250x12.5	11707,0	367,38
330	RHS	RHS 300x200x12.5	11707,0	351,81
331	HEB	HEB 260	11844,0	371,78
332	CHS	CHS 610x6.3	11948,0	375,18
333	SHS	SHS 260x12.5	12207,0	383,08
334	CHS	CHS 323.9x12.5	12229,0	383,99
335	HEA	HEA 320	12437,0	390,62
336	CHS	CHS 406.4x10	12453,0	392,50
337	CHS	CHS 219.1x20	12510,0	392,81
338	CHS	CHS 508x8	12566,0	395,64
339	SHS	SHS 220x16	12781,0	401,92
340	RHS	RHS 260x180x16	12781,0	401,92
341	CHS	CHS 273x16	12918,0	405,63
342	HEB	HEB 280	13136,0	412,60
343	SHS	SHS 250x14.2	13177,0	414,48
344	RHS	RHS 300x200x14.2	13177,0	414,48
345	HEA	HEA 340	13347,0	419,19
346	IPE	IPE 550	13442,0	427,04
347	CHS	CHS 355.6x12.5	13474,0	423,90
348	SHS	SHS 350x10	13493,0	423,90
349	RHS	RHS 450x250x10	13493,0	423,90
350	SHS	SHS 260x14.2	13745,0	430,18
351	CHS	CHS 323.9x14.2	13816,0	433,32
352	CHS	CHS 711x6.3	13947,0	437,95
353	CHS	CHS 457x10	14043,0	439,60
354	CHS	CHS 244.5x20	14106,0	442,92
355	SHS	SHS 300x12.5	14207,0	445,88
356	RHS	RHS 350x250x12.5	14207,0	445,88
357	RHS	RHS 400x200x12.5	14207,0	430,31
358	HEA	HEA 360	14276,0	448,39
359	SHS	SHS 250x16	14701,0	461,58
360	RHS	RHS 300x200x16	14701,0	461,58
361	HEB	HEB 300	14908,0	468,17
362	CHS	CHS 762x6.3	14957,0	469,64
363	CHS	CHS 610x8	15130,0	475,08

364	CHS	CHS 355.6x14.2	15230,0	477,28
365	SHS	SHS 260x16	15341,0	480,42
366	CHS	CHS 406.4x12.5	15468,0	486,70
367	CHS	CHS 323.9x16	15477,0	485,97
368	SHS	SHS 400x10	15493,0	486,70
369	RHS	RHS 500x300x10	15493,0	486,70
370	IPE	IPE 600	15598,0	483,56
371	CHS	CHS 508x10	15645,0	489,84
372	CHS	CHS 273x20	15896,0	499,15
373	HEA	HEA 400	15898,0	499,26
374	SHS	SHS 300x14.2	16017,0	502,40
375	RHS	RHS 350x250x14.2	16017,0	502,40
376	RHS	RHS 400x200x14.2	16017,0	502,40
377	HEB	HEB 320	16134,0	506,48
378	SHS	SHS 350x12.5	16707,0	524,38
379	RHS	RHS 450x250x12.5	16707,0	524,38
380	CHS	CHS 355.6x16	17070,0	536,00
381	HEB	HEB 340	17090,0	536,63
382	CHS	CHS 244.5x25	17239,0	541,32
383	CHS	CHS 457x12.5	17455,0	548,10
384	CHS	CHS 406.4x14.2	17496,0	549,50
385	CHS	CHS 711x8	17668,0	554,79
386	HEA	HEA 450	17803,0	558,92
387	SHS	SHS 300x16	17901,0	562,06
388	RHS	RHS 350x250x16	17901,0	562,06
389	RHS	RHS 400x200x16	17901,0	562,06
390	HEB	HEB 360	18063,0	567,08
391	CHS	CHS 610x10	18850,0	591,88
392	SHS	SHS 350x14.2	18857,0	593,46
393	RHS	RHS 450x250x14.2	18857,0	593,46
394	CHS	CHS 762x8	18950,0	595,03
395	CHS	CHS 323.9x20	19095,0	599,57
396	SHS	SHS 400x12.5	19207,0	602,88
397	RHS	RHS 500x300x12.5	19207,0	602,88
398	CHS	CHS 508x12.5	19458,0	610,99
399	CHS	CHS 273x25	19478,0	611,61
400	CHS	CHS 406.4x16	19624,0	616,18
401	HEA	HEA 500	19754,0	620,15
402	CHS	CHS 457x14.2	19754,0	620,26

403	HEB	HEB 400	19778,0	621,09
404	CHS	CHS 813x8	20232,0	635,28
405	CHS	CHS 355.6x20	21086,0	662,11
406	SHS	SHS 350x16	21101,0	662,54
407	RHS	RHS 450x250x16	21101,0	662,54
408	HEA	HEA 550	21176,0	665,05
409	SHS	SHS 400x14.2	21697,0	681,38
410	RHS	RHS 500x300x14.2	21697,0	681,38
411	HEB	HEB 450	21798,0	684,52
412	CHS	CHS 711x10	22023,0	691,51
413	CHS	CHS 508x14.2	22029,0	691,70
414	CHS	CHS 457x16	22167,0	696,05
415	HEA	HEA 600	22646,0	711,21
416	CHS	CHS 914x8	22770,0	714,99
417	CHS	CHS 610x12.5	23464,0	736,76
418	CHS	CHS 323.9x25	23476,0	737,13
419	CHS	CHS 762x10	23625,0	741,82
420	HEB	HEB 500	23864,0	749,20
421	HEA	HEA 650	24164,0	758,62
422	CHS	CHS 406.4x20	24278,0	762,34
423	SHS	SHS 400x16	24301,0	763,02
424	RHS	RHS 500x300x16	24301,0	763,02
425	CHS	CHS 508x16	24731,0	776,54
426	CHS	CHS 813x10	25227,0	792,13
427	CHS	CHS 1016x8	25334,0	795,48
428	HEB	HEB 550	25406,0	797,87
429	CHS	CHS 355.6x25	25965,0	815,31
430	HEA	HEA 700	26048,0	817,97
431	CHS	CHS 610x14.2	26579,0	834,58
432	HEB	HEB 600	26996,0	847,80
433	CHS	CHS 711x12.5	27430,0	861,30
434	CHS	CHS 457x20	27458,0	862,17
435	CHS	CHS 914x10	28400,0	891,76
436	HEA	HEA 800	28583,0	897,41
437	HEB	HEB 650	28634,0	898,98
438	CHS	CHS 762x12.5	29433,0	924,19
439	CHS	CHS 610x16	29858,0	937,53
440	CHS	CHS 406.4x25	29955,0	940,59
441	SHS	SHS 400x20	29971,0	942,00
442	RHS	RHS 500x300x20	29971,0	942,00
443	HEB	HEB 700	30638,0	962,10
444	CHS	CHS 508x20	30662,0	962,79

445	CHS	CHS 711x14.2	31085,0	976,06
446	CHS	CHS 813x12.5	31436,0	987,08
447	CHS	CHS 1016x10	31604,0	992,38
448	HEA	HEA 900	32053,0	1.006,37
449	CHS	CHS 1067x10	33207,0	1.042,69
450	CHS	CHS 762x14.2	33360,0	1.047,50
451	HEB	HEB 800	33418,0	1.049,39
452	CHS	CHS 457x25	33929,0	1.065,38
453	HEA	HEA 1000	34685,0	1.088,95
454	CHS	CHS 711x16	34935,0	1.096,94
455	CHS	CHS 914x12.5	35402,0	1.111,62
456	CHS	CHS 406.4x30	35475,0	1.113,91
457	CHS	CHS 813x14.2	35635,0	1.118,94
458	CHS	CHS 1168x10	36380,0	1.142,32
459	CHS	CHS 610x20	37071,0	1.164,02
460	HEB	HEB 900	37128,0	1.165,88
461	CHS	CHS 762x16	37498,0	1.177,44
462	CHS	CHS 508x25	37935,0	1.191,15
463	CHS	CHS 1219x10	37982,0	1.192,63
464	CHS	CHS 1016x12.5	39407,0	1.237,39
465	HEB	HEB 1000	40005,0	1.256,00
466	CHS	CHS 813x16	40062,0	1.257,93
467	CHS	CHS 914x14.2	40141,0	1.260,42
468	CHS	CHS 457x30	40244,0	1.263,66
469	CHS	CHS 1067x12.5	41410,0	1.300,28
470	CHS	CHS 711x20	43417,0	1.363,29
471	CHS	CHS 1016x14.2	44691,0	1.403,29
472	CHS	CHS 508x30	45050,0	1.414,58
473	CHS	CHS 914x16	45138,0	1.417,35
474	CHS	CHS 1168x12.5	45376,0	1.424,82
475	CHS	CHS 610x25	45946,0	1.442,70
476	CHS	CHS 406.4x40	46043,0	1.445,76
477	CHS	CHS 762x20	46621,0	1.463,91
478	CHS	CHS 1067x14.2	46966,0	1.474,73
479	CHS	CHS 1219x12.5	47379,0	1.487,71
480	CHS	CHS 813x20	49826,0	1.564,53
481	CHS	CHS 1016x16	50265,0	1.578,34
482	CHS	CHS 1168x14.2	51472,0	1.616,21
483	CHS	CHS 457x40	52402,0	1.645,42
484	CHS	CHS 1067x16	52829,0	1.658,83
485	CHS	CHS 1219x14.2	53747,0	1.687,65
486	CHS	CHS 711x25	53878,0	1.691,78
487	CHS	CHS 610x30	54664,0	1.716,44

488	CHS	CHS 914x20	56172,0	1.763,79
489	CHS	CHS 762x25	57884,0	1.817,55
490	CHS	CHS 1168x16	57906,0	1.818,24
491	CHS	CHS 508x40	58811,0	1.846,65
492	CHS	CHS 1219x16	60469,0	1.898,74
493	CHS	CHS 813x25	61889,0	1.943,33
494	CHS	CHS 1016x20	62581,0	1.965,03
495	CHS	CHS 711x30	64183,0	2.015,34
496	CHS	CHS 1067x20	65785,0	2.065,65
497	CHS	CHS 762x30	68989,0	2.166,27
498	CHS	CHS 914x25	69822,0	2.192,41
499	CHS	CHS 610x40	71628,0	2.249,13
500	CHS	CHS 508x50	71942,0	2.258,99
501	CHS	CHS 1168x20	72131,0	2.264,91
502	CHS	CHS 813x30	73796,0	2.317,19
503	CHS	CHS 1219x20	75335,0	2.365,53
504	CHS	CHS 1016x25	77833,0	2.443,95
505	CHS	CHS 1067x25	81838,0	2.569,73
506	CHS	CHS 914x30	83315,0	2.616,09
507	CHS	CHS 711x40	84320,0	2.647,66
508	CHS	CHS 610x50	87965,0	2.762,09
509	CHS	CHS 1168x25	89771,0	2.818,81
510	CHS	CHS 762x40	90729,0	2.848,90
511	CHS	CHS 1016x30	92928,0	2.917,95
512	CHS	CHS 1219x25	93777,0	2.944,58
513	CHS	CHS 1067x30	97735,0	3.068,88
514	CHS	CHS 711x50	103830,0	3.260,25
515	CHS	CHS 762x50	111841,0	3.511,80
516	CHS	CHS 711x60	122711,0	3.853,11

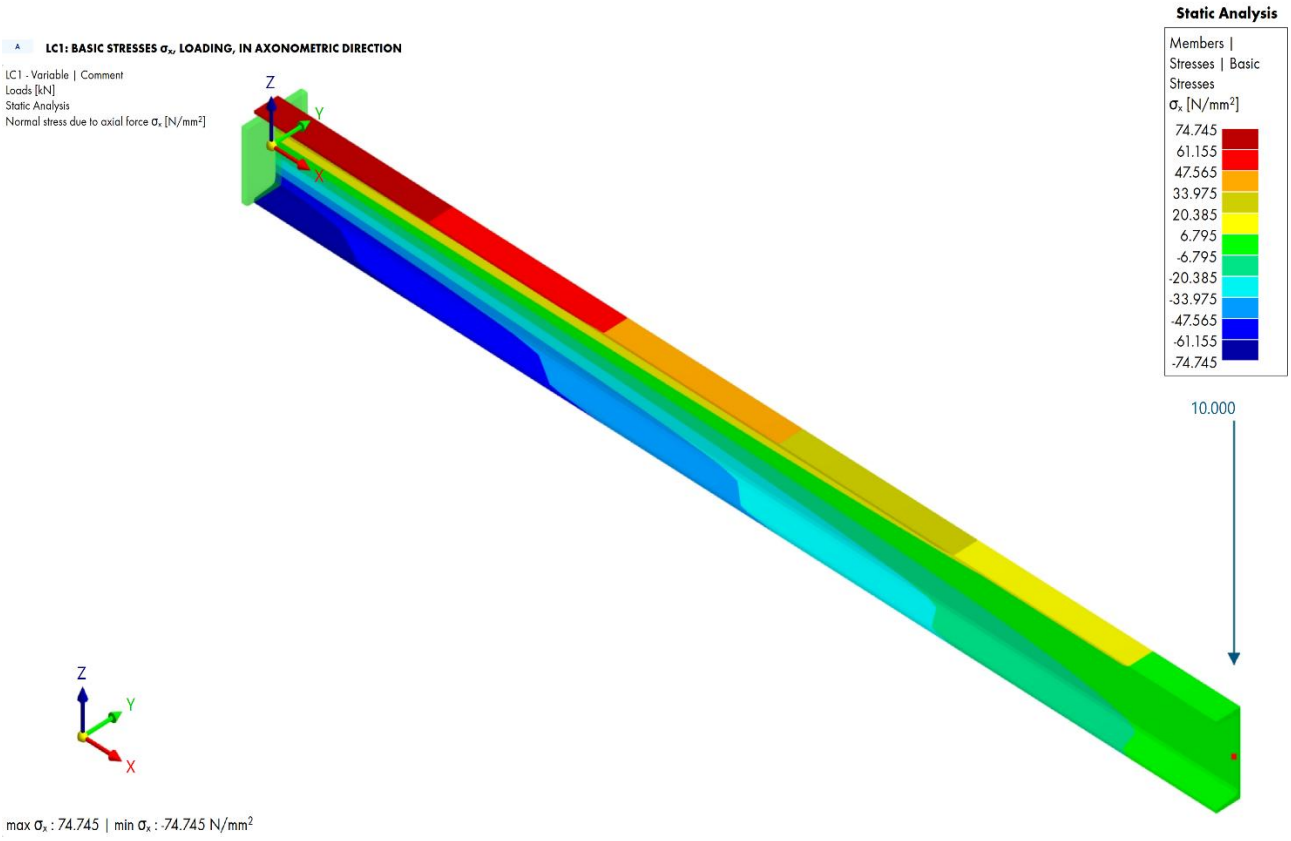


Figure 125: 1D cantilever beam. UNP 300 stresses.

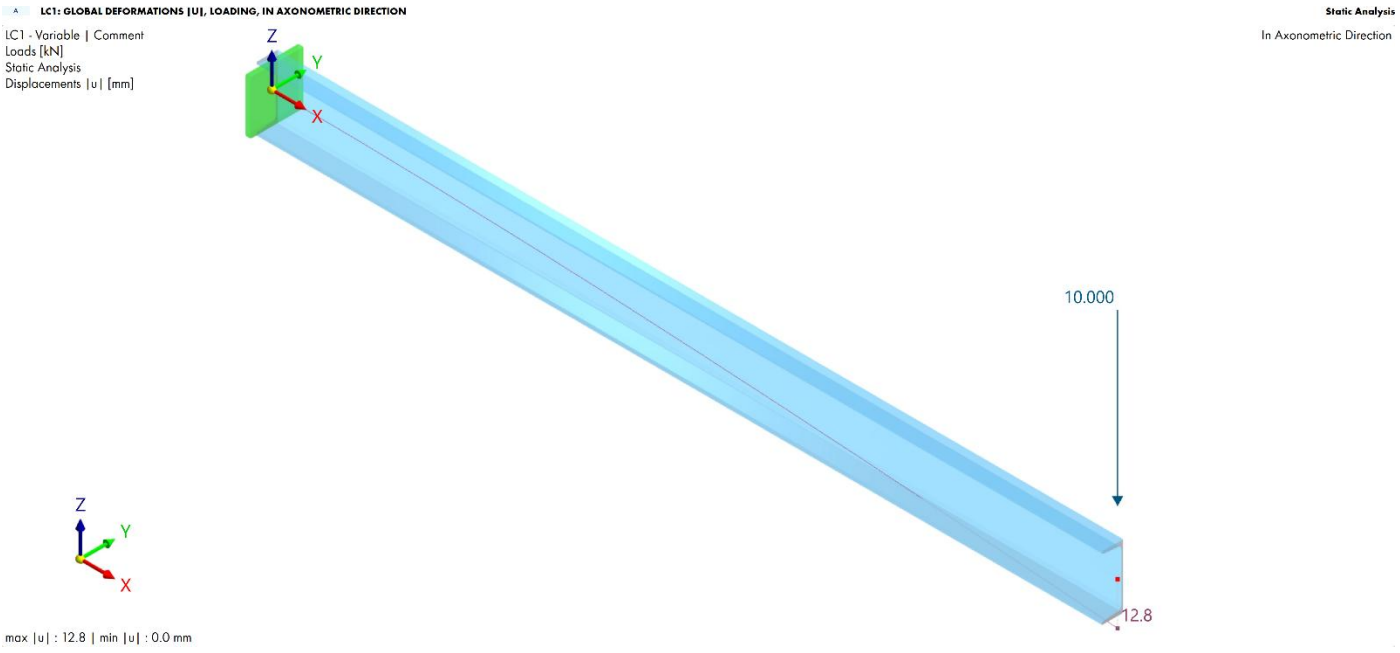


Figure 126: 1D cantilever beam. UNP 300 deflection.

Appendix C: 2D optimization problem: Cantilever Truss

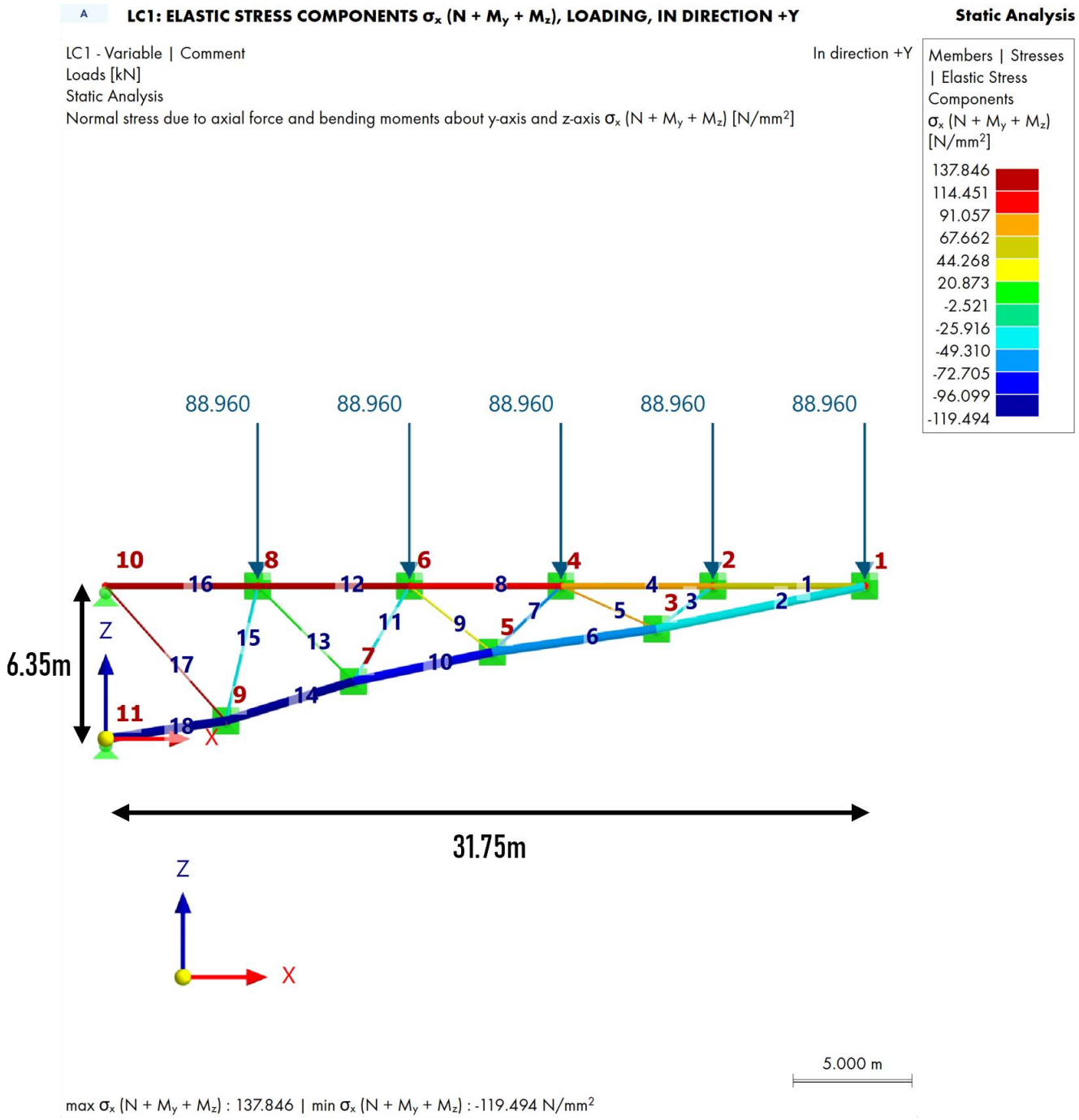


Figure 127: Optimal truss layout and cross-sections found by Gholizadeh. CHS dataset. Stresses based on static analysis.

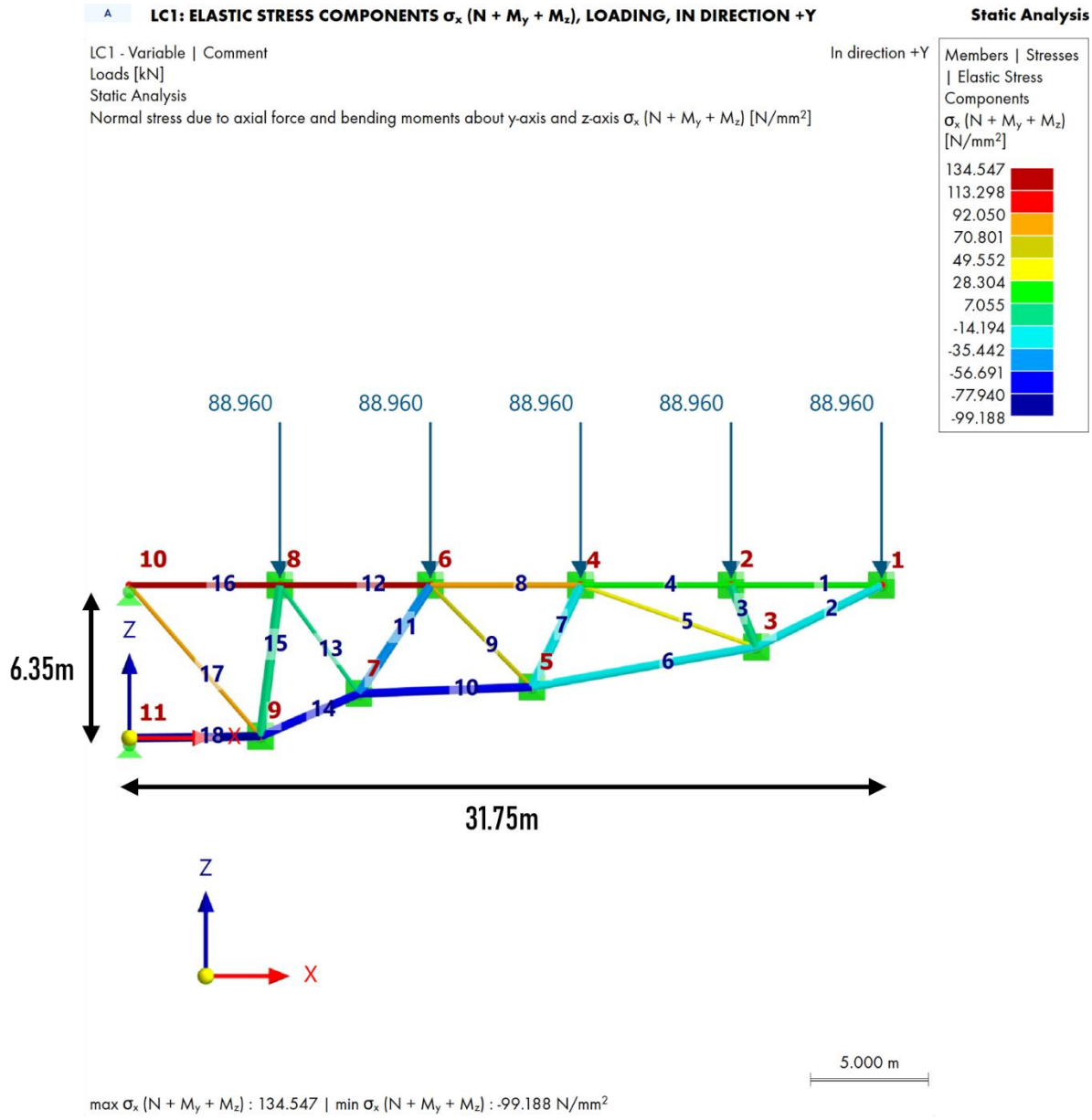


Figure 128: Optimal truss layout and cross-sections found by Bayesian optimization algorithm. Discrete CHS cross-sections. 4GPs approach.

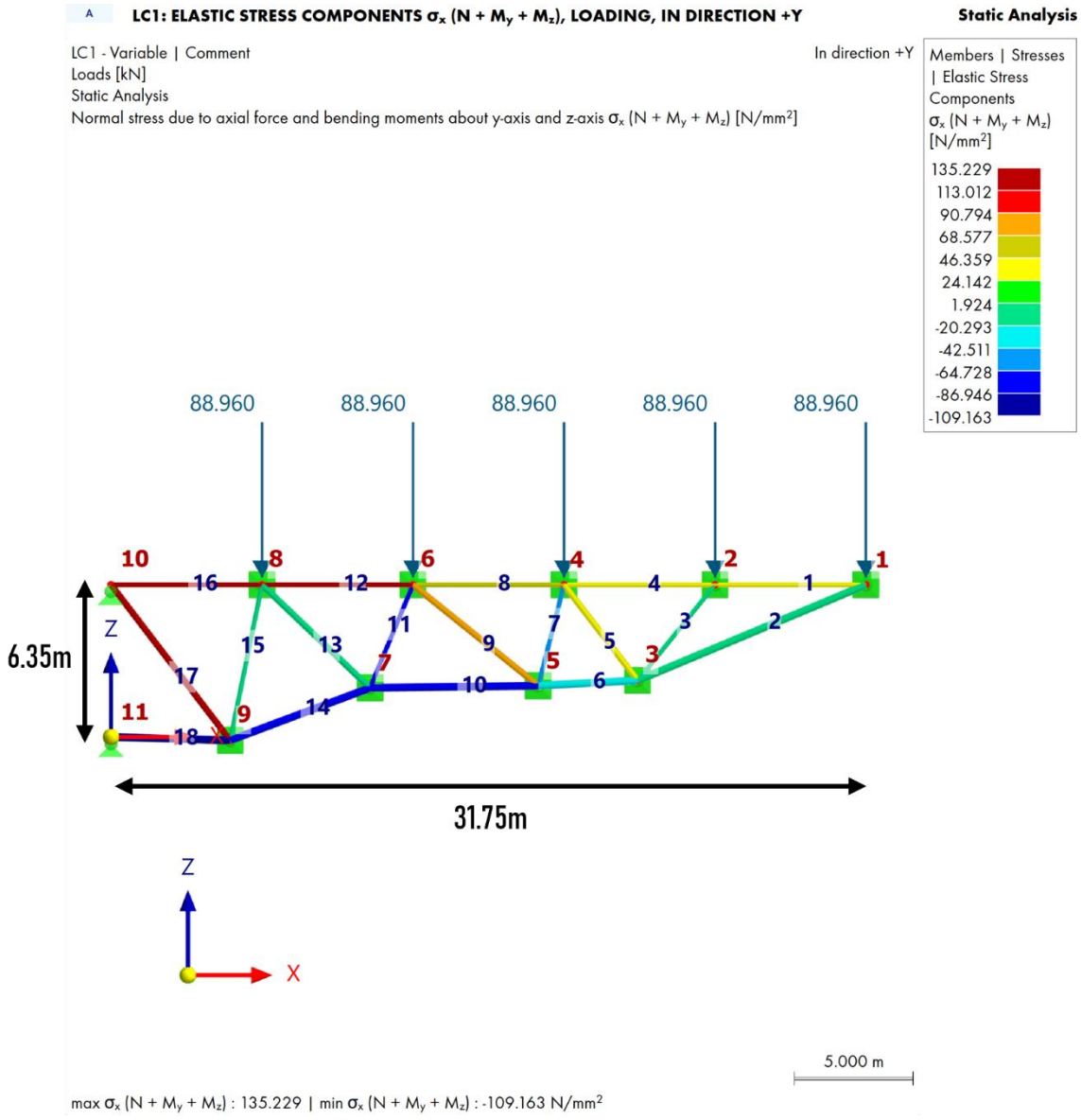


Figure 129: Optimal truss layout and cross-sections found by Bayesian optimization algorithm. Discrete CHS cross-sections. 18GPs approach.

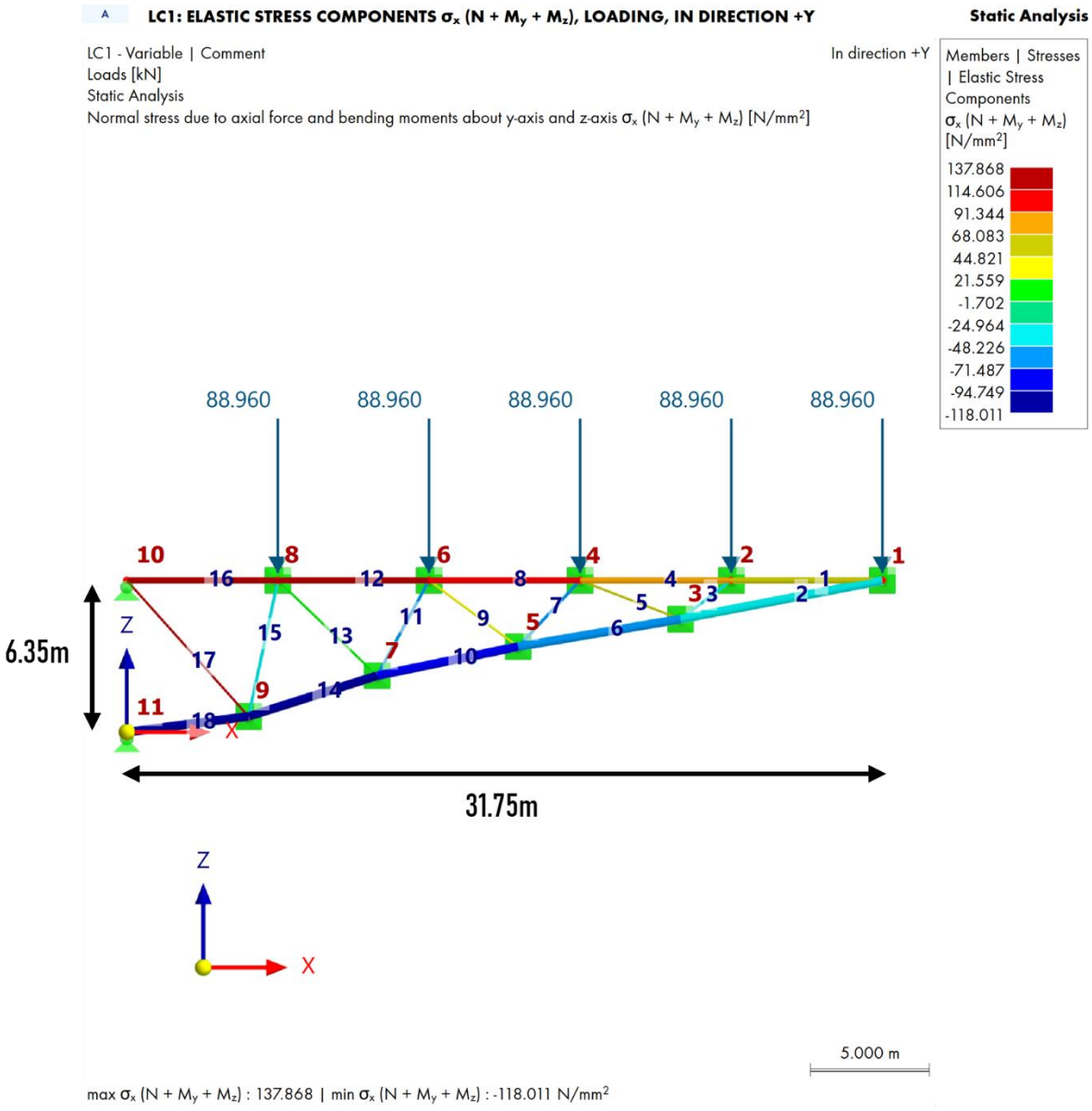


Figure 130: Optimal truss layout and cross-sections found by Bayesian optimization algorithm. Parametric CHS cross-sections. 18GPs approach.

Table 17: Cantilever truss case study cross-sectional areas in in^2, mm^2 and the corresponding parametric CHS profile.

Area (in^2)	Area (mm^2)	Parametric CHS profile
2	1290.32	CHS 51.0/10
2.25	1451.61	CHS 56.0/10
2.5	1612.9	CHS 61.0/10
2.75	1774.19	CHS 66.0/10
3	1935.48	CHS 72.0/10
3.25	2096.77	CHS 77.0/10
3.5	2258.06	CHS 82.0/10
3.75	2419.35	CHS 87.0/10
4	2580.64	CHS 92.0/10
4.25	2741.93	CHS 97.0/10
4.5	2903.22	CHS 102.0/10
4.75	3064.51	CHS 108.0/10
5	3225.8	CHS 113.0/10
5.25	3387.09	CHS 118.0/10
5.5	3548.38	CHS 123.0/10
5.75	3709.67	CHS 128.0/10
6	3870.96	CHS 133.0/10
6.25	4032.25	CHS 138.0/10
6.5	4193.54	CHS 143.0/10
6.75	4354.83	CHS 149.0/10
7	4516.12	CHS 154.0/10
7.25	4677.41	CHS 159.0/10
7.5	4838.7	CHS 164.0/10
7.75	4999.99	CHS 169.0/10
8	5161.28	CHS 174.0/10
8.25	5322.57	CHS 179.0/10
8.5	5483.86	CHS 185.0/10
8.75	5645.15	CHS 190.0/10
9	5806.44	CHS 195.0/10
9.25	5967.73	CHS 200.0/10
9.5	6129.02	CHS 205.0/10
9.75	6290.31	CHS 210.0/10
10	6451.6	CHS 215.0/10
10.25	6612.89	CHS 220.0/10
10.5	6774.18	CHS 226.0/10
10.75	6935.47	CHS 231.0/10
11	7096.76	CHS 236.0/10
11.25	7258.05	CHS 241.0/10
11.5	7419.34	CHS 246.0/10
11.75	7580.63	CHS 251.0/10
12	7741.92	CHS 256.0/10
12.25	7903.21	CHS 262.0/10
12.5	8064.5	CHS 267.0/10
12.75	8225.79	CHS 272.0/10
13	8387.08	CHS 277.0/10
13.25	8548.37	CHS 282.0/10
13.5	8709.66	CHS 287.0/10
13.75	8870.95	CHS 292.0/10
14	9032.24	CHS 298.0/10
14.25	9193.53	CHS 303.0/10
14.5	9354.82	CHS 308.0/10
14.75	9516.11	CHS 313.0/10
15	9677.4	CHS 318.0/10
15.25	9838.69	CHS 323.0/10

15.5	9999.98	CHS 328.0/10
15.75	10161.27	CHS 333.0/10
16	10322.56	CHS 339.0/10
16.25	10483.85	CHS 344.0/10
16.5	10645.14	CHS 349.0/10
16.75	10806.43	CHS 354.0/10
17	10967.72	CHS 359.0/10
17.25	11129.01	CHS 364.0/10
17.5	11290.3	CHS 369.0/10
17.75	11451.59	CHS 375.0/10
18	11612.88	CHS 380.0/10
18.25	11774.17	CHS 385.0/10
18.5	11935.46	CHS 390.0/10

18.75	12096.75	CHS 395.0/10
19	12258.04	CHS 400.0/10
19.25	12419.33	CHS 405.0/10
19.5	12580.62	CHS 410.0/10
19.75	12741.91	CHS 416.0/10
20	12903.2	CHS 421.0/10
20.25	13064.49	CHS 426.0/10
20.5	13225.78	CHS 431.0/10
20.75	13387.07	CHS 436.0/10
21	13548.36	CHS 441.0/10
21.25	13709.65	CHS 446.0/10
21.5	13870.94	CHS 452.0/10
21.75	14032.23	CHS 457.0/10

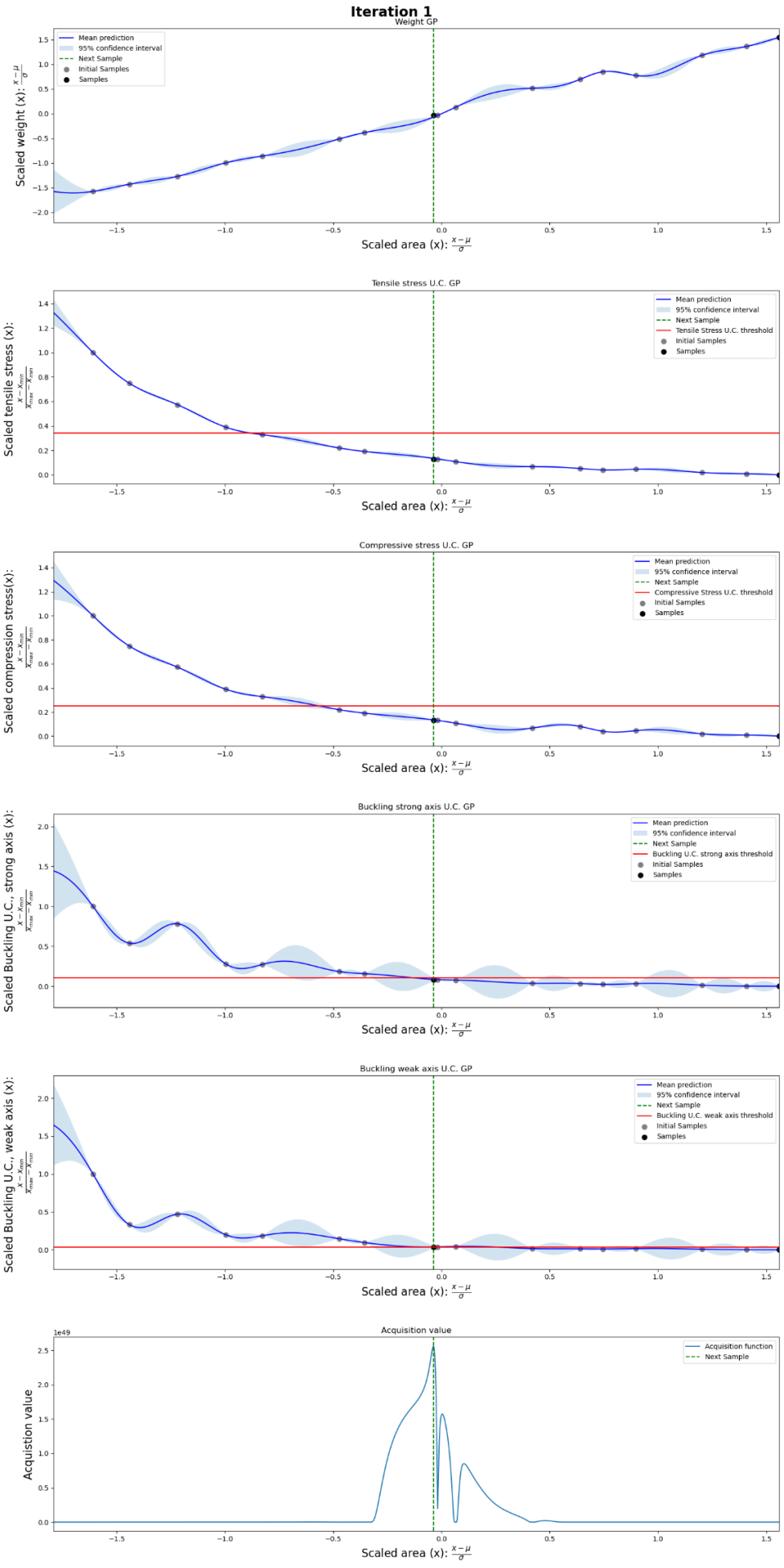


Figure 131: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 1.

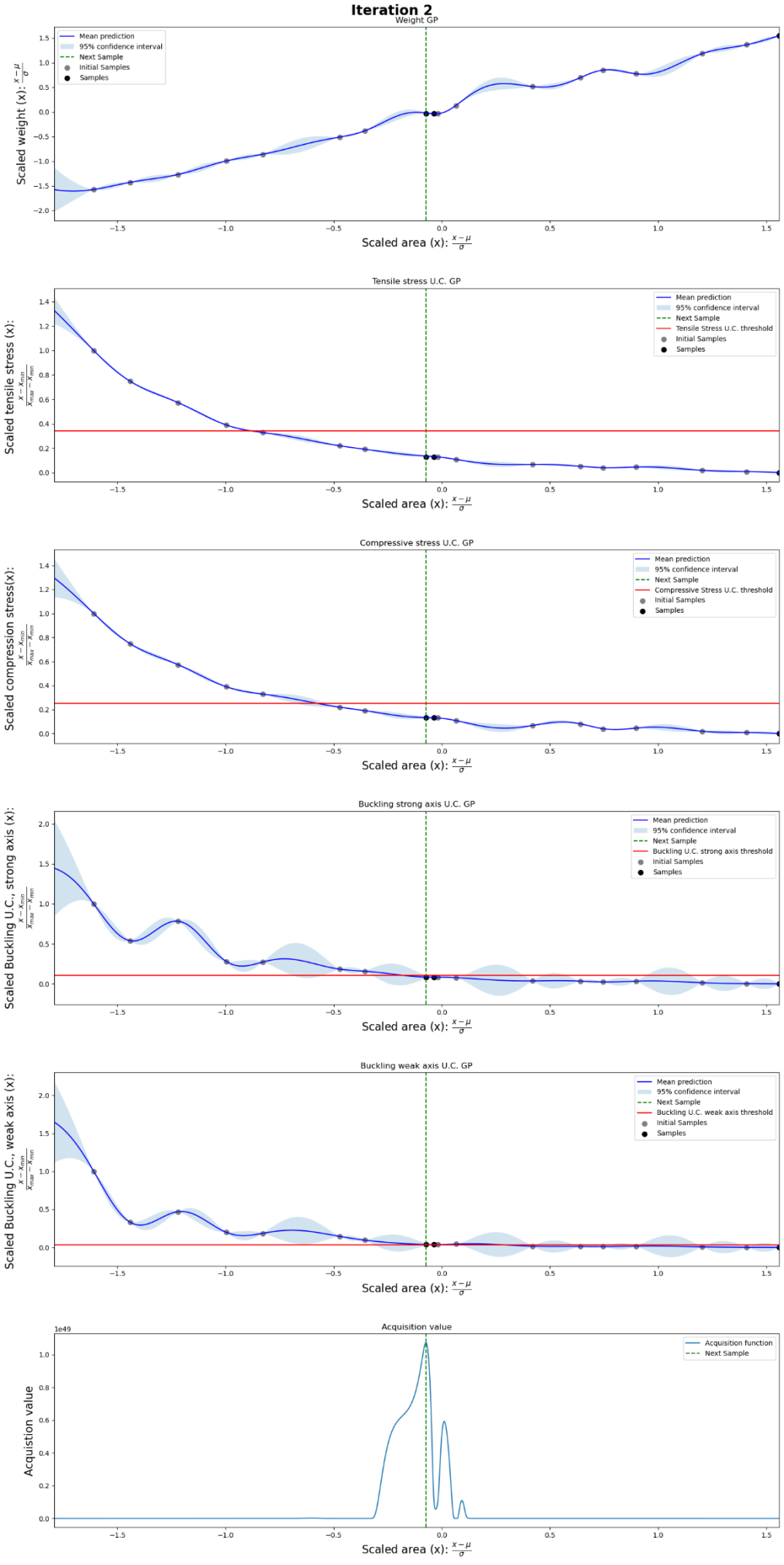


Figure 132: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 2.

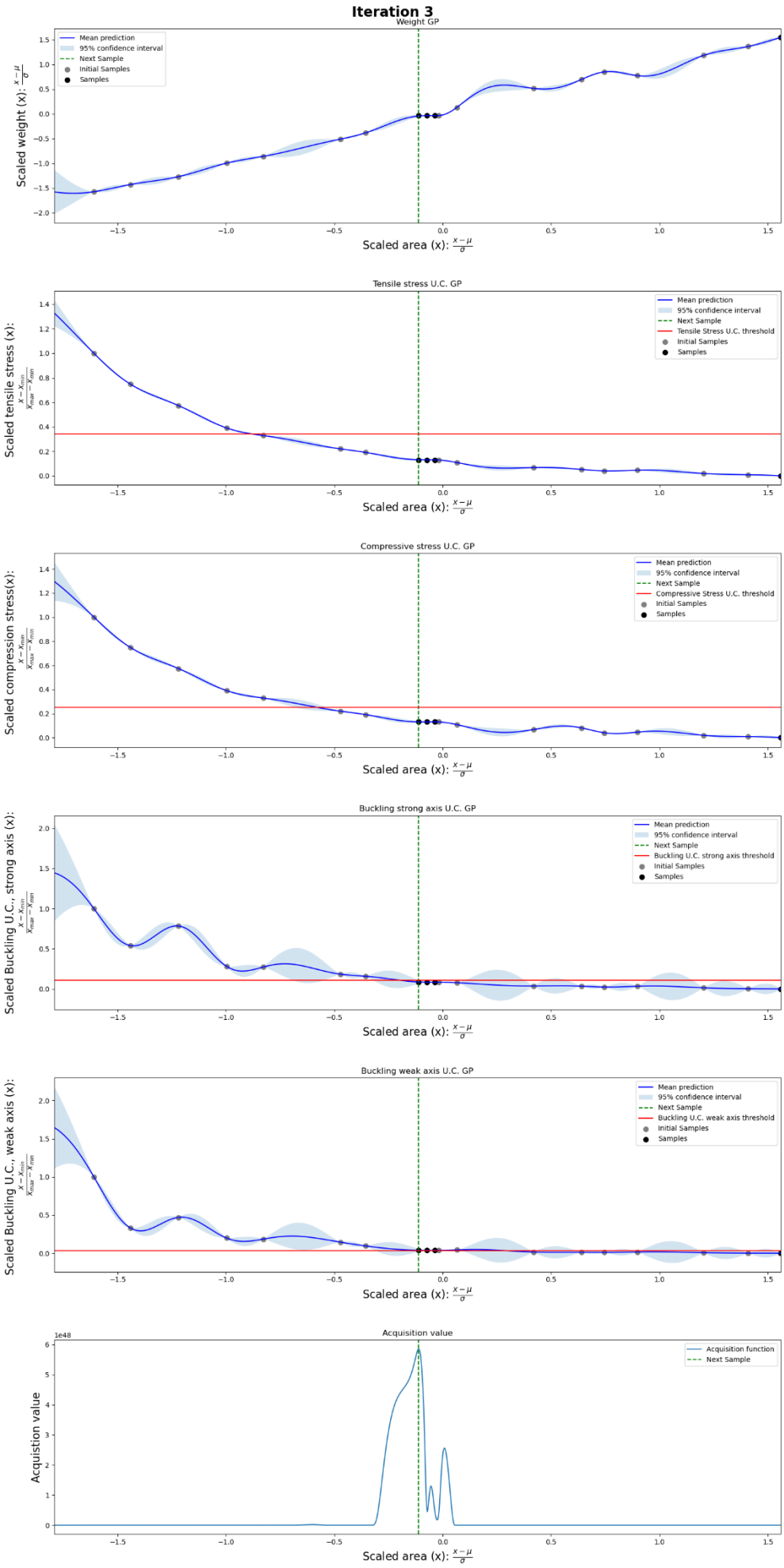


Figure 133: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 3.

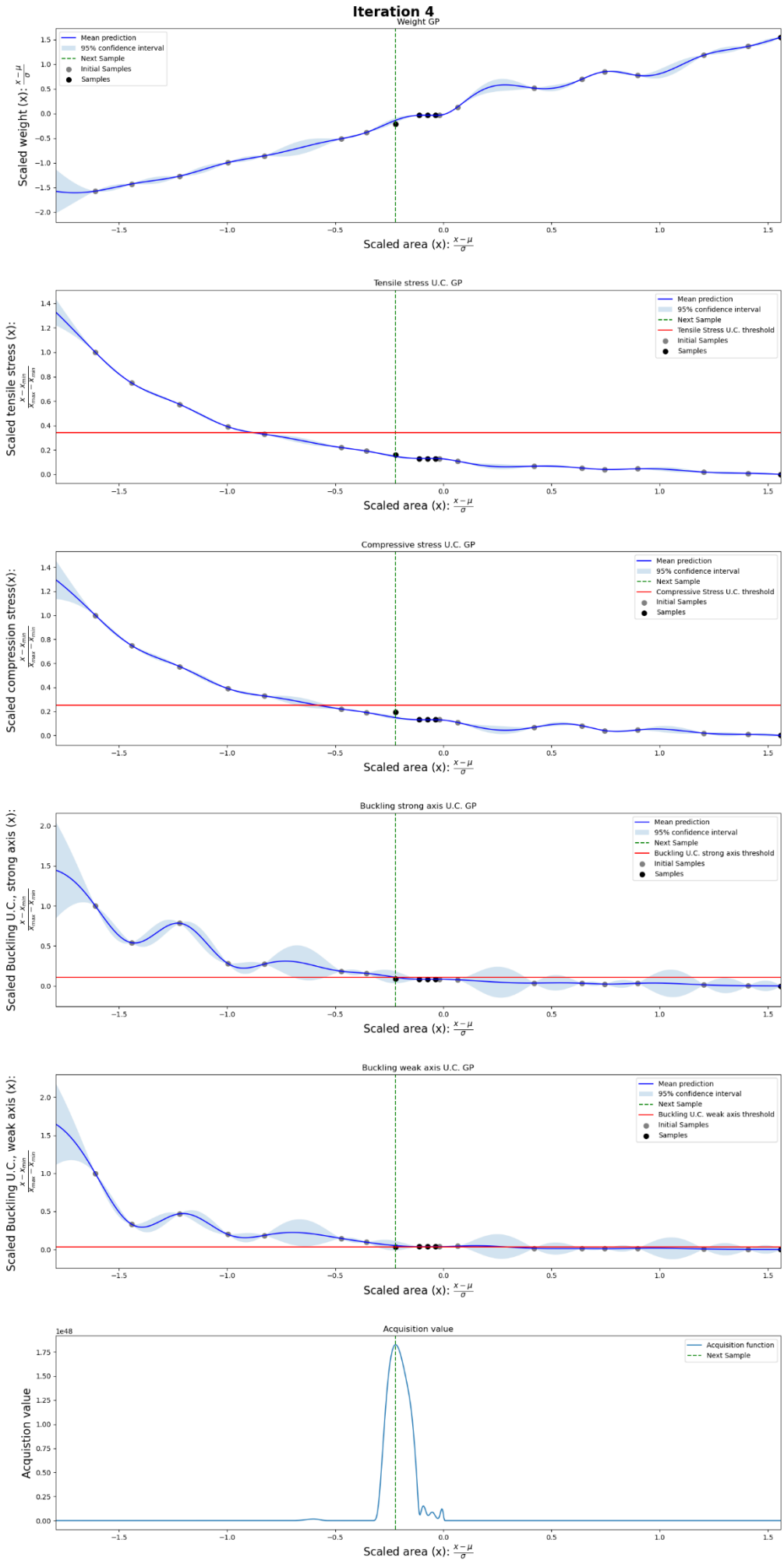


Figure 134: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 4.

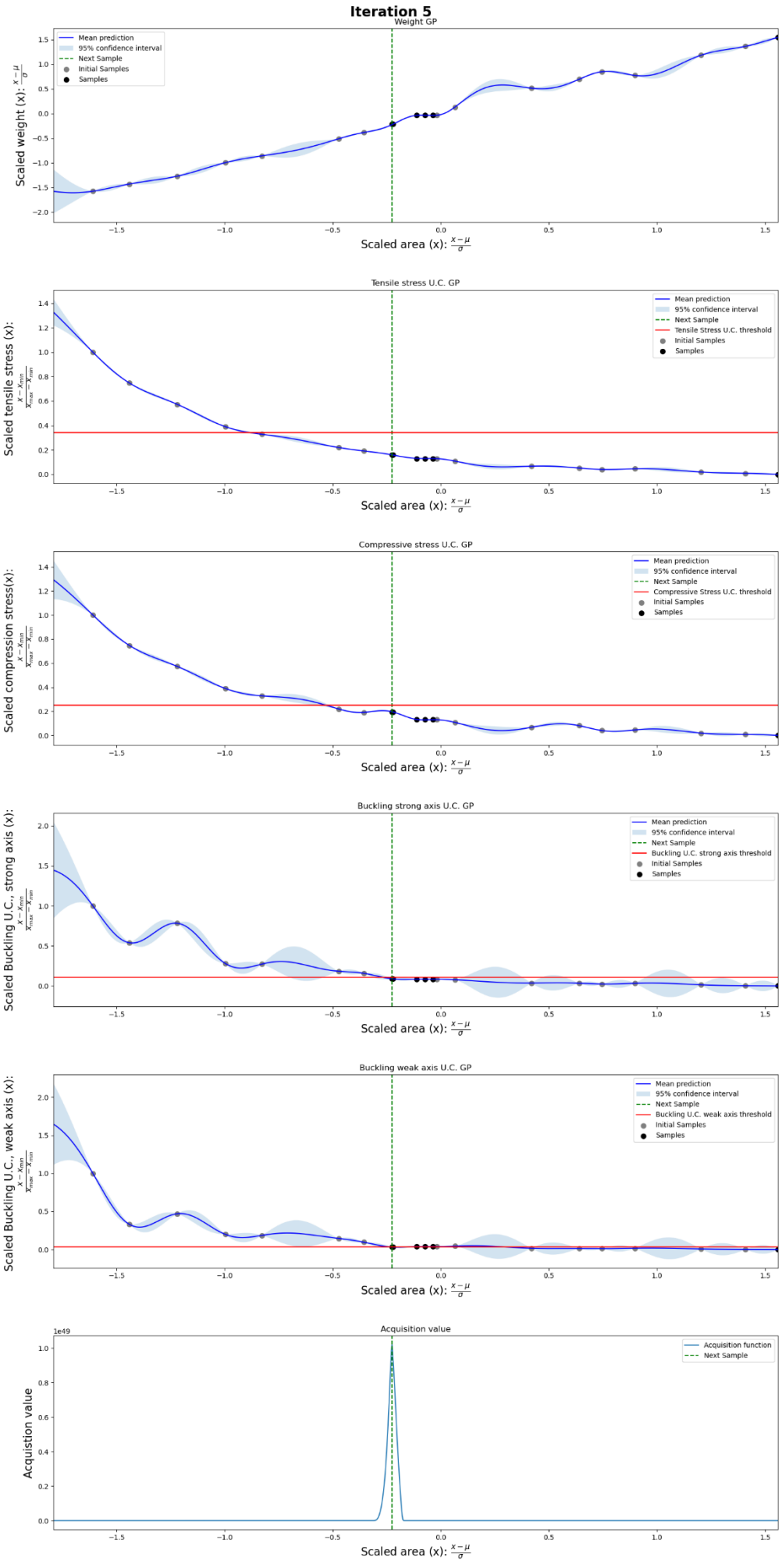


Figure 135: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 5.

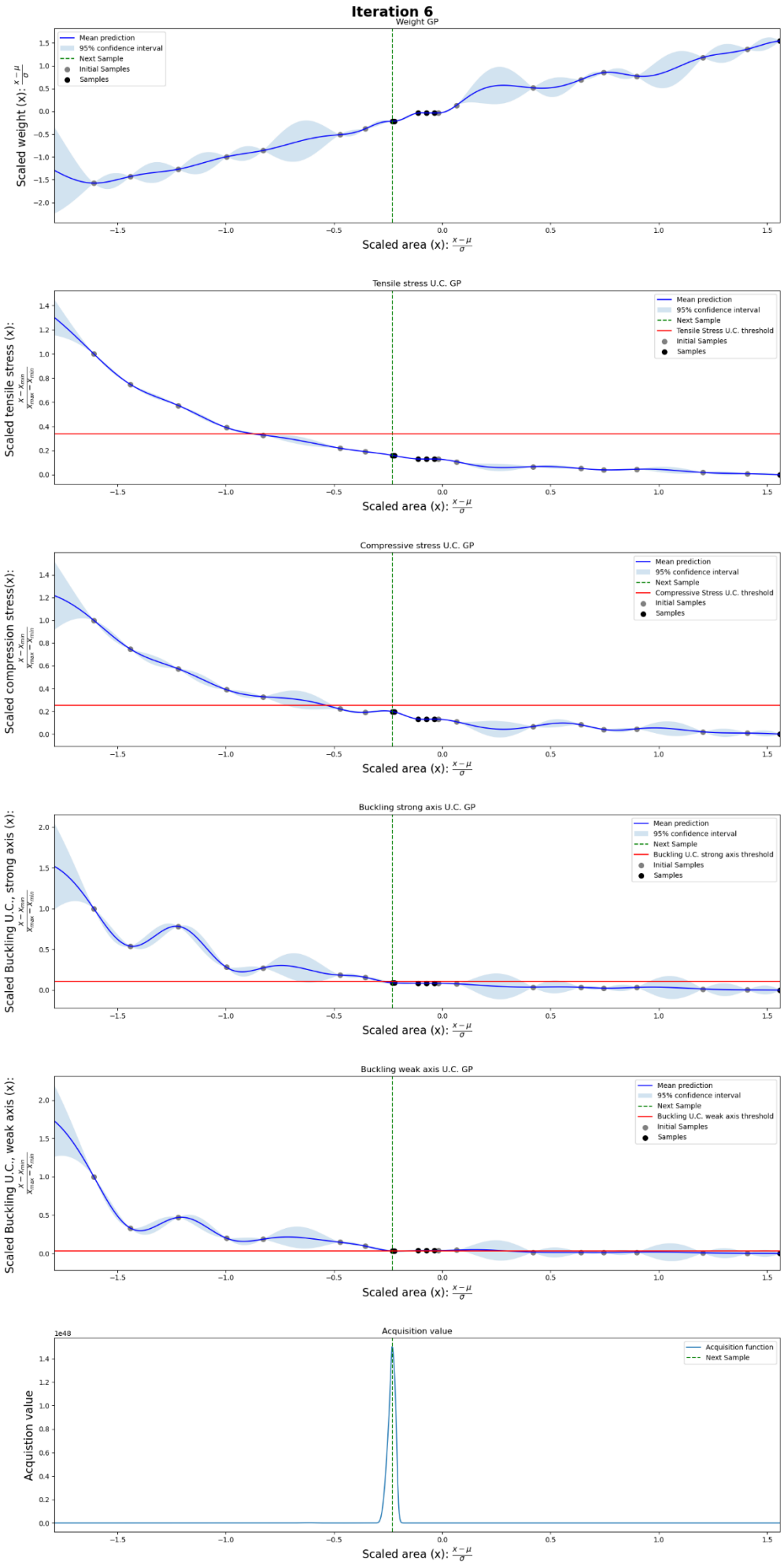


Figure 136: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 6.

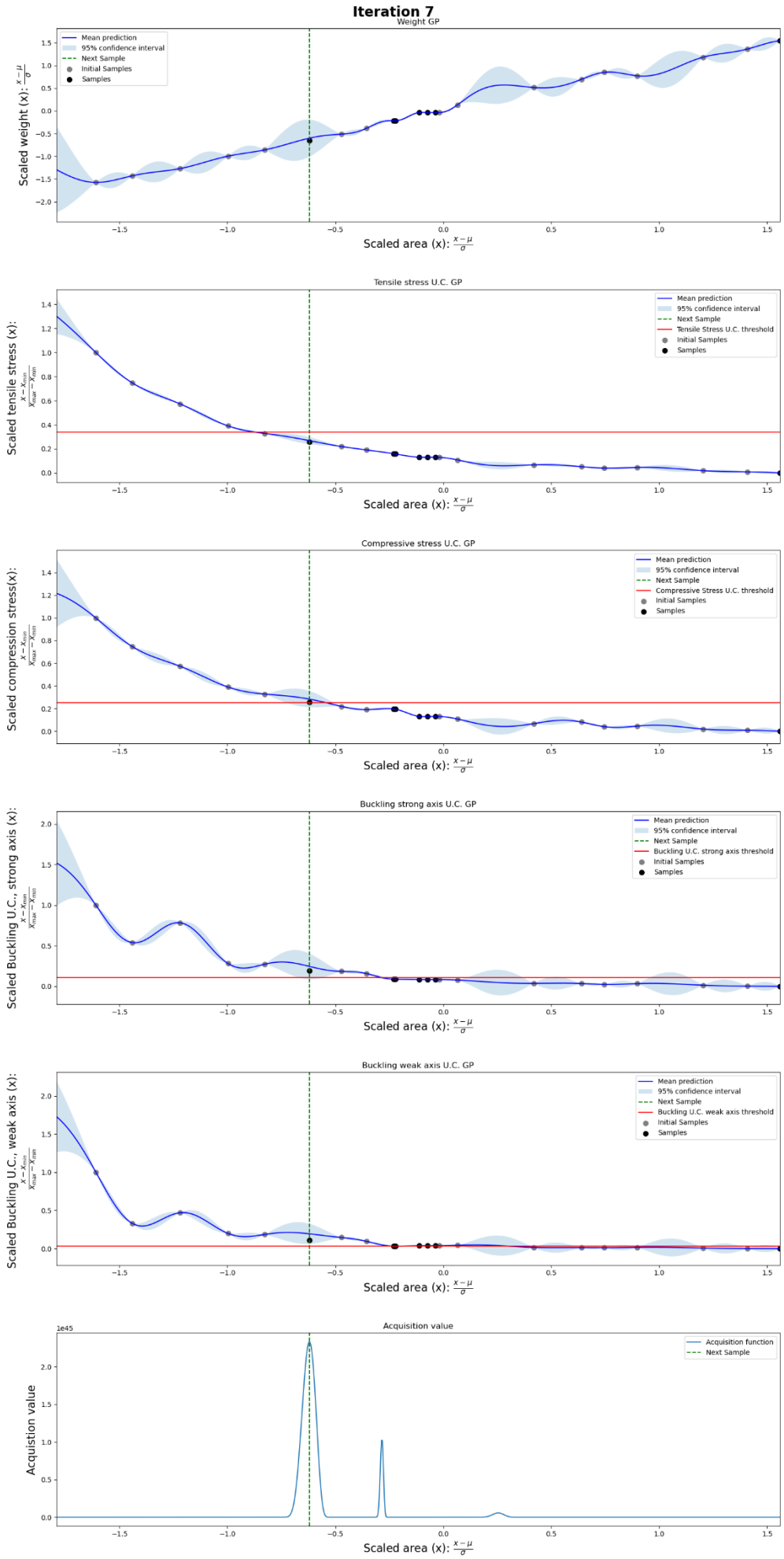


Figure 137: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 7.

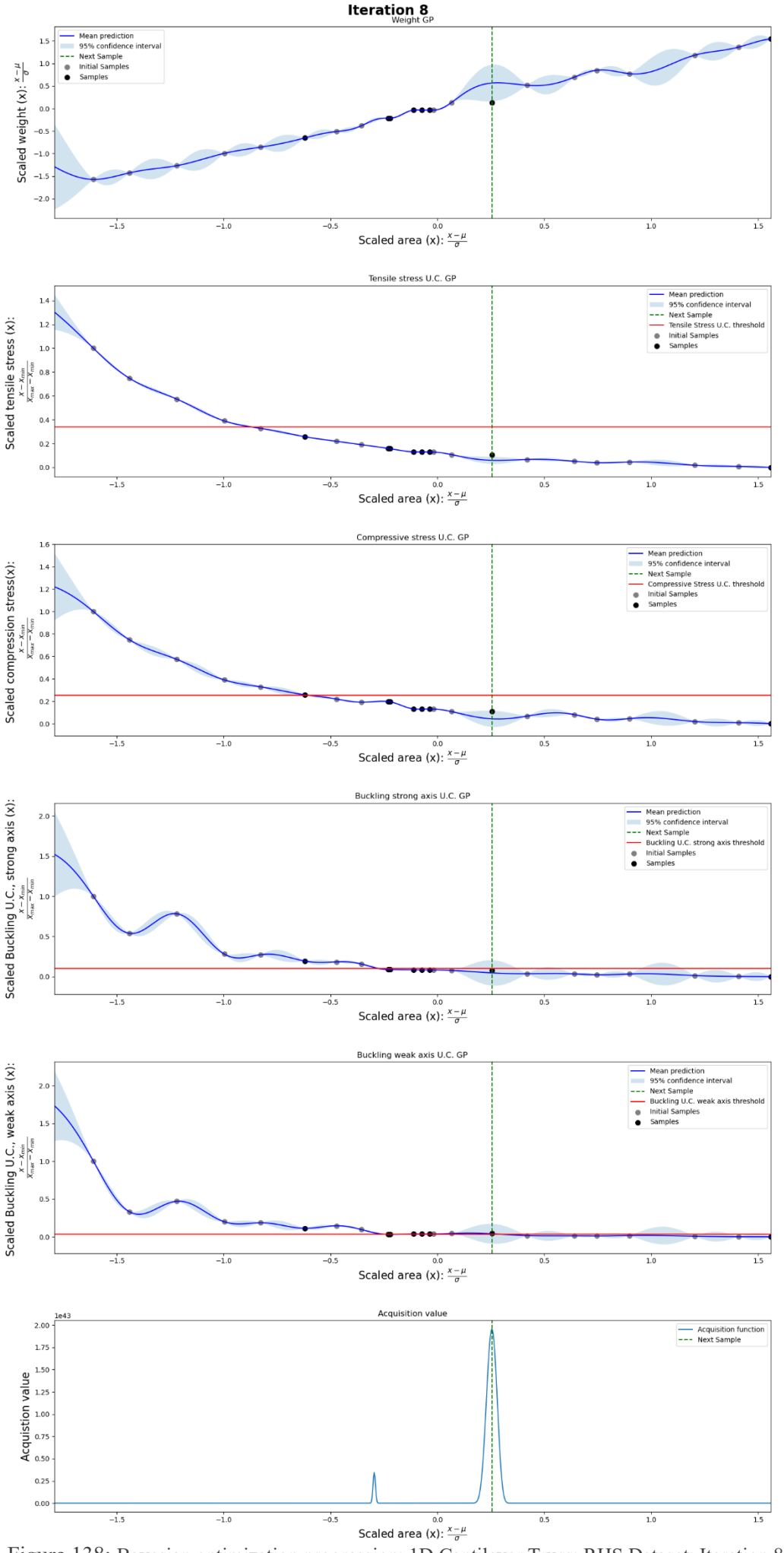


Figure 138: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 8.

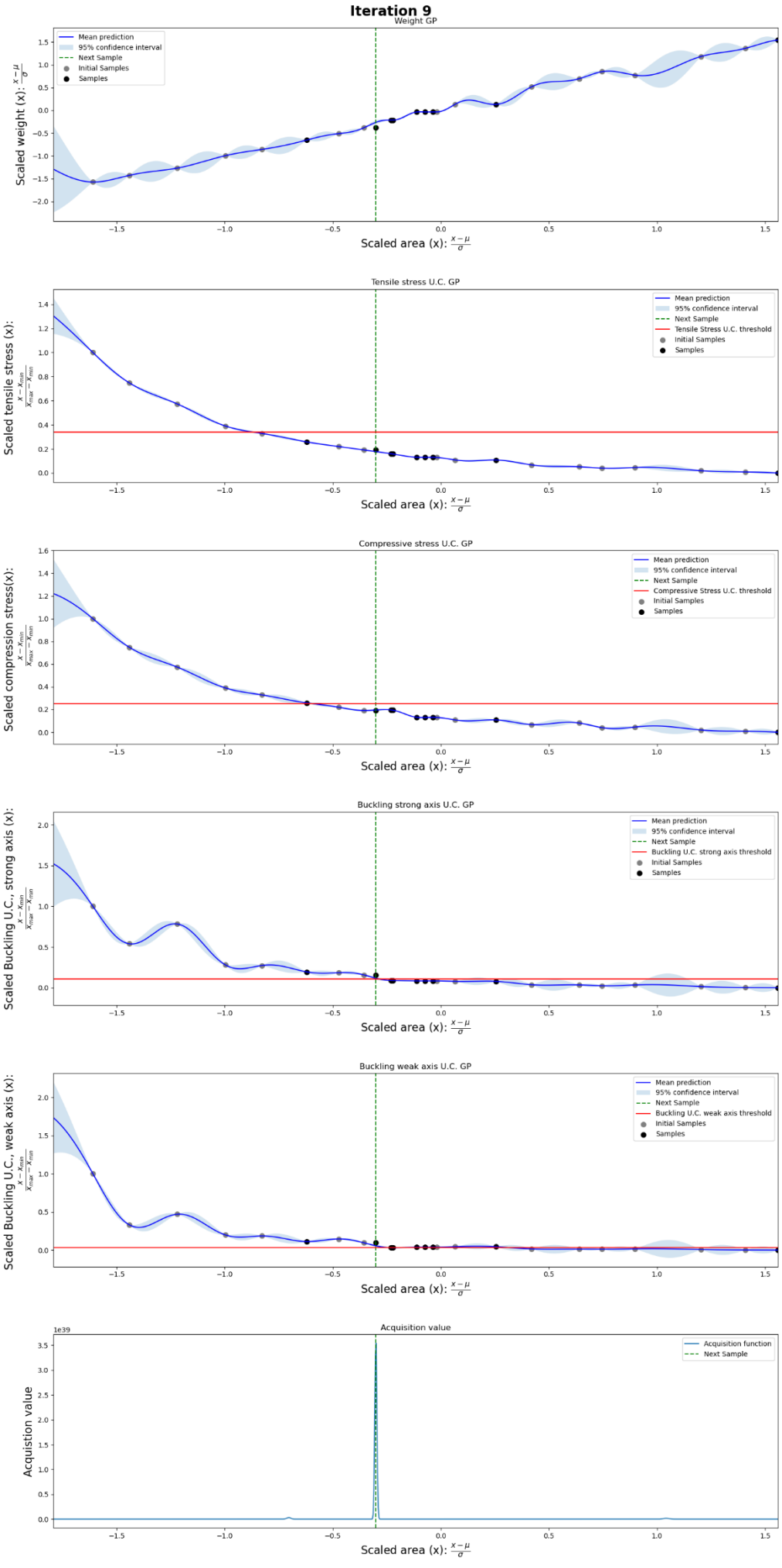


Figure 139: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 9.

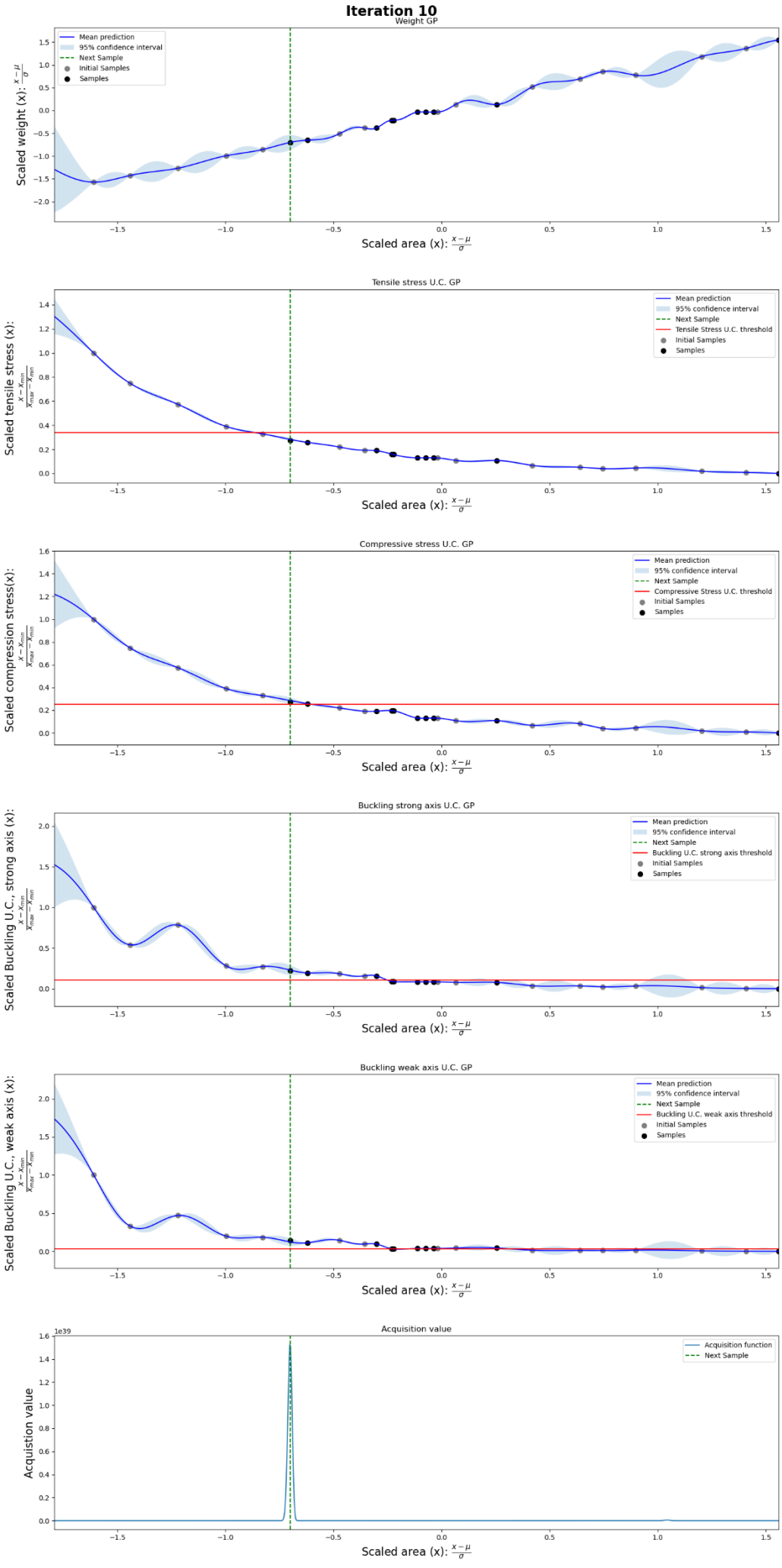


Figure 140: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 10.

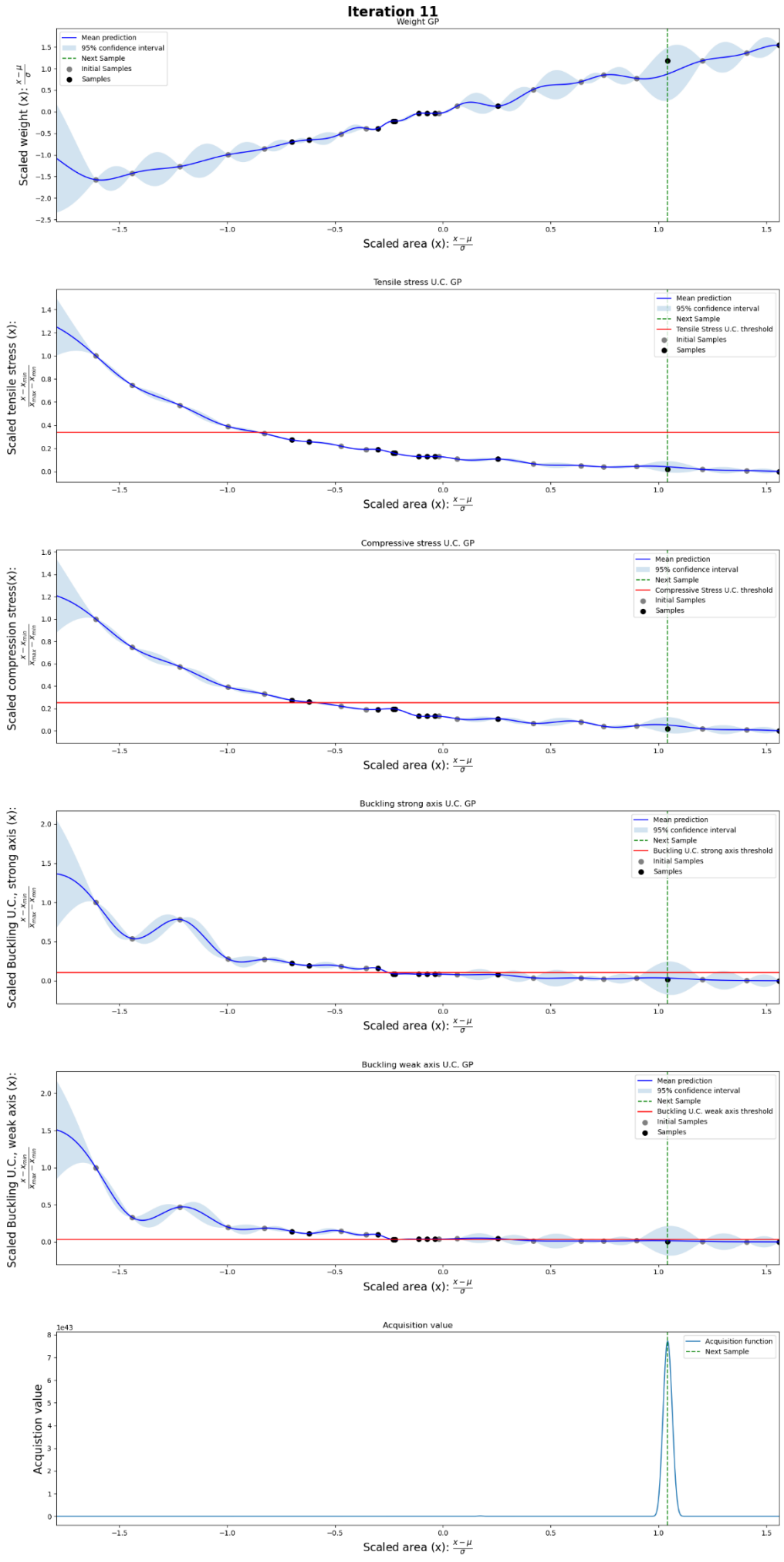


Figure 141: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 11.

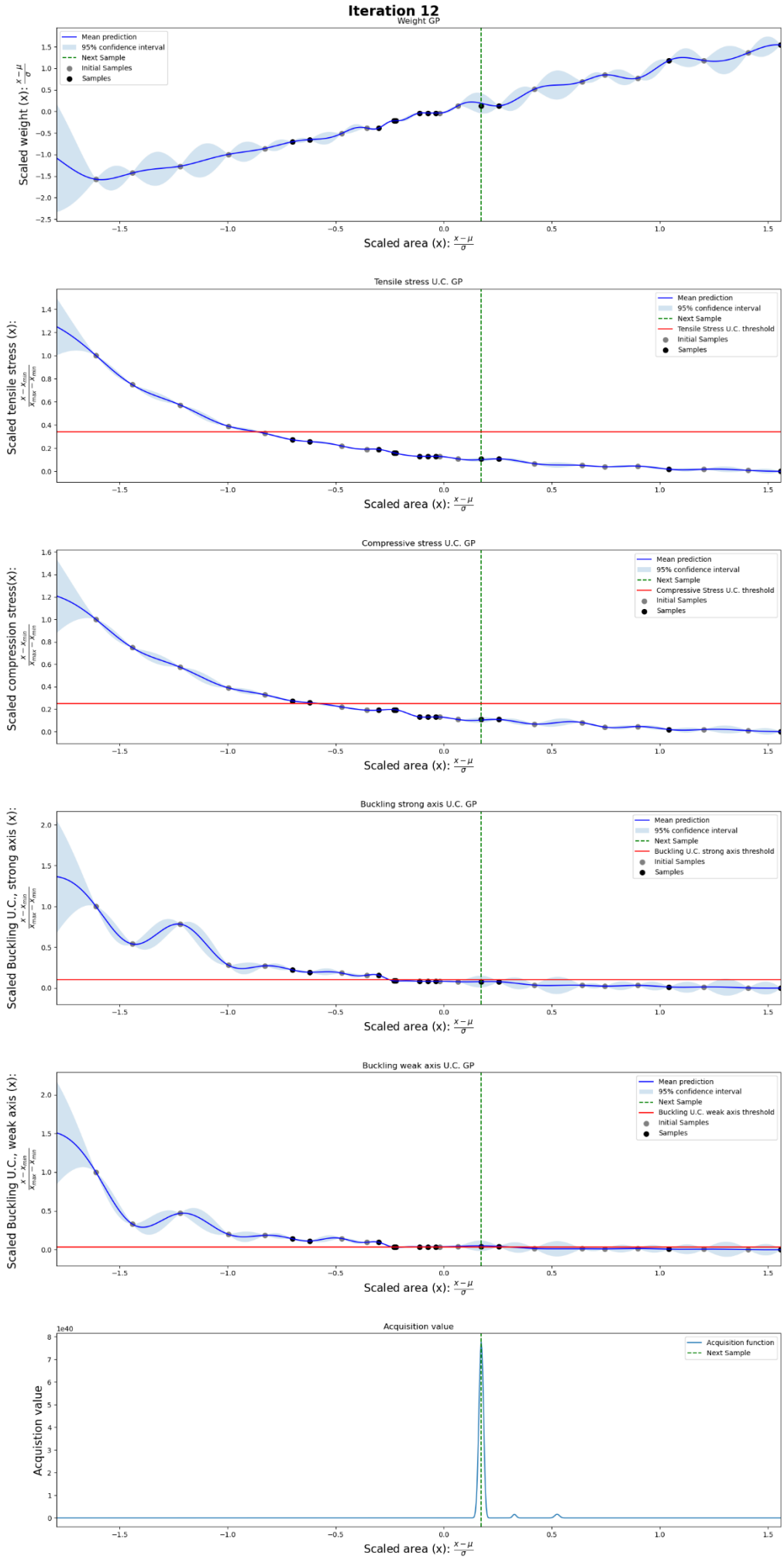


Figure 142: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 12.

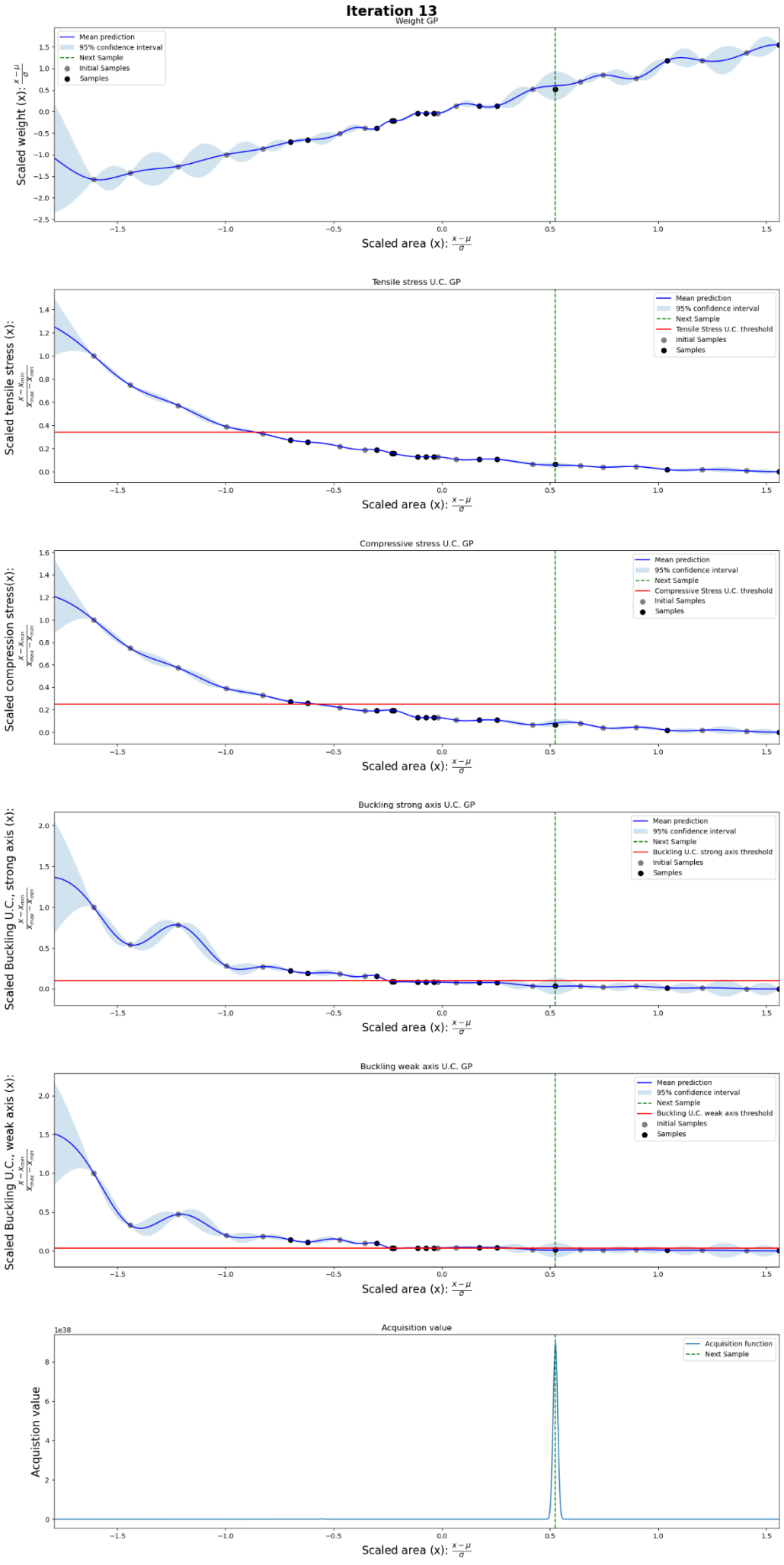


Figure 143: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 13.

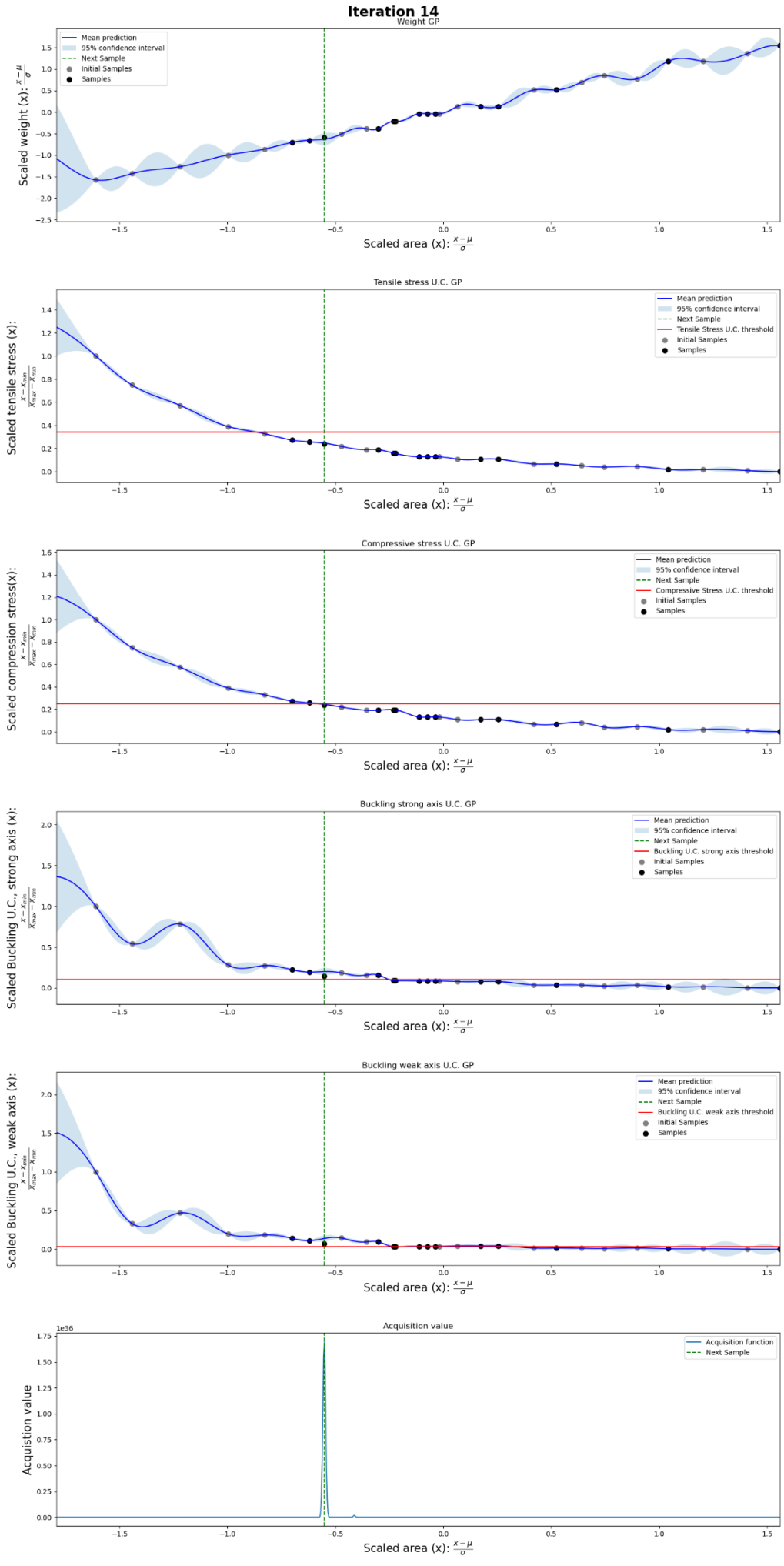


Figure 144: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 14.

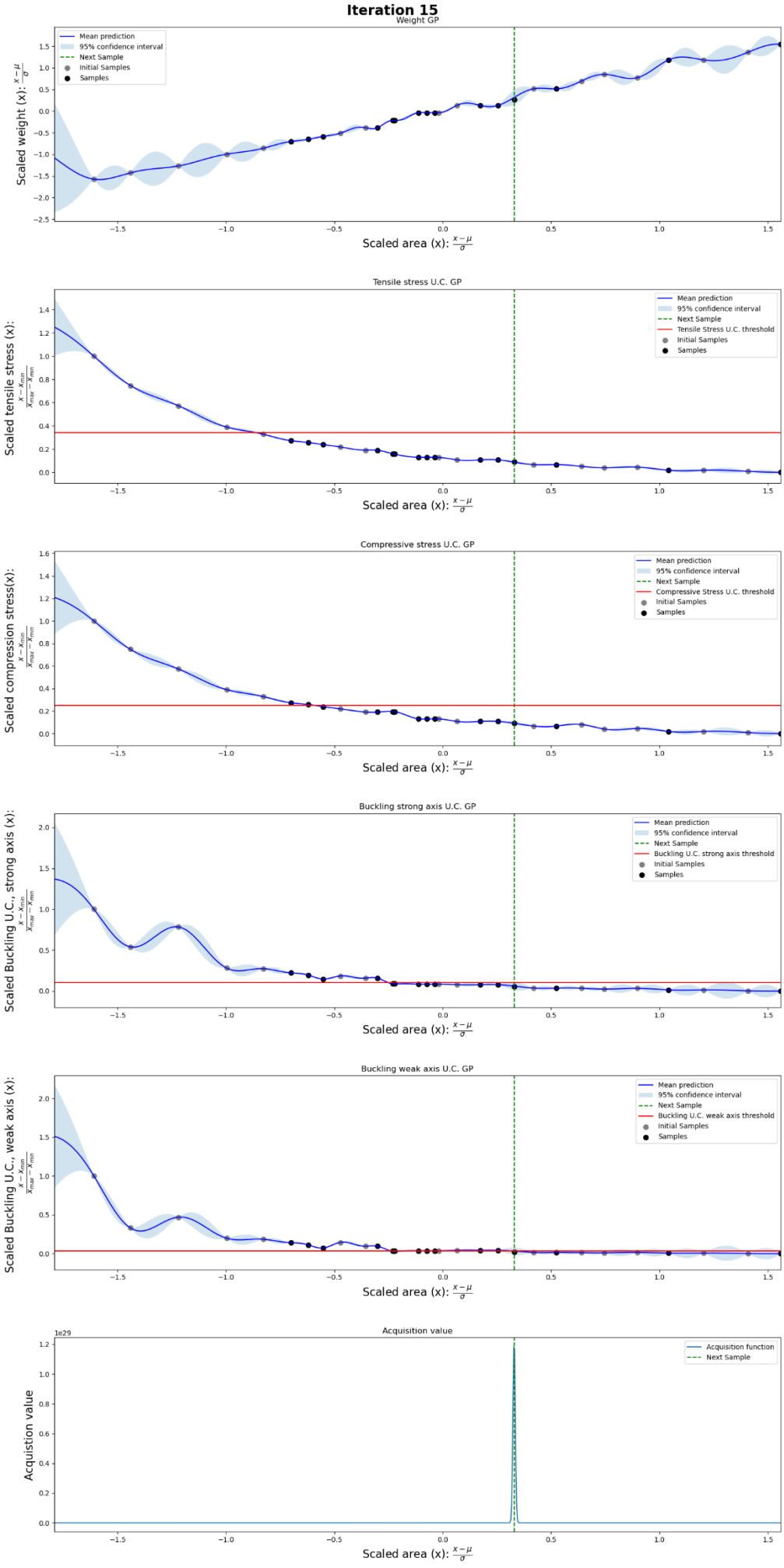


Figure 145: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 15.

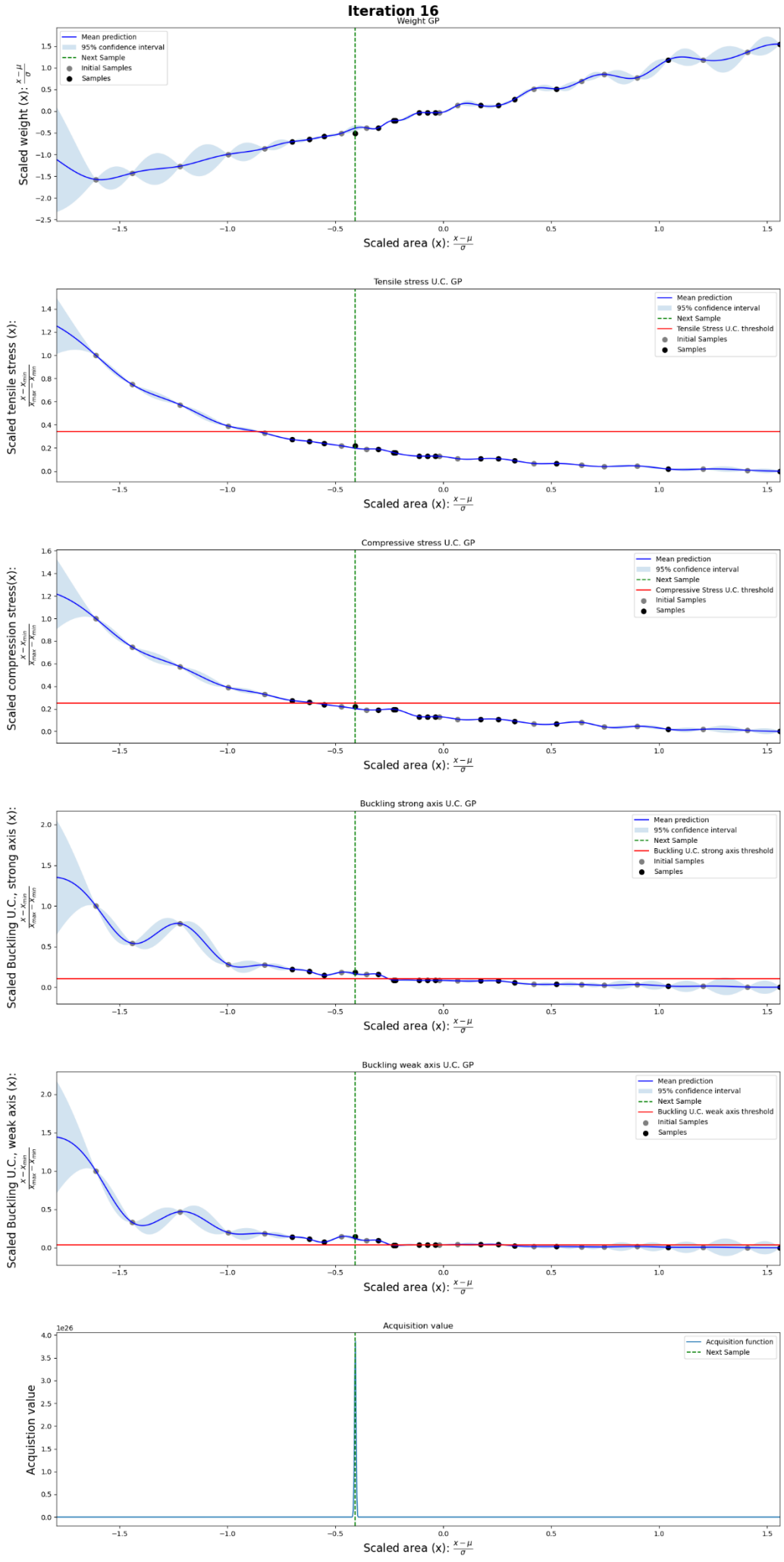


Figure 146: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 16.

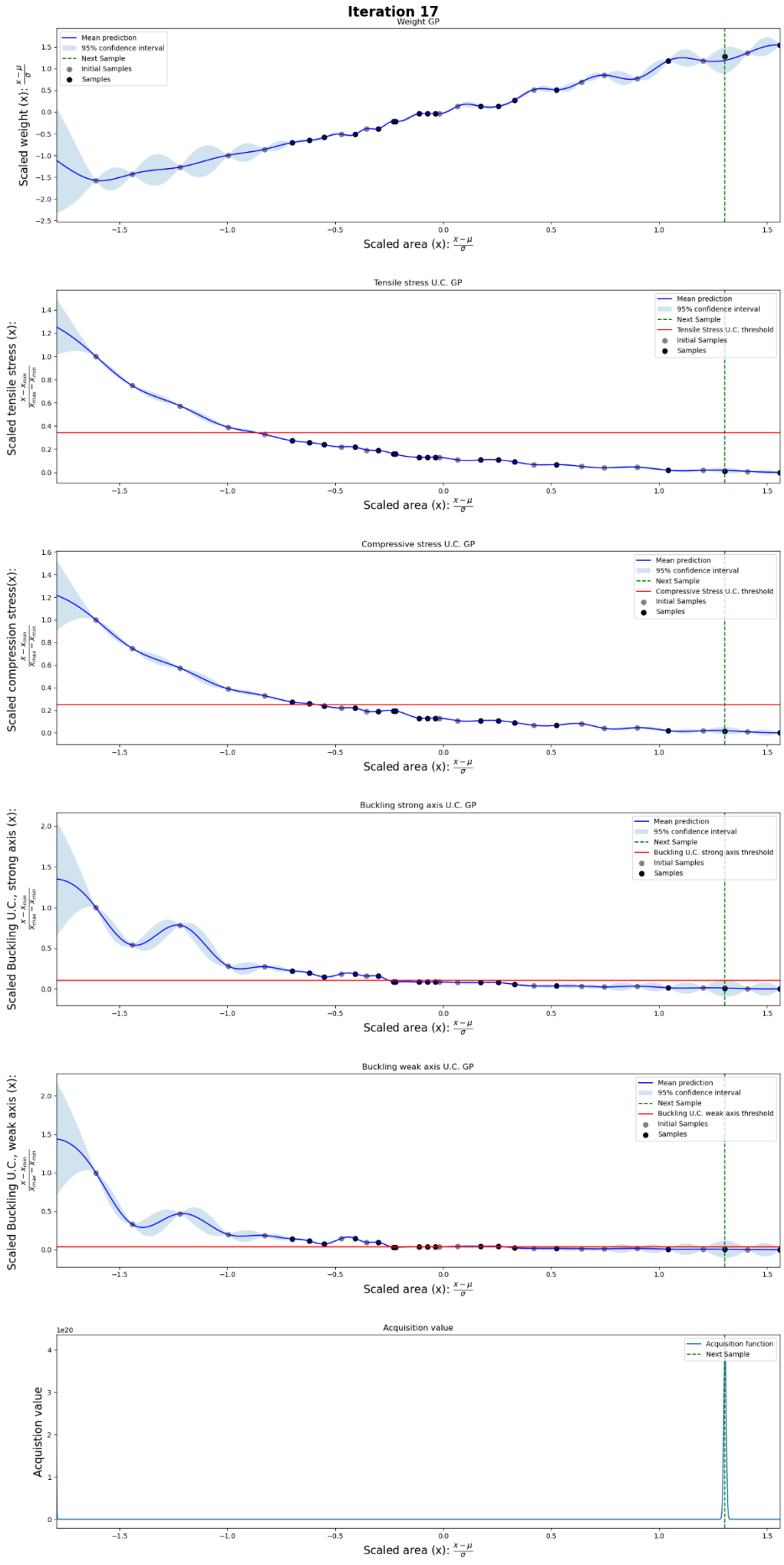


Figure 147: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 17.

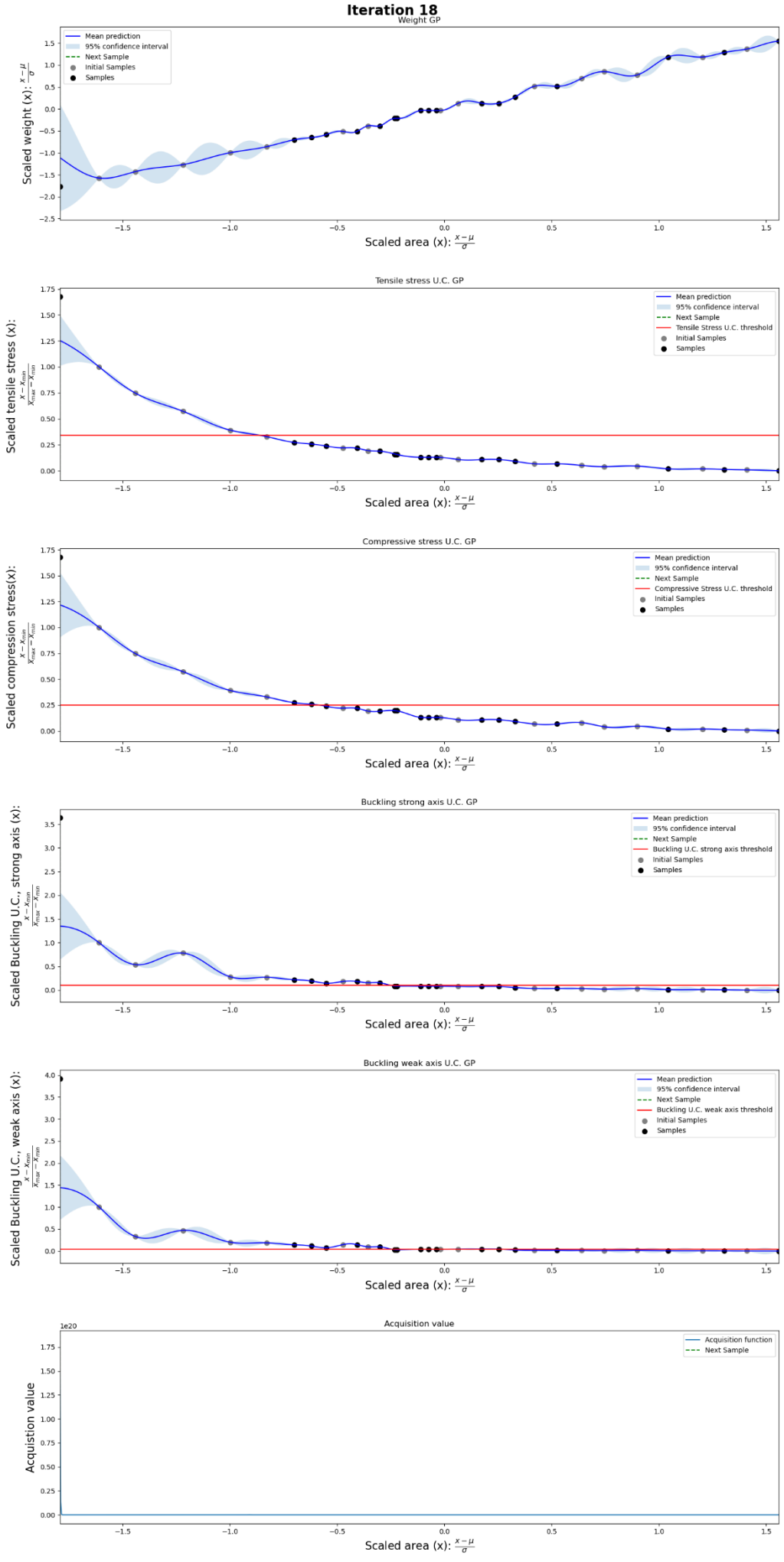


Figure 148: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 18.

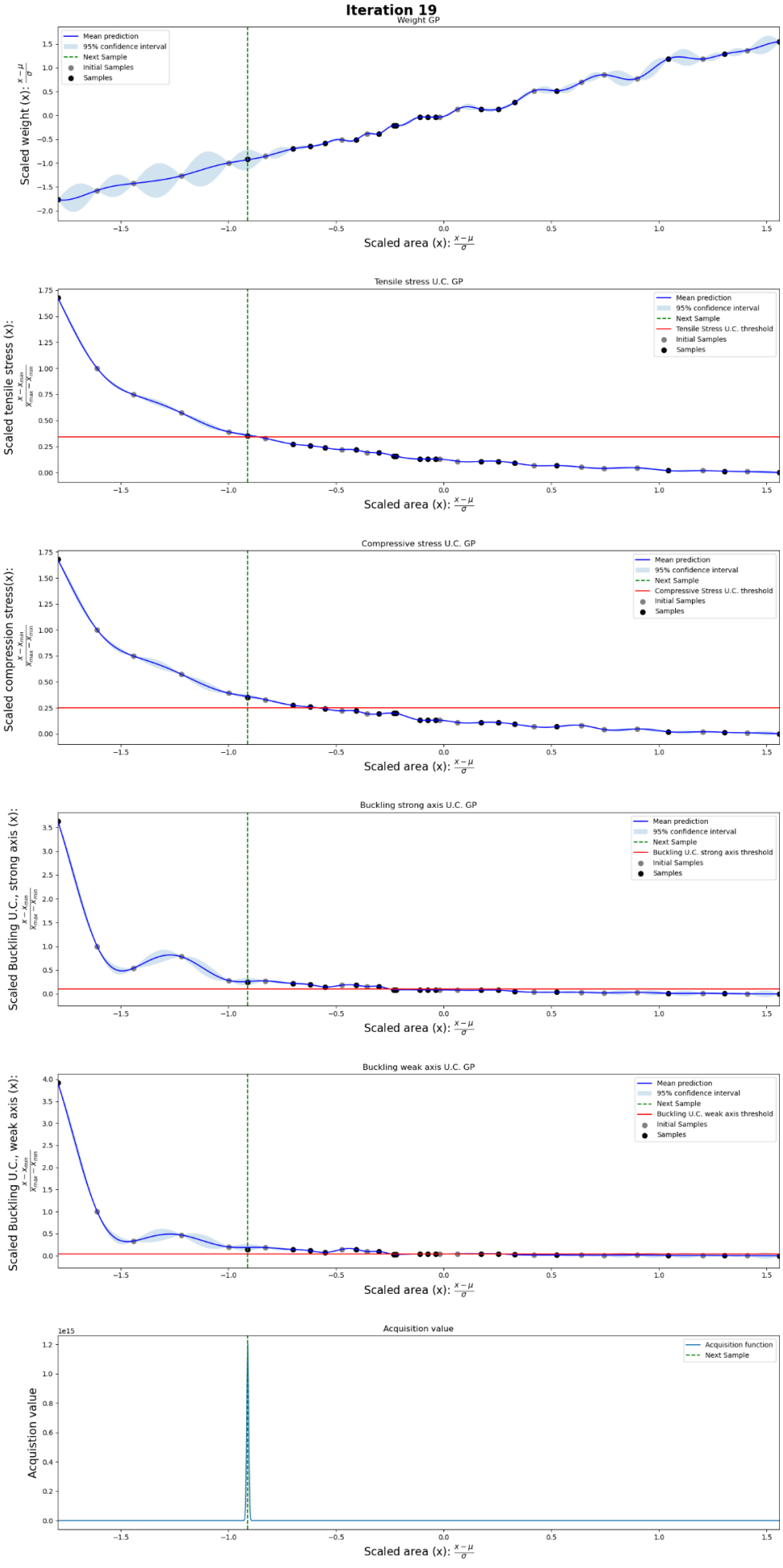


Figure 149: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 19.

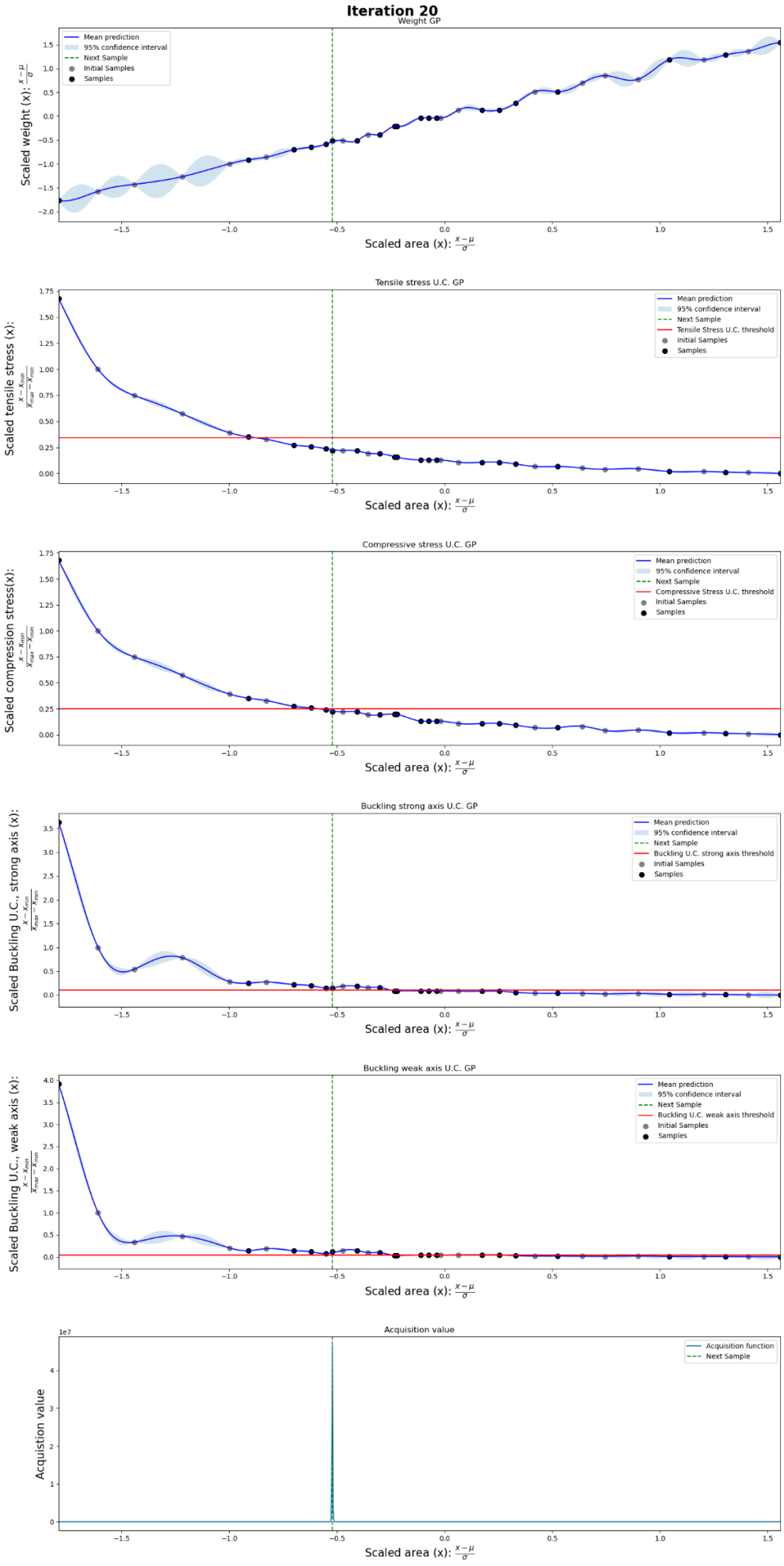


Figure 150: Bayesian optimization progression: 1D Cantilever Truss; RHS Dataset; Iteration 20.

Appendix D: 3D optimization problem: Gridshells

In the figures of D1 and D2, the load combinations used for the analysis in Chapter 5.3 and Chapter 5.4 are shown. The load per surface is uniformly distributed over the profiles that enclose the surface based on surface area divided by perimeter;

The load per profile is summed from the surfaces that it helps define (i.e., the surfaces to the left and right of the line, where applicable).

The line load is determined with:

$$q_{imposed}\left[\frac{kN}{m}\right] = \frac{A_{surface}[m^2] * Q_{glass}[\frac{kN}{m^2}]}{P_{surface}[m]}$$

D1. 4x4 Gridshell Additional Figures

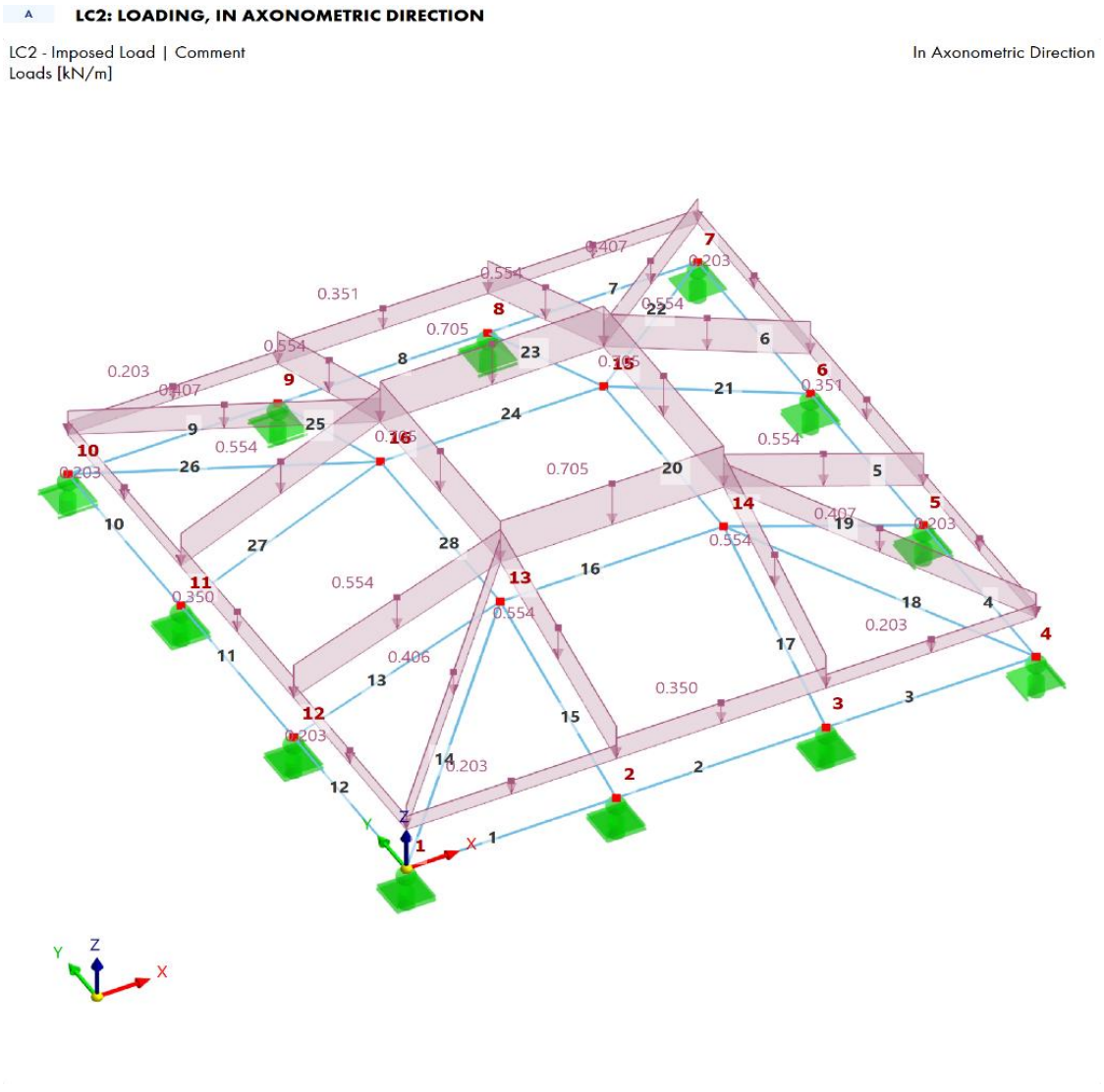


Figure 151: Gridshell 4×4 node model: Validation load combination: self-weight + imposed load of 1kN/m^2.

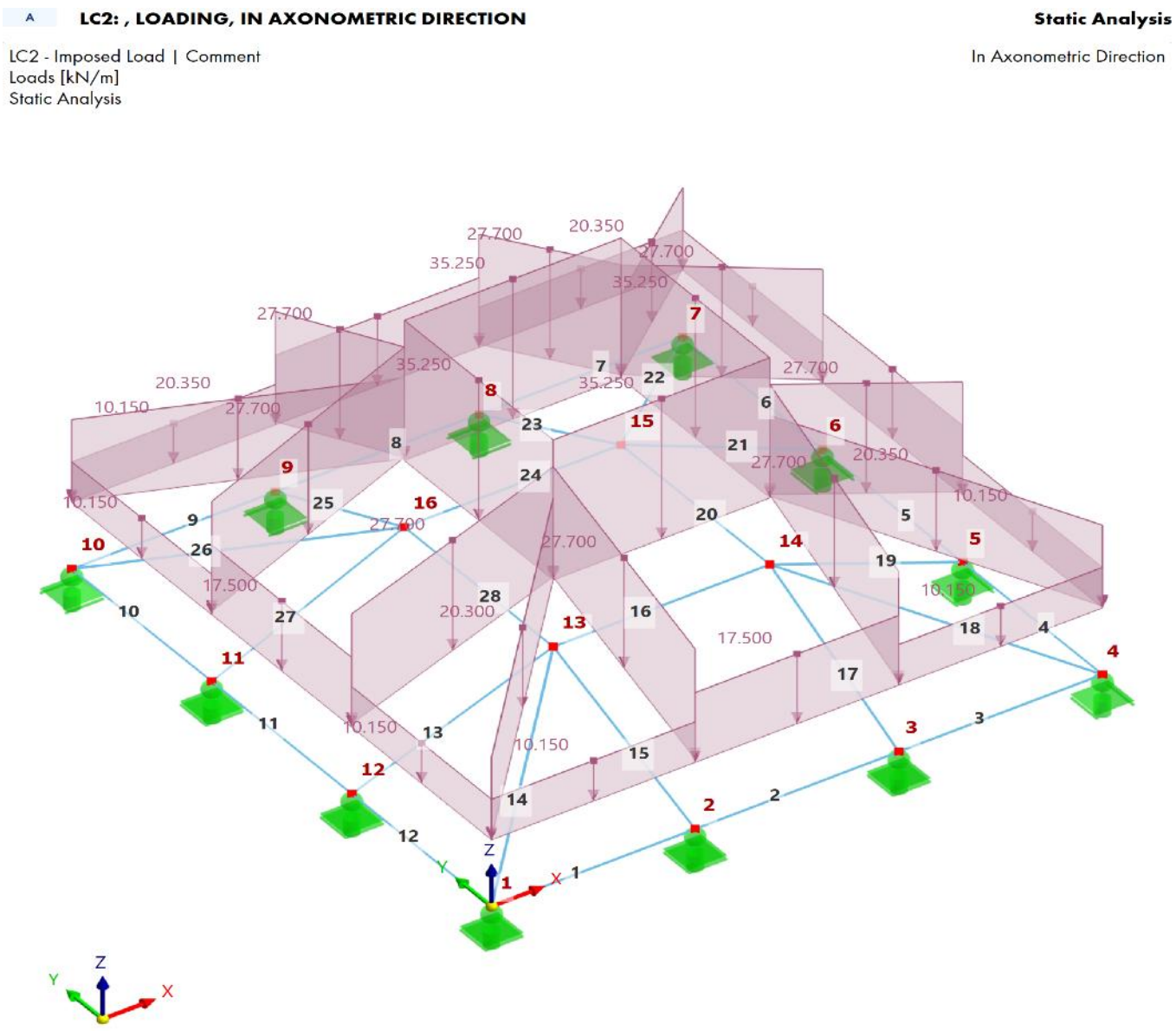


Figure 152: Gridshell 4×4 node model: Test load combination: 1.35 * self-weight + 1.35 * 50 * imposed load of 1kN/m^2.

The heatmaps in Figure 152 to Figure 155 present the learned lengthscales per input dimension for multiple Gaussian Process kernels, evaluated in the original design space without applying PCA. Each row corresponds to one kernel, and each column to one design dimension. The color intensity encodes the log-scaled lengthscale values:

- **Bright colors (yellow/green)**: short lengthscales → model is highly sensitive to changes in that input dimension.
- **Dark colors (blue/purple)**: long lengthscales → input dimension has little influence on the predictive model.

The heatmaps are produced immediately after the first optimization of the hyperparameters for the Gaussian processes following the fitting on the initial samples. In this way the effect of any subsequent samples generated from the acquisition function is disregarded.

In Figure 152 it can be seen that the GPs are correctly registering the most influential input dimension for each steel element. Since the first 12 elements share the same cross-section which is the first in the input vector the length scales are the shortest at index 0 as expected. The diagonal that forms from the kernel 13 to kernel 28 corresponds to the fact that each GP models the constraints for an individual element rather than a group of elements.

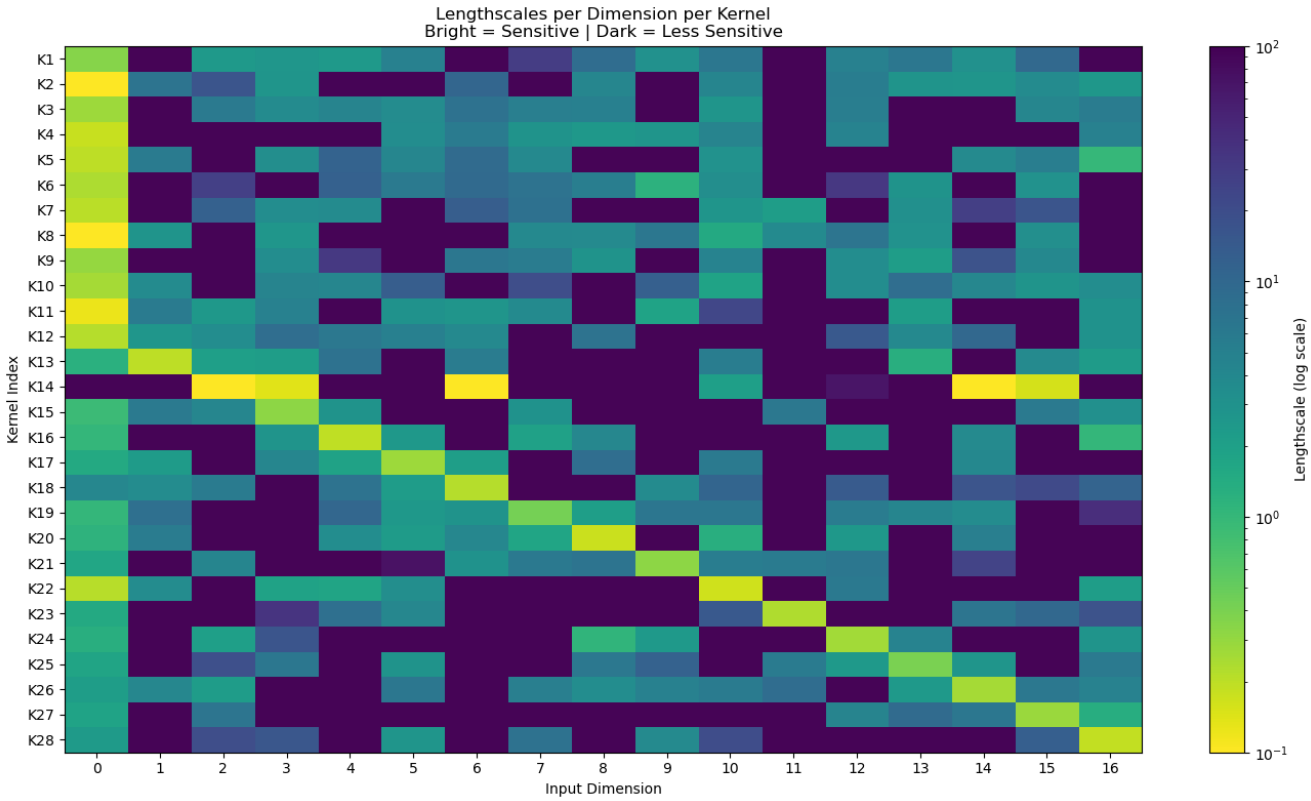


Figure 153: Heatmap of log-length scales across 17 input dimensions and 28 kernels. 17 initial samples.

With 28 input dimensions and only 280 training samples, the heatmap in Figure 154 reveals a sparse and somewhat noisy sensitivity pattern. The 28 individual GPs are able to form the expected diagonal with the input dimensions, indicating that since there are no groups of elements each structural element is dependent on the corresponding cross-sectional area. However, there is also other random dependencies as well. This suggests that with limited data, the GP struggles to robustly distinguish between relevant and irrelevant dimensions. In high-dimensional spaces, this effect is expected due to the *curse of dimensionality*, where the number of samples needed to represent the same part of the design space grows exponentially with every new dimension. Nevertheless, even at this low data regime the BO framework is able to produce reasonable designs leading to total weight reduction as discussed in subchapter 5.3.

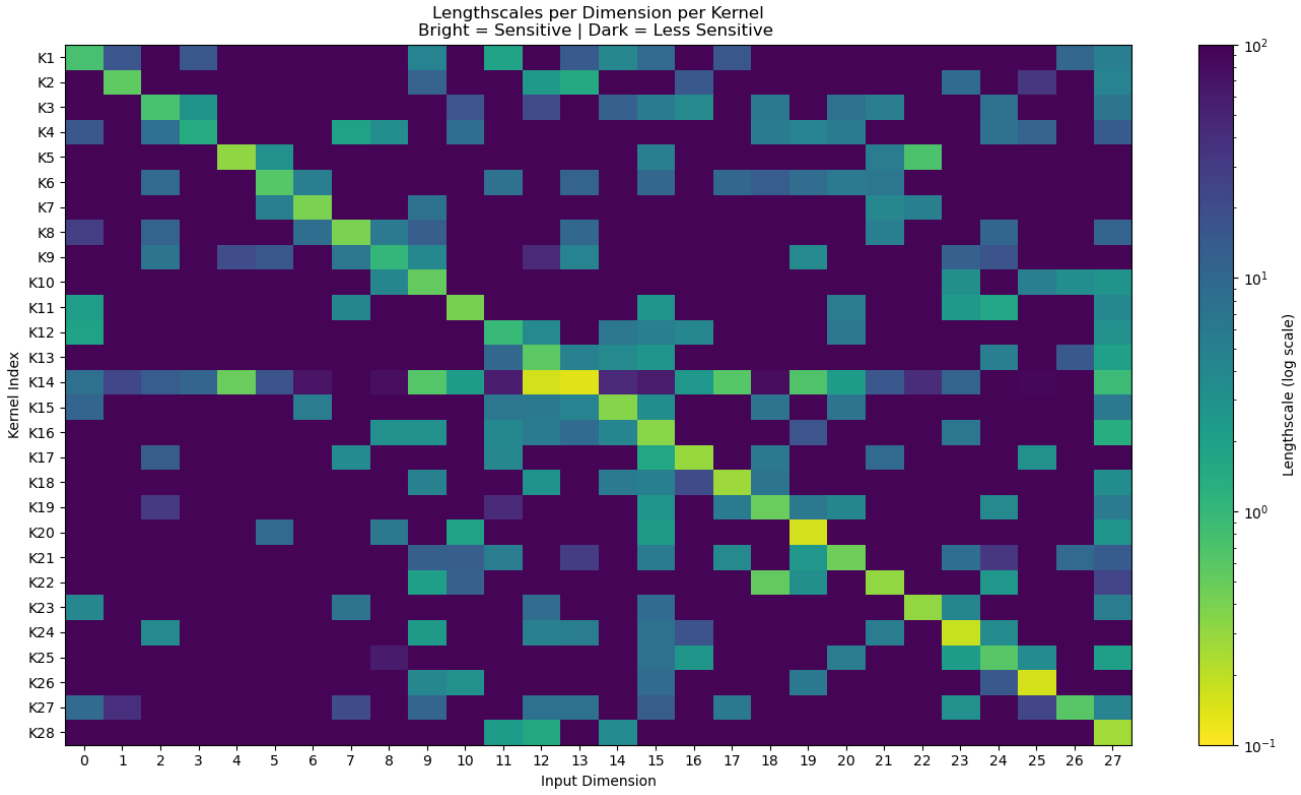


Figure 154: Heatmap of log-length scales across 28 input dimensions and 28 kernels. 280 initial samples.

Increasing the number of samples by an order of magnitude drastically clarifies the sensitivity structure. The heatmap in Figure 155 shows a much stronger diagonal dominance, where different kernels consistently identify specific input dimensions as relevant, with relatively stable short length scales. The noise observed in the 280-sample case is reduced, and irrelevant dimensions are more uniformly assigned long length scales (dark regions). This demonstrates that sample size has a direct impact on GP interpretability in high dimensions: with sufficient data, the model is better able to disentangle the influence of individual variables and establish a coherent sensitivity profile. However, the computational cost is substantially increased with execution time being around 15 hours compared to the ca. 1.6 hours as discussed in Figure 81 in subchapter 5.3.

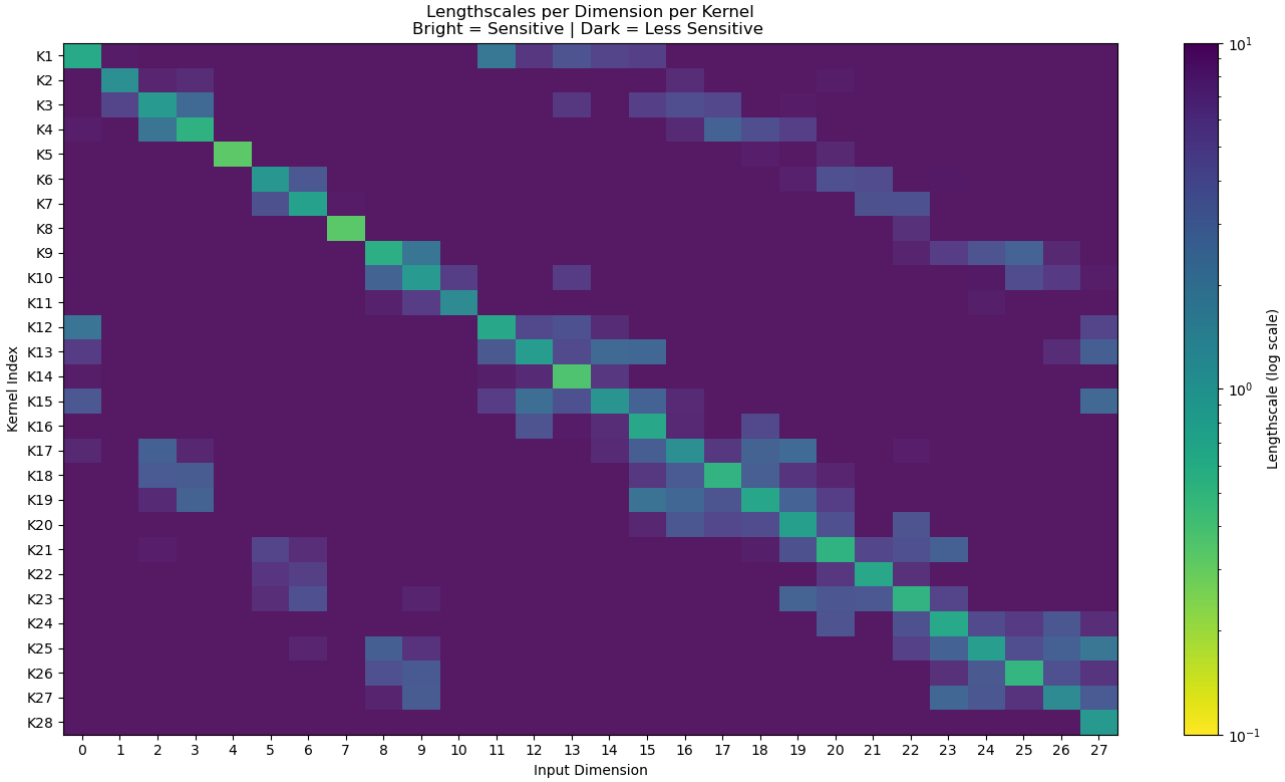


Figure 155: Heatmap of log-length scales across 28 input dimensions and 28 kernels. 2800 initial samples.

D2. 9x9 Gridshell Additional Figures

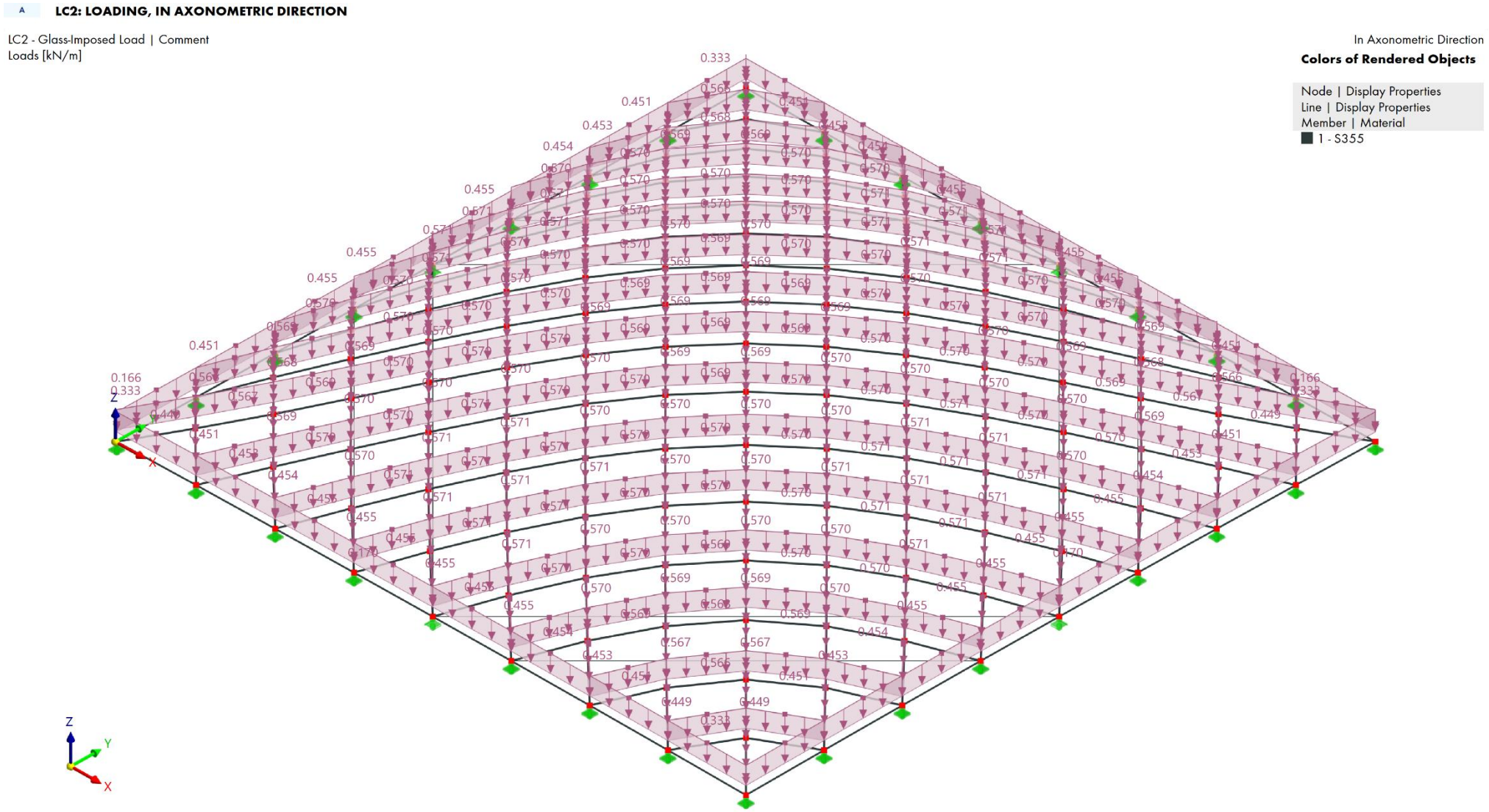


Figure 156: Glass imposed load applied along the true length of the elements.

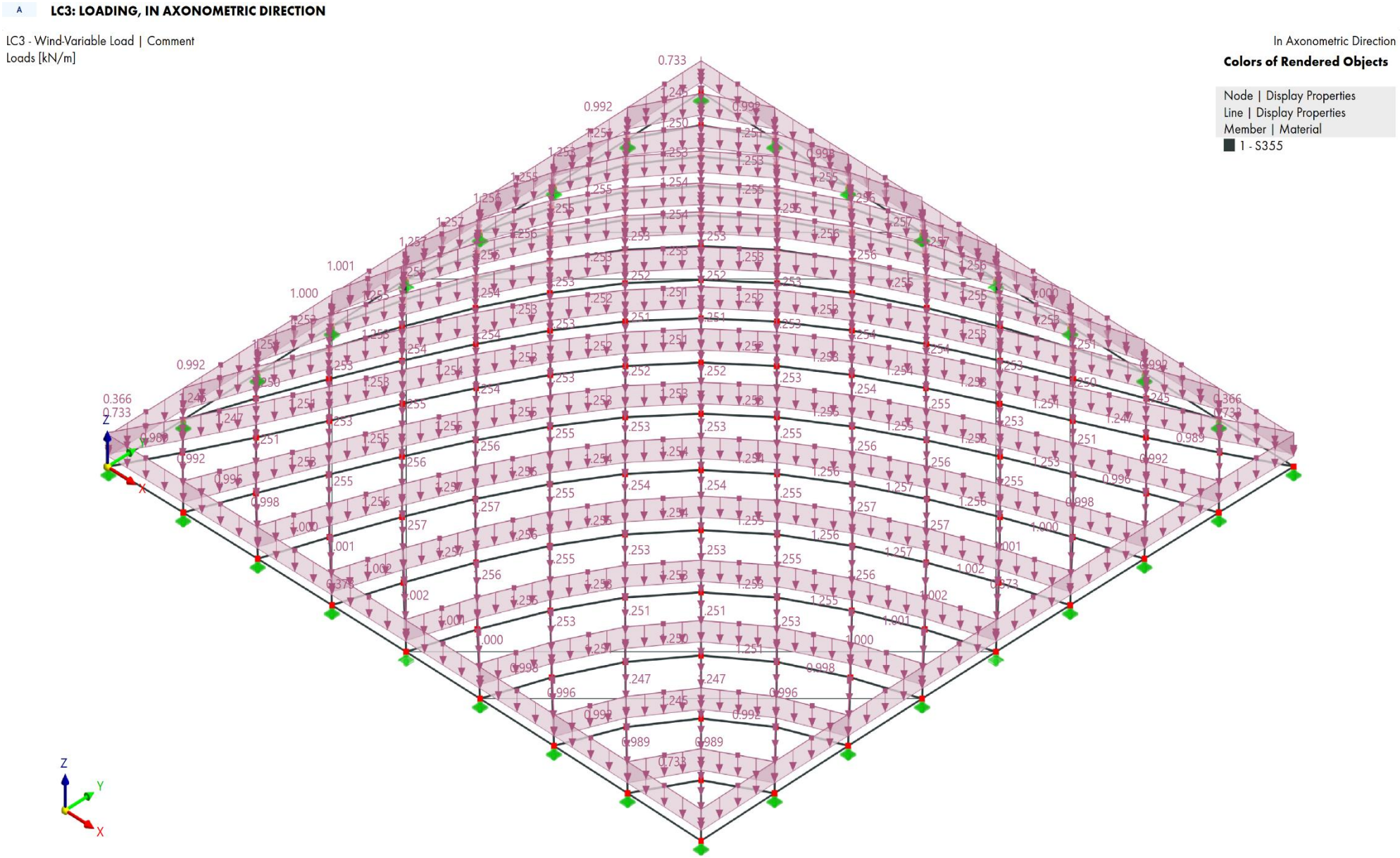


Figure 157: Wind variable load applied along the true length of the elements.

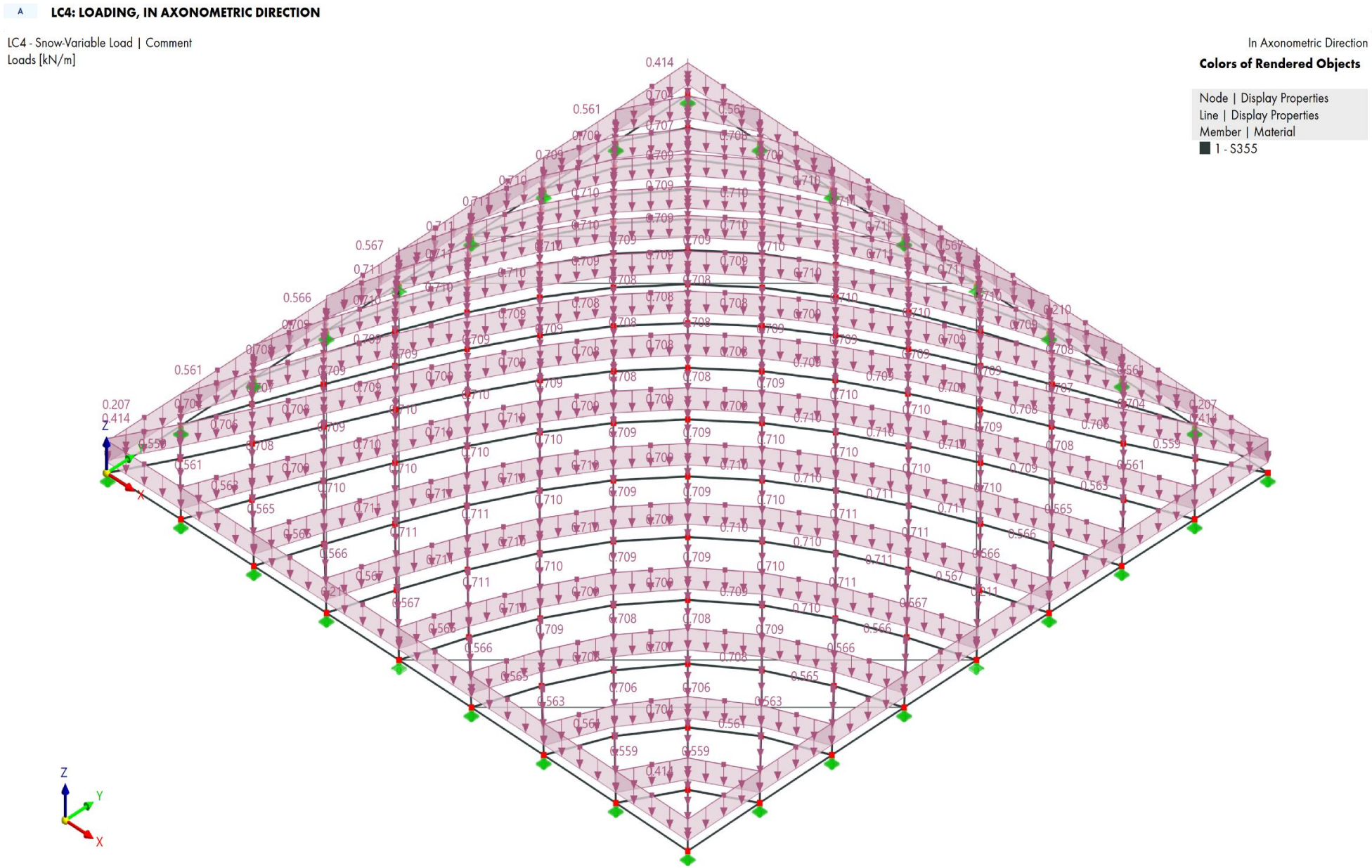


Figure 158: Snow variable load applied along the true length of the elements.

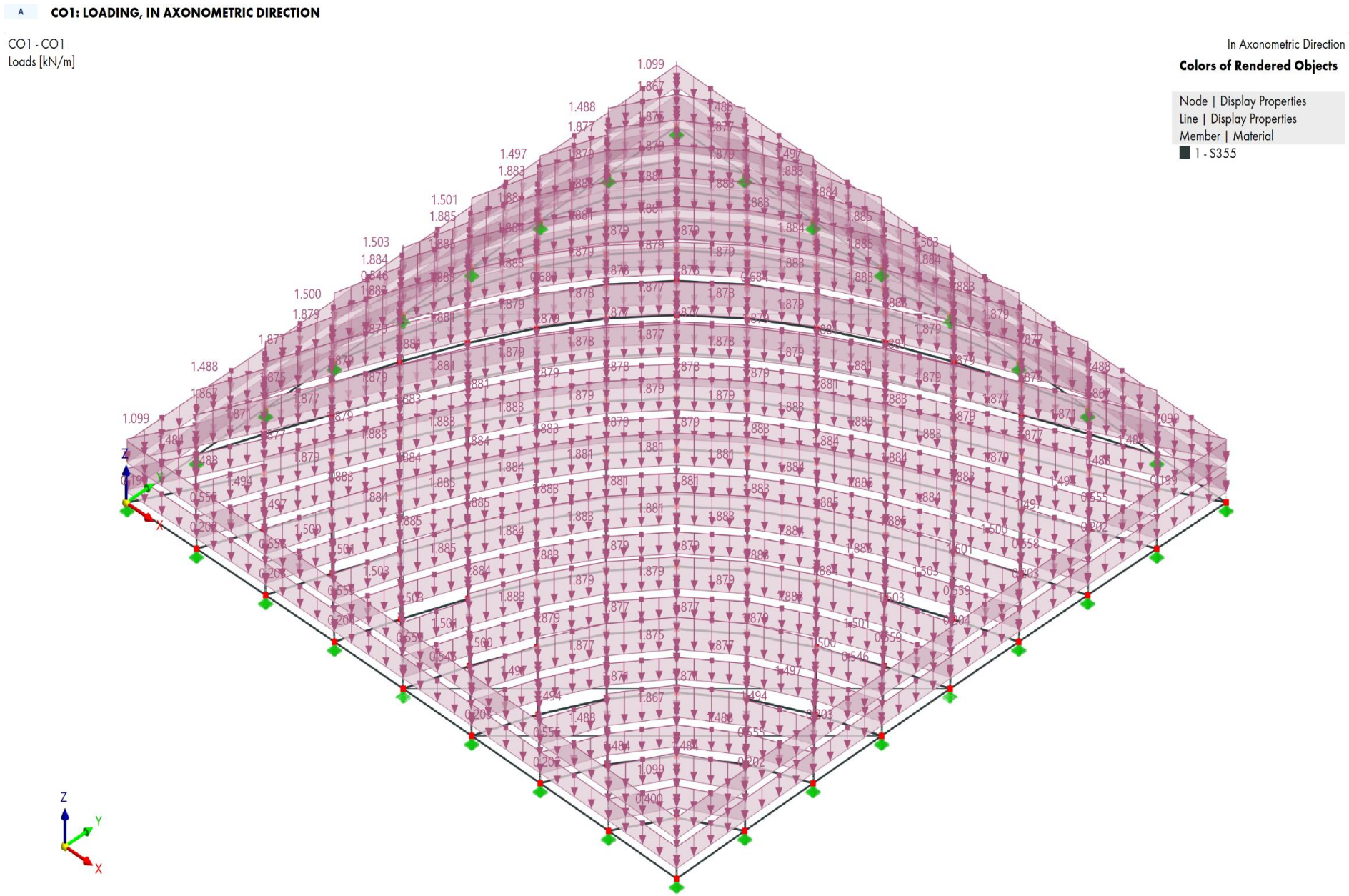


Figure 159: ULS Load Combination: 1.20 * Self-weight + 1.20 * Glass + 1.50 * Wind

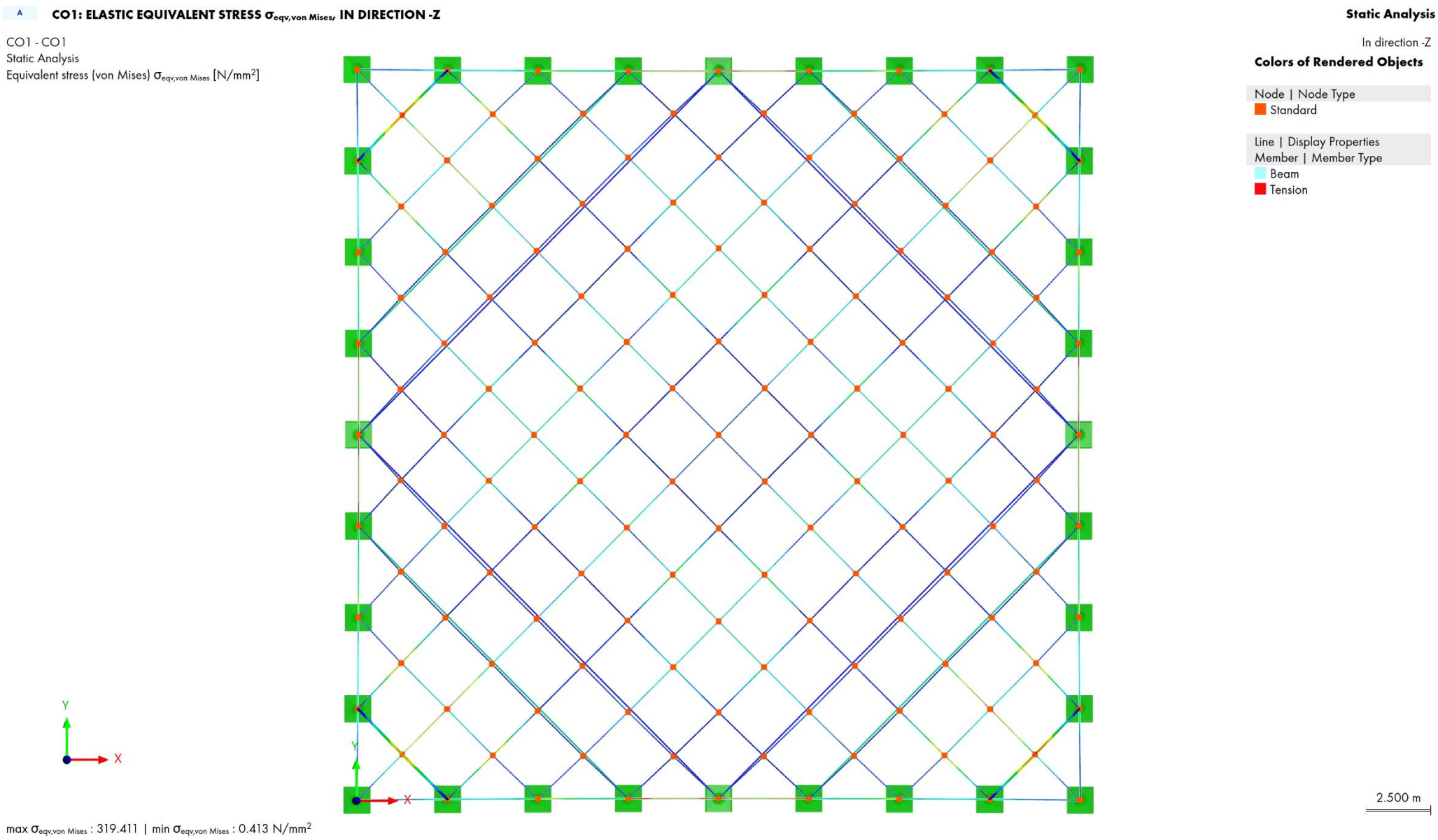


Figure 160: 9×9 gridshell: von Mises equivalent stresses.

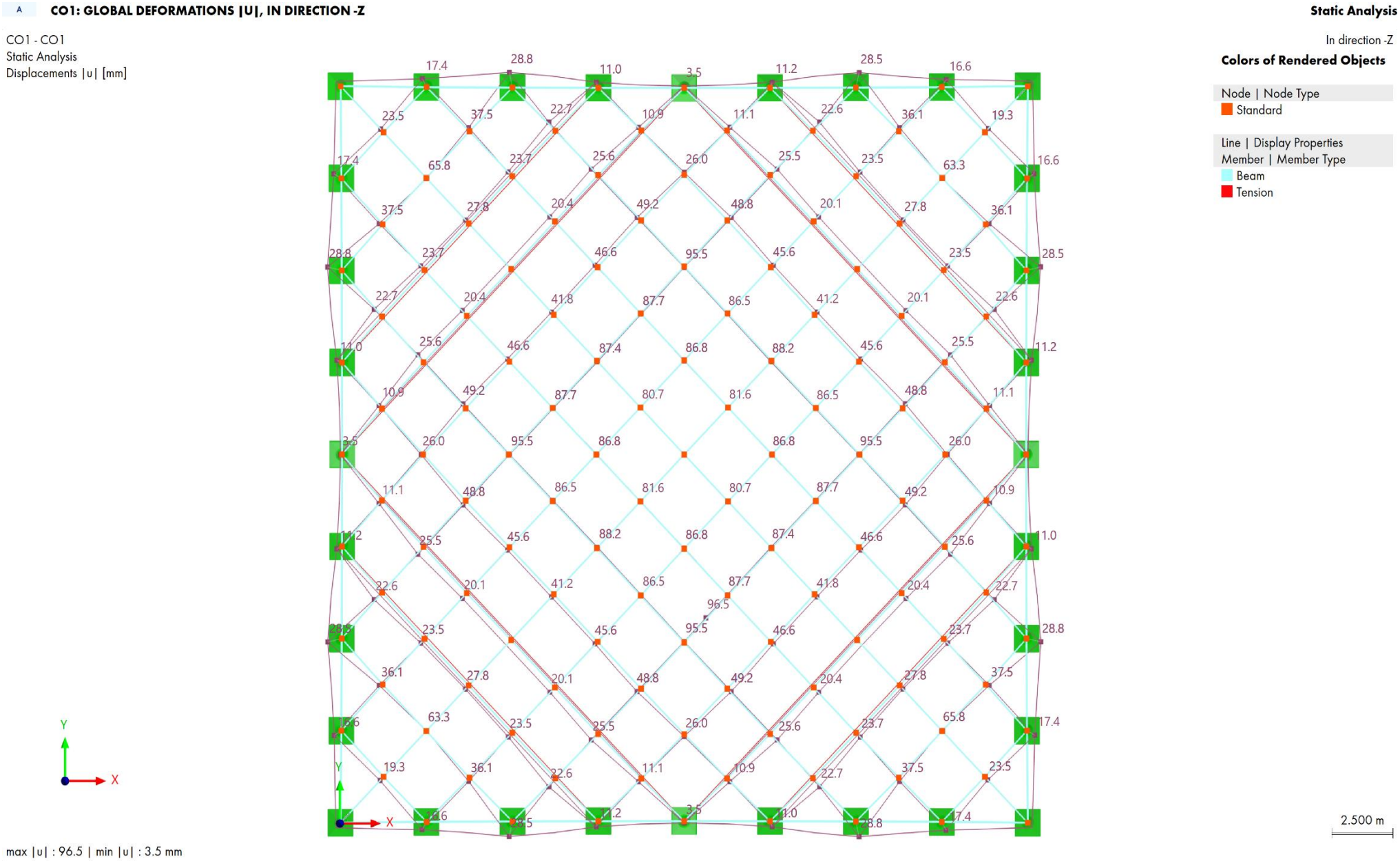


Figure 161: Displacement due to LC1: Top view.

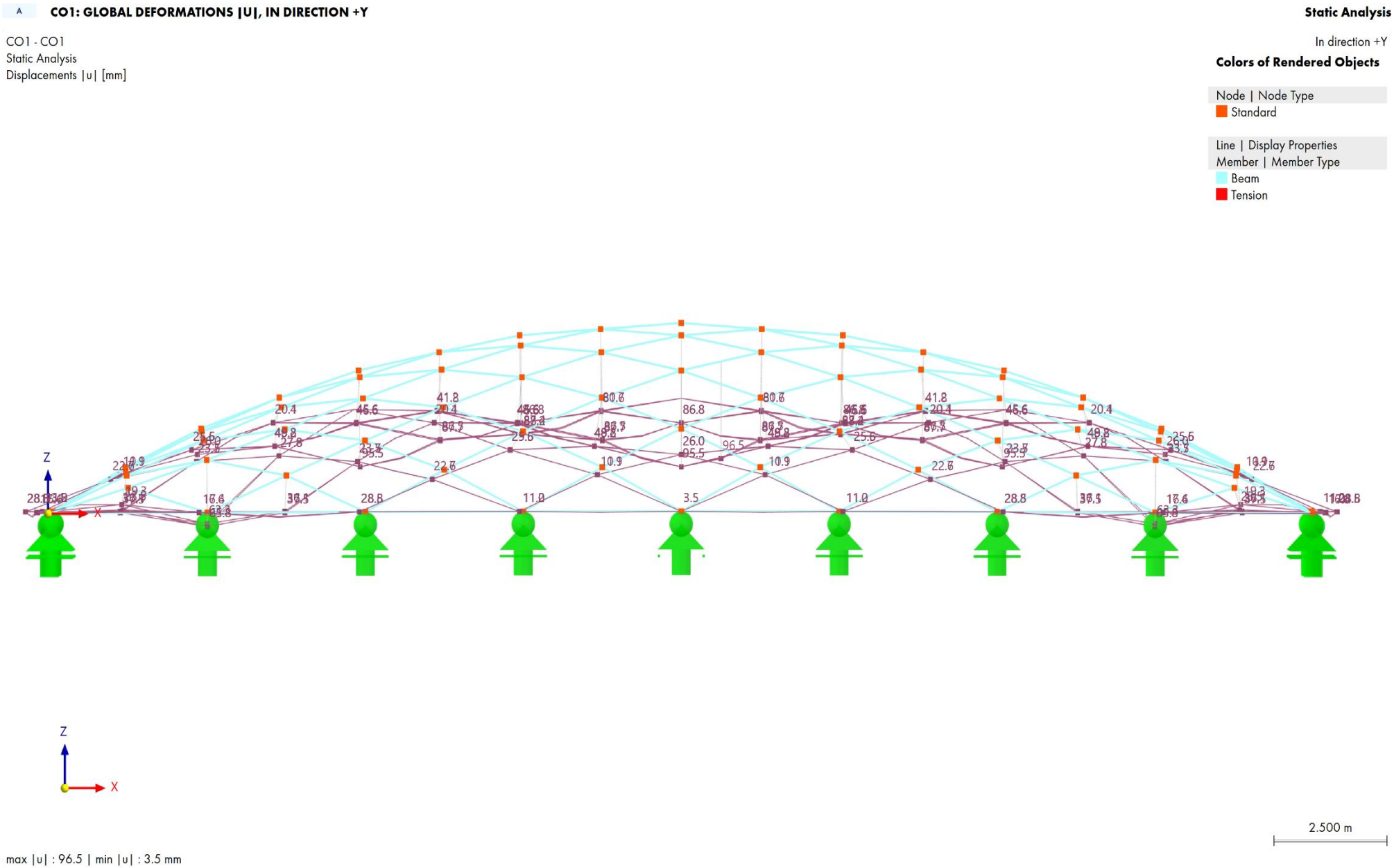


Figure 162: Displacement due to LC1: Side view.

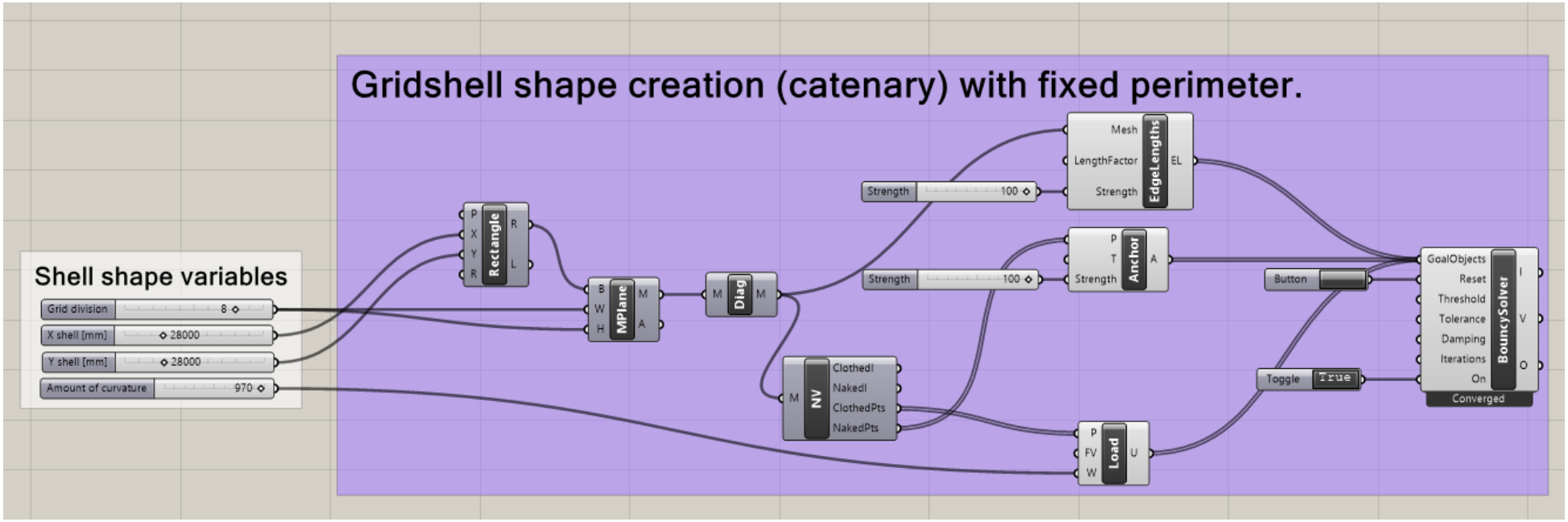


Figure 163: Gridshell 9×9. Grasshopper model using Kangaroo2 plugin for shape generation.

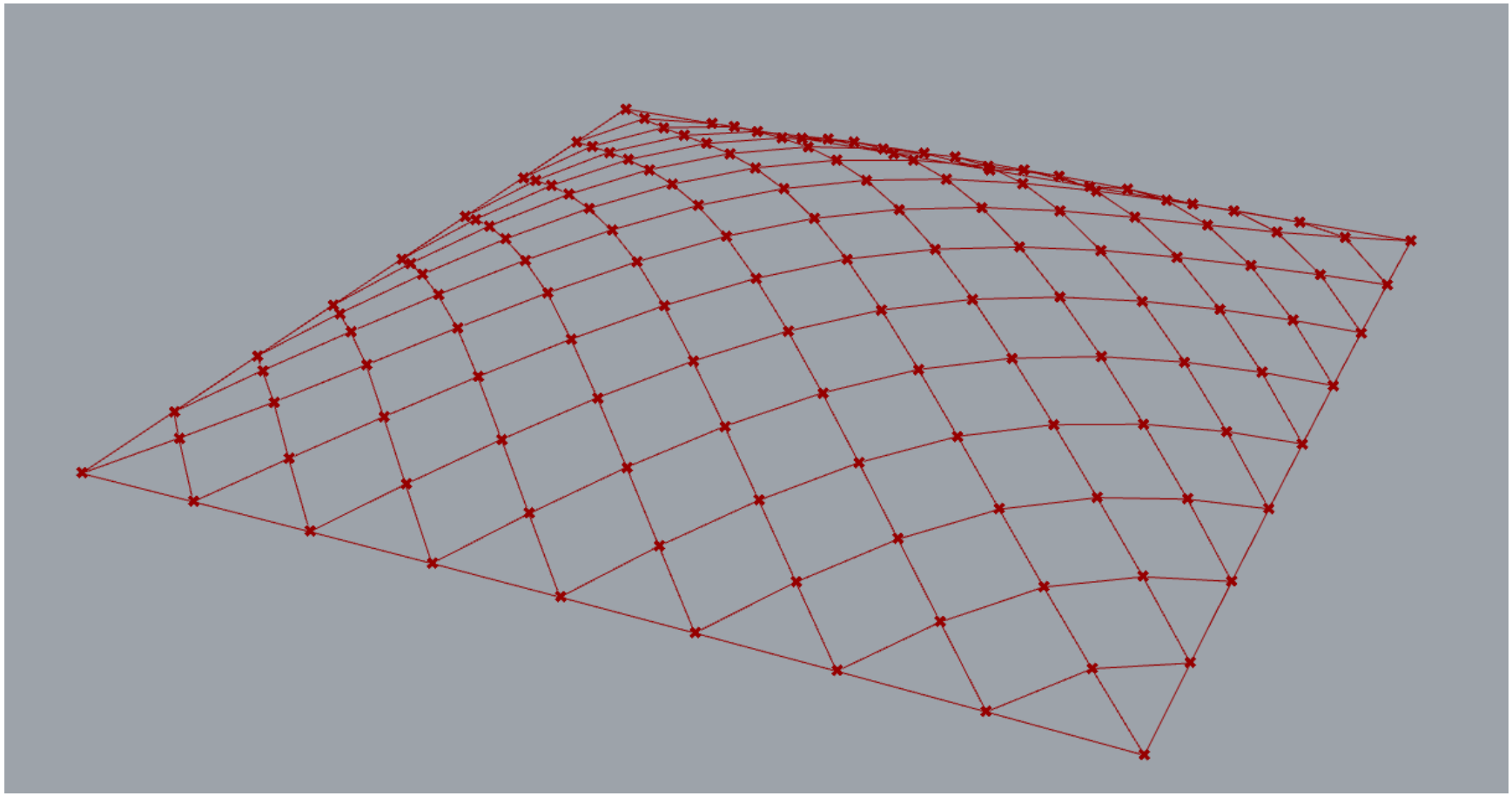


Figure 164: Gridshell 9×9. Generated geometry.

Appendix E: Supplementary Material

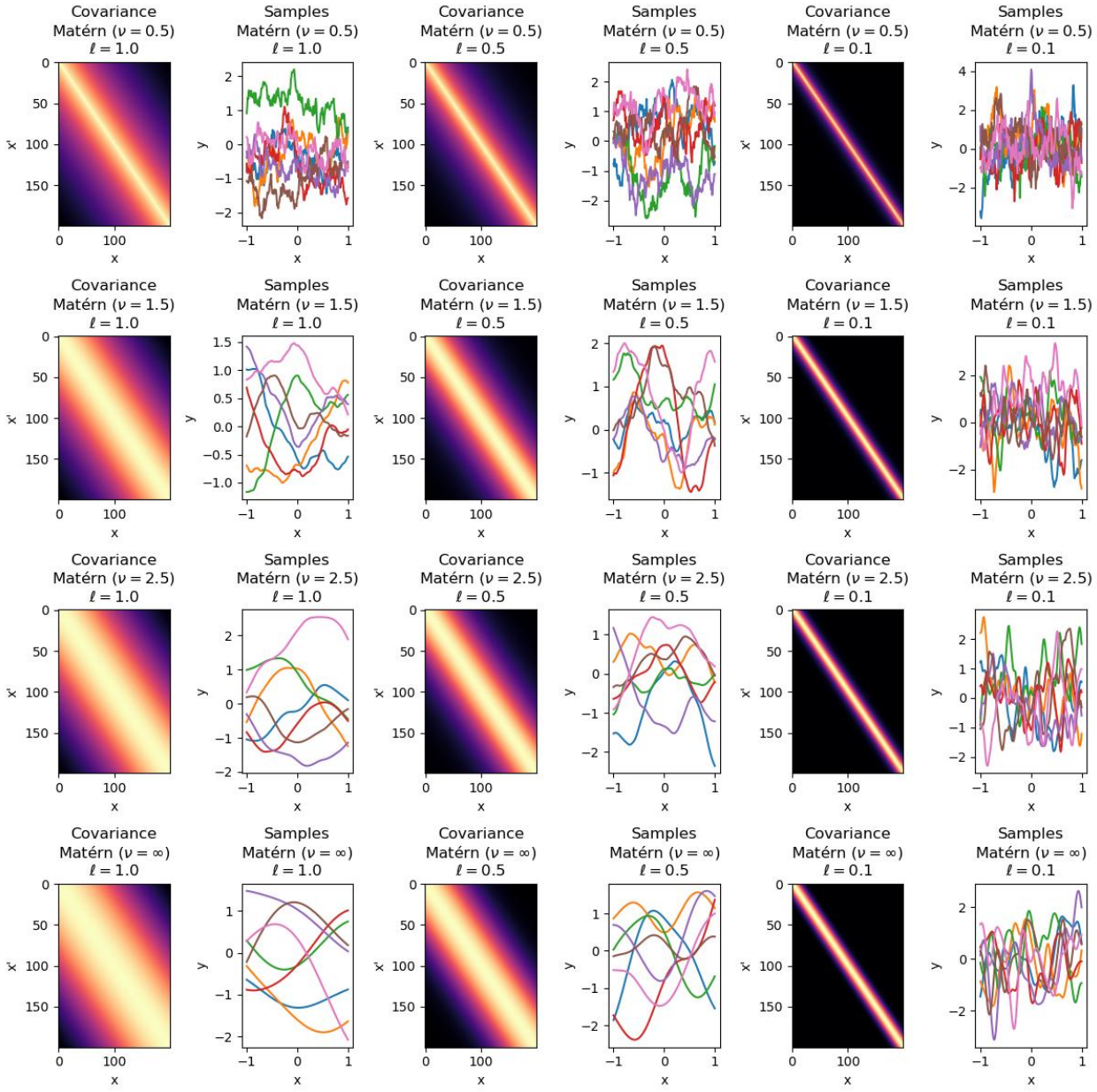


Figure 165: Matérn kernel samples based on different hyperparameter combinations.

Constrained BO algorithm

Start Bayesian optimization

Problem Setup

- Specify the number of design variables such as the geometry (shape) via the node coordinates and/or the areas of individual elements or groups of elements.
- Define upper and lower bounds for each design variable.
- Establish the number of initial design samples.
- Specify the number and type of constraint functions (e.g., stresses, buckling, deflection).

Initial Sampling and Feasibility Check

- Generate an initial set of design samples using a space-filling sampling strategy (e.g., Latin Hypercube Sampling).
- Ensure at least one initially feasible sample is included. If that is not achieved automatically, a manual sample has to be added to serve as a starting point of the optimisation.
- For each sample:
    - Evaluate the objective function and all constraints using the RFEM6 solver.
- Identify and retain the subset of samples that satisfy all constraints or have minimal constraint violation.

Data Normalization

- Fit appropriate normalization or scaling transforms to the design variables, objective values, and each constraint.
- Apply normalization to all sampled data to facilitate efficient modelling.

Surrogate Model Construction

- Initialize surrogate models (e.g., Gaussian Processes) for the objective function and each constraint, with suitable kernel functions and corresponding starting hyperparameters.

Optimization Loop

For each optimization iteration:

1) If necessary (e.g., at regular intervals), update or re-optimize surrogate model hyperparameters.
2) Define an acquisition function that combines expected improvement in the objective with the probability of constraint satisfaction.
3) Maximize the acquisition function over the feasible domain using a global optimization algorithm to select the next candidate design such as the differential evolution algorithm.
4) Denormalize the selected candidate to obtain real design variable values.
5) Evaluate the objective and constraint functions at the new candidate using the RFEM6 solver.
6) Normalize and incorporate the new data into the surrogate models.
7) Update the record of the best feasible solution found so far.
8) Check for improvement; if the objective does not improve sufficiently over a set number of iterations, terminate early.

Result Extraction

- Return the design variables, objective value, and constraint values corresponding to the best feasible solution identified by the algorithm.

End Bayesian optimization

Constrained BO with PCA algorithm

The differences with the previous algorithm are outlined.

Data Normalization

- Fit appropriate normalization or scaling transforms to the design variables, objective values, and each constraint.
- Apply normalization to all sampled data to facilitate efficient modelling.
- Choose a number of principle components based on the explained variance metric.
- Fit the PCA on all the scaled initial samples

Surrogate Model Construction

- Initialize surrogate models (e.g., Gaussian Processes) for the objective function and each constraint, with suitable kernel functions and starting hyperparameters.
- Fit the GPs on the initial samples transformed in the PCA latent space.

Optimization Loop

For each optimization iteration:

1) If necessary (e.g., at regular intervals), update or re-optimize surrogate model hyperparameters.
2) Define an acquisition function that combines expected improvement in the objective with the probability of constraint satisfaction.
3) Maximize the acquisition function over the bounded domain in the PCA latent space using a global optimization algorithm to select the next candidate design such as the differential evolution algorithm.
4) Transform the sampled point from the PCA latent space back to the scaled space.
5) Denormalize the selected candidate to obtain real design variable values.
6) Evaluate the objective and constraint functions at the new candidate using the RFEM6 solver.
7) Normalize and incorporate the new data into the surrogate models.
8) Update the record of the best feasible solution found so far.
9) Check for improvement; if the objective does not improve sufficiently over a set number of iterations, terminate early.