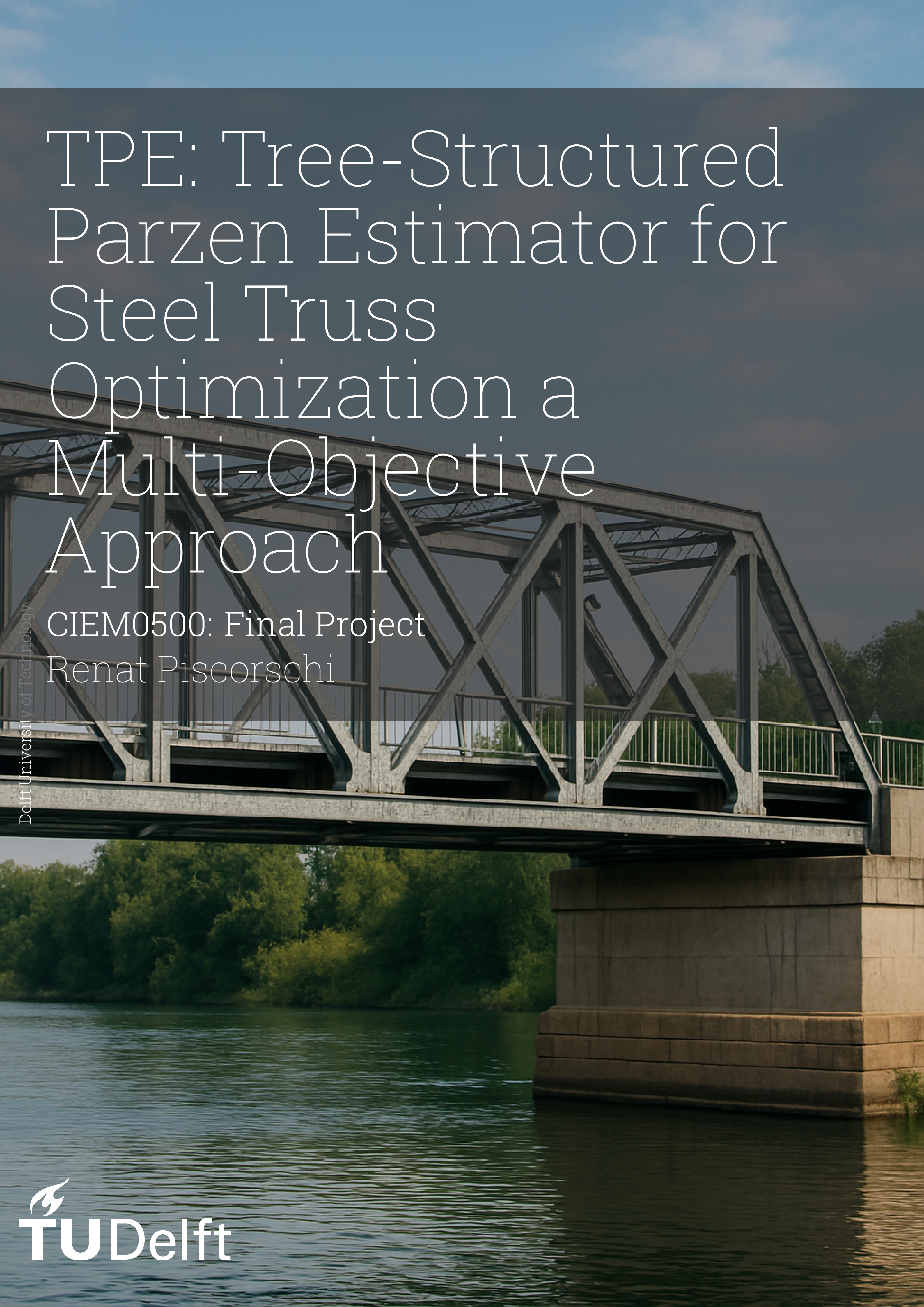


TPE: Tree-Structured Parzen Estimator for Steel Truss Optimization a Multi-Objective Approach

CIEM0500: Final Project

Renat Piscorschi

Delft University of Technology

TPE: Tree-Structured Parzen Estimator for Steel Truss Optimization a Multi-Objective Approach

by

Renat Piscorschi

to obtain the degree of Master of Science

Master of Science in Civil Engineering
Track: Structural Engineering
Specialization: Structural Optimization

at the Delft University of Technology,
Faculty of Faculty Of Civil Engineering and Geosciences,
to be publicly defended on Thursday, 28 May 2025, at 15:30.

Student number:	5853788	
Thesis committee:	Dr. ir. Roel Schipper,	Thesis supervisor, TU Delft
	ir. Chris van der Ploeg,	External supervisor, ABT B.V.
	Dr. ir. Maria Nogal Macho,	External supervisor, TU Delft
	MSc. Tom van Woudenberg,	Thesis supervisor, TU Delft
Project Duration:	September 2024 - May 2025	

Preface

Standing at the threshold of completing this thesis, I am filled with immense gratitude and reflection. What began as an exploration into the unfamiliar domain of generative design has evolved into a deeply transformative journey one marked not only by intellectual growth but also by personal development. This work represents far more than a research project it is a culmination of curiosity, resilience, and the unwavering support of those who have accompanied me along the way.

I extend my deepest thanks to my graduation committee for their insightful feedback, academic guidance, and critical engagement throughout this process. Their expertise has been instrumental in shaping the direction and depth of this research. In particular, I am sincerely grateful to Dr. ir. Roel Schipper and Tom van Woudenberg. Dr. Roel, as chair of my committee, offered both technical insight and moral encouragement, your pragmatic advice and steady presence made a lasting impact. Tom, your continuous involvement and thoughtful feedback helped keep this project focused and grounded, especially during the key stages of the paper submission. I would also like to thank Dr. ir. Maria Nogal Macho for her participation in my defense and her kind support during the final stages of this journey.

I owe special thanks to ir. Chris van der Ploeg, whose role extended far beyond that of a supervisor. From the very beginning, Chris was a source of steady motivation and honest critique, encouraging me to push boundaries while always providing clarity when I needed it most. His commitment and attention to detail helped me maintain a high standard throughout, and for that, I am truly thankful.

To my family and friends, thank you for standing by me. Your encouragement, patience, and belief in me helped me persevere through the most demanding moments. This thesis is, in many ways, as much a product of your support as it is of my own effort.

Completing this work has often felt like running a marathon, at times exhilarating, at other times grueling. But through the highs and challenges, I have emerged with a greater sense of purpose, discipline, and appreciation for the research process. It is my hope that the outcomes of this work contribute meaningfully to the field and inspire further inquiry at the intersection of design, optimization, and engineering.

Renat Piscorschi
Delft, May 2025

Summary

The design of 2D rectangular steel trusses demands a critical balance between structural performance and constructability, a core challenge in civil engineering. Using distinct cross-sectional profiles minimizes material use but elevates structural complexity, whereas standardized profiles facilitate construction simplicity at the cost of efficiency. This thesis develops a multi-objective optimization framework to navigate these trade-offs, targeting four essential objectives: mass minimization to reduce material requirements, connection degree to simplify joint configurations, symmetry to enhance aesthetics and standardization, and beam continuity to streamline assembly processes. By controlling the number of unique HEA profiles, the study delivers tailored solutions for preliminary structural design, aligning with engineering priorities and stakeholder preferences to optimize truss performance and practicality.

A computational framework employs the Tree-structured Parzen Estimator (TPE), a sample-efficient Bayesian optimization method, to efficiently explore the complex, discrete design space of truss configurations. TPE performance is rigorously validated against exhaustive search (EXS) to ensure accuracy in identifying optimal designs. Stakeholder-defined weights, implemented through weighted scalarization, enable customized trade-off analyses, though without direct stakeholder engagement. This approach supports the exploration of diverse configurations, effectively balancing performance and standardization while addressing the computational demands of large search spaces, thus providing a robust tool for 2D truss optimization.

The findings indicate that intermediate profile grouping often produces designs that balance structural performance and constructability. The multi-parallel plot, a dynamic visualization tool, potentially empowers stakeholders, including engineers and project managers to transparently explore trade-offs, pending practical validation. Despite limitations, such as untuned TPE hyperparameters and a focus on 2D trusses, this promising framework enhances transparency and adaptability in preliminary structural design. By integrating efficient optimization with intuitive visualization, the study establishes a foundation for future advancements in steel truss optimization, offering a versatile methodology with potential to inform broader structural engineering applications.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
BCN	Beam Continuity
EXS	Exhaustive Search Algorithm
FEA	Finite Element Analysis
HEA	Flanged steel section
KDE	Kernel Density Estimators
M	Mass
SYM	Symmetry
TPE	Tree-structured Parzen Estimator Algorithm
UC	Unity Check

Symbols

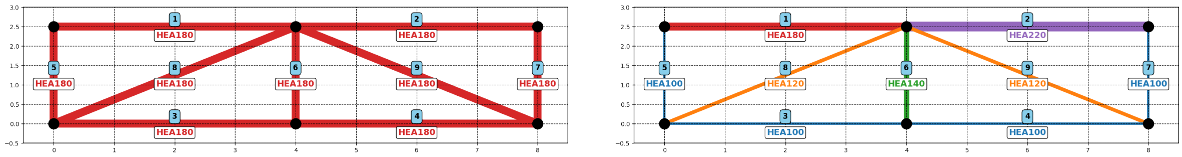
Symbol	Definition
A_i	Cross-sectional area of element i
$best\_config_n$	Best configuration for batch n
$best\_value_n$	Best objective value for batch n
$d(v_j)$	Degree of node v_j
E	Set of edges (elements) in the truss graph
e_i, e_j	Truss elements in a mirrored pair
f	Function for Structural Complexity objectives
f_{all}	Function for all objectives (*M*, *CN*, *SYM*, *BCN*)
f_{BCN}	Beam Continuity objective function
f_{CN}	Connection Degree objective function
$f_{\textit{M}}$	Mass objective function
f_{SYM}	Symmetry objective function
$G = (V, E)$	Undirected graph representing the truss
k	Number of elements
L_i	Length of element i
m	Number of nodes in the truss
n	Number of unique HEA profile types
N_{trials}	Number of trials
P	Set of mirrored element pairs
$p(x \mid D^{(g)})$	Probability density in the good-performing group
$p(x \mid D^{(l)})$	Probability density in the low-performing group
$\mathcal{P}$	Profile set (e.g. HEA100, HEA120)
R	Number of horizontal rows in the truss
ρ	Material density of S355 structural steel kg/m³
s	Configuration in the search space
$\mathcal{S}$	Search space of all configurations
$\mathcal{S}_n$	Subset of configurations with exactly n profiles
V	Set of nodes (joints) in the truss graph
w_{BCN}	Weight for beam continuity objective
w_{CN}	Weight for connection degree objective

Symbol	Definition
w_M	Weight for mass objective
w_{SYM}	Weight for symmetry objective
$\mathcal{W}$	Weight distributions
$\delta(e_i, e_j)$	Function for profile mismatch in mirrored pairs
γ_r	Number of uninterrupted profile matches found along row r

1
Introduction

Steel trusses are critical in civil engineering for their load-bearing efficiency in structures such as bridges and buildings, nevertheless their design demands a balance between constructability and performance. The construction of steel truss buildings poses a persistent challenge for engineers, requiring the minimization of design complexity while ensuring structural performance.

Following the principle of commonality [49], assigning unique cross-sectional profiles to each truss member minimizes material use but increases fabrication and construction costs. This principle asserts that fewer distinct profiles reduce costs through bulk discounts, simplified fabrication, and fewer unique connections, although overdesign may increase material use [49]. In contrast, grouping elements to share profiles improves buildability by reducing profile diversity [5]. Figure 1.1 illustrates this trade-off, comparing a standardized truss where every members sharing a standard HEA profile with a unique configuration where almost every member has different profiles for a structurally determinate 2D rectangular steel truss.



(a) Standardization - Example design: every member has a standard profile.

(b) Uniqueness - Example design: almost every member has a different profile.

Figure 1.1: Comparison between examples of Standardized VS Unique Truss Configuration

Traditionally, grouping of truss members is done manually based on expert judgment, where engineers select a limited number of profile types to reduce diversity. However, this manual approach often leaves many alternatives unevaluated due to time constraints. Recent advancements in algorithmic design and computational tools have enabled structural engineers to explore complex trade-offs more rigorously, facilitating the identification of optimal solutions for individual truss members. In this line, research has demonstrated how automated search procedures and grouping strategies can reduce structural diversity while retaining standardization performance [27] [48]. However, these approaches may depend on thorough analyses or heuristics, which might not be able to scale effectively or apply generally across design challenges, because thorough analyses can be computationally intensive for large-scale structures, while heuristics may not generalize well across diverse design scenarios.

This thesis explores the function of grouping in truss optimization employing a multi-objective method in order to address the existing gap [38]. The study investigates how various grouping layouts, each characterized by a set number of allowed profile types (n), affect structural behavior over numerous performance objectives. It also takes into account how the solution space changes when the priority of these goals varies, which resembles the case when structural complexity objectives such as symmetry or connection degree comes first rather than structural mass [6]. To tackle these challenges, this thesis focuses on balancing conflicting goals of minimizing structural complexity and maximizing standardization while seeking efficiency through customized truss configurations, with the specific optimization methodology to be detailed in later chapters.

To support this goal, the research is highly relevant to stakeholders, including structural engineers, constructors, and clients, whose influence critically shapes construction project outcomes by balancing design efficiency with practical constraints such as cost, constructability, and safety [33]. It provides a systematic framework for truss design optimization, empowering stakeholders to make informed decisions that enhance structural performance and economic viability. By leveraging advanced computational tools for multi-objective optimization, this work enables stakeholders to collaboratively explore diverse design alternatives, addressing their unique priorities and fostering effective decision-making in complex construction environments.

1.1. Research questions

This thesis primarily investigates the optimization of grouping strategies using a 9-element, structurally determinate 2D rectangular steel truss as the benchmark model, with results for three additional truss topologies (Typology 1,2,3) detailed in Appendix A. The analysis optimizes grouping strategies balancing conflicting objectives: mass, symmetry, connection degree, and beam continuity via different optimization algorithms. The research is guided by the following main research question:

To what extent can grouping strategies in steel truss structures be optimized to balance multiple, often conflicting objectives, there by enabling stakeholders to make informed and transparent design decisions?

This question drives the development of a multi-objective optimization framework that integrates stakeholder preferences to balance performance objectives. The focus is on creating transparent solutions for the steel truss, enabling stakeholders to evaluate trade-offs between constructability and efficiency. To address this main research question, the following sub-questions are investigated:

RQ1: How do grouping configurations with a fixed number of allowed HEA profiles (n) affect structural mass, symmetry, connection complexity, and beam continuity?

This sub-question explores how varying the number of allowed HEA profiles (n) impacts key performance objectives in the operated steel truss. By analyzing grouping configurations, the study aims to quantify their effects on structural metrics, balancing standardization with reduced complexity, as inspired by the principle of commonality [49].

RQ2: What trade-offs emerge between standardization (e.g. symmetry) and uniqueness (e.g. mass) when different objectives are prioritized?

This sub-question investigates how prioritizing objectives, for instance symmetry over mass, shapes design trade-offs in the operated steel truss. The goal is to provide insights into balancing standardization for constructability, while informing stakeholder decision-making.

RQ3: In which core aspects of steel truss optimization does the Tree-structured Parzen Estimator (TPE) offer advantages over simple heuristics, metaheuristics like NSGA-II, and exhaustive search for identifying high-quality groupings in single and multi-objective scenarios?

This sub-question examines the core aspects where the Tree-structured Parzen Estimator (TPE) offers advantages over simple heuristics like RandomSampler, metaheuristics like NSGA-II, and Exhaustive Search (EXS) for operated steel truss optimization. The focus is on assessing TPE computational efficiency, scalability, and suitability for discrete single and multi-objective optimization, compared to the inefficiencies of simple heuristics, the limitations of other metaheuristics, and the reliability but computational infeasibility of exhaustive search [35].

RQ4: In what ways can stakeholder-defined weights, reflecting their objectives or design preferences, influence the resulting optimal truss layouts?

This sub-question investigates the influence of stakeholder-defined weights on the optimization of designs for the operated steel truss in various scenarios. It examines whether stakeholders own specific design preferences or necessitate adaptable solutions, hence strengthening the framework's adaptability.

RQ5: In what ways can visual or computational tools enhance the interpretability of optimized truss designs and assist in navigating complex trade-off spaces?

This sub-question investigates the effectiveness of parallel coordinates plots for visualizing trade-offs in optimized designs of the operated steel truss. The aim is to develop intuitive tools that facilitate stakeholder decision-making, prioritizing parallel coordinates plots for their clarity over alternatives like spider diagrams.

1.2. Contributions and objectives

This thesis addresses structural optimization by introducing a novel framework for grouping-based design of a structurally determined 2D rectangular steel truss, addressing trade-offs between standardization and performance, the main contributions are:

- A multi-objective optimization framework leveraging TPE and Exhaustive Search to target mass, symmetry, connection degree, and beam continuity, ensuring efficient and constructable designs for rectangular steel truss configurations.
- A grouping-based modeling approach that controls the number of unique HEA profiles, reducing fabrication complexity while maintaining structural performance.
- A comparative analysis of Exhaustive Search and TPE benchmark trusses designs, demonstrating their effectiveness in single and multi-objective scenarios,
- A stakeholder-driven weighting system that adapts truss designs to diverse preferences, navigating complex trade-off spaces.

1.3. Thesis organization

This thesis is structured as follows: Chapter 2 (Preliminaries) provides foundational knowledge on truss modeling, addressing mass, connectivity degree, symmetry, and beam continuity, thereby delineating the essential concepts relevant to classification techniques and vital structural objectives. Chapter 3 (Literature Review), presents a critical analysis of prior research on structural optimization by element grouping. It examines relevant algorithms, multi-objective optimization methodologies, and modern developments in balancing structural complexity with performance. Chapter 4 (Methodology) explains the logic of truss generation, parameterization, objective metric definitions, the role of stakeholder-defined weights and the implementation for the Exhaustive Search (EXS) and Tree-structured Parzen Estimator (TPE). Chapter 5 (Experimental Setup and Validation), outlines the experimental design, detailing the truss benchmark model, simulation environment, and optimization configurations. The framework facilitates the comparison of various optimization techniques in both single and multi-objective contexts, incorporating validation through exhaustive search. Chapter 6 (Results and Discussion) reports findings on grouping impacts and optimization performance, supported by visualizations, and addresses the research questions listed in Chapter 1. This assesses the impact of grouping level and stakeholder preference vectors on structural metrics. Chapter 7 (Limitations and Future Work) delineates framework's limitations and proposes future research, addressing implementation challenges, and prospects for expanding the work to more extensive design settings. Chapter 8 (Conclusion) is summarizing the principal contributions and outcomes of the research. It examines the practical ramifications of grouping-based truss optimization and the function of multi-objective techniques in facilitating informed stakeholder-driven truss design choices.

2
Preliminaries

2.1. Introduction to steel truss systems

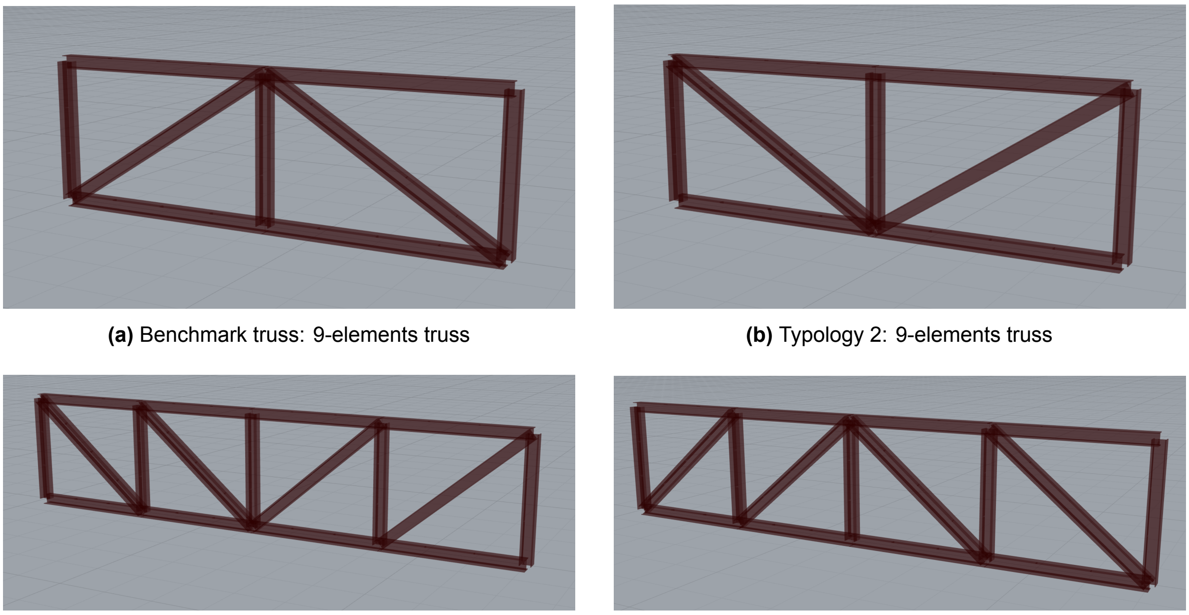
Steel truss systems are fundamental structural components in civil engineering, known for their high strength-to-weight ratio, ease of prefabrication, and material efficiency [12]. These systems consist of interconnected triangles, typically made of steel, which distribute loads efficiently across the structure. Widely utilized in bridges, buildings, and load-bearing structures, steel trusses are regulated by standards such as Eurocode [21]. Thus, their geometric precision and mechanical consistency could make them particularly suitable for analyzing and testing optimization and parametric modeling, especially in computational design processes.

This study focuses on rectangular steel trusses, chosen for their significance in wide-reange applications and adaptability to computational optimization techniques. The rectangular configuration provides balanced load distribution and usually is representative of modular pedestrian and light vehicle bridges. The relevance of steel truss systems lies in their ability to span large distances with minimal material, perhaps making them cost-effective and environmentally friendly [30]. Their modular nature facilitates standardization and industrialization in construction, aligning with modern engineering practices emphasizing sustainability and efficiency [6] [19].

2.1.1. Characteristics of rectangular steel trusses

Rectangular steel trusses are widely used in applications such as construction projects due to their simplicity and efficient load distribution. Their consistent geometry makes them ideal for benchmark comparisons in optimization studies. This study employs a 9-element reference truss as the benchmark problem, consisting of vertical, diagonal, and horizontal members, as illustrated in Subfigure 2.1a. Each member is made of S355 structural steel, a widely used grade in Europe due to its balance of strength and ductility, as specified in Eurocode [21].

To isolate axial behavior and simplify modeling, the trusses are idealized as planar and braced out-of-plane, with all joints considered pinned, forming a statically determinate structure. Under these assumptions, the structural response is driven solely by axial forces (tension or compression) without internal moments [4]. A uniformly distributed load applied over the top section of the truss is assumed to act directly at the nodes for computational simplicity, ensuring only axial forces are considered [23]. Additionally, three truss topologies (Topology 1,2,3) are modeled to verify optimization findings under different configurations and loading conditions, as depicted in Figure G.5. Neverhteless, the 9-element benchmark truss remains the primary framework for all visualizations and optimization results in this thesis.



(a) Benchmark truss: 9-elements truss

(b) Typology 2: 9-elements truss

(c) Typology 1: 17-elements truss

(d) Typology 3: 17-elements truss

Figure 2.1: Overview of the four truss typologies opeated in this study, illustrating variations in the number of elements (k).

2.1.2. Selection of HEA profiles for optimization

Throughout this study the HEA profiles are selected due to their symmetry, moderate flange width, and standardized dimensions, which are particularly advantageous for truss grouping methodologies. These attributes may facilitate easier connection and standardization in truss design, reducing the complexity and cost associated with using a variety of profile types [38]. The symmetrical shape ensures uniform load distribution, while moderate flange widths allow for efficient bolted or welded connections, and standardized dimensions align with common manufacturing processes [19]. HEA profiles are widely used in Europe and comply with Eurocode [21] standards, ensuring their suitability for structural optimization in steel trusses.

For the primary testing in this thesis, the 9-element benchmark truss employs a set of five distinct HEA sections: HEA100, HEA120, HEA140, HEA180, and HEA220, while other examples within the study engage different profiles. This profile range is chosen for the benchmark to maintain compatibility with design software and production processes while ensuring computational manageability. While larger HEA profiles (e.g., HEA300 and above) are typically used for longer spans or heavier loads in practical applications, such as large bridges or industrial structures, the selection of smaller profiles for the benchmark truss is intentional. The 9-element rectangular truss benchmark is designed to allow comprehensive exploration of optimization techniques without the computational burden of larger structures. This choice enables the focus to remain on the optimization methodology and the trade-offs between different objectives, rather than on absolute structural dimensions.

In truss design, where axial forces dominate, the axial stiffness and cross-sectional area of the profiles are critical performance metrics [29]. These properties directly influence the structural mass and the Unity Check (UC), ensuring that the design is both efficient and safe [2]. The use of HEA profiles allows for a balance between material efficiency and structural performance, making them ideal for this optimization study. While prior research has explored the dynamic properties of trusses with HEA profiles, such as modal behavior [15], this thesis narrows its scope by excluding these aspects, instead concentrating on objectives potentially most commonly prioritized by typical stakeholder groups, with a focus on structural efficiency.

2.2. Element grouping in structural design

Element grouping is a critical strategy in optimizing steel trusses, balancing structural simplicity with performance. Grouping involves assigning identical cross-sections to sets of truss elements, reducing the number of unique profiles used. This approach lowers fabrication and assembly costs, enhances constructability, and improves design efficiency [49].

In practice, structural engineers often manually group truss members based on their functional roles or load conditions. For instance, top chord members, typically under compression, may share one cross-section, while bottom chord members, usually in tension, use another. Web members, such as verticals and diagonals, may be grouped separately, with diagonals sometimes distinguished by tension or compression states. This manual grouping might have a substantial potential to simplify the design and fabrication while maintaining reasonable structural efficiency [49]. Nonetheless, identifying an optimal balance between full standardization and complete member uniqueness poses a non-trivial challenge, particularly in high-dimensional design spaces. This trade-off underscores the need for a formalized optimization framework capable of systematically navigating the design space and evaluating competing objectives, as addressed in this thesis.

2.2.1. Standardization and uniqueness

As followed from Figure 1.1 standardized truss configurations assign the same cross-section to all truss elements, minimizing component variety, conversely, unique truss configurations tailor a specific cross-section to each element, optimizing material use based on individual load and boundary conditions.

Standardized truss configurations often lead to overdesign, where elements are sized for the most critical member's forces, resulting in larger sections than necessary for others. This ensures the truss can handle worst-case loads but may waste materials. In contrast, unique truss configurations allow precise sizing, achieving utilization ratios (UC) near optimal levels (e.g., UC ≈ 0.85 – 1.0). However, they increase manufacturing and assembly complexity [23]. Moreover, unique truss configurations with varied member sections might complicate connection design, raising engineering and fabrication costs due to custom detailing [45]. In contrast standardized truss configurations may enable standardized connections, which are more cost-effective to design and produce [42].

2.2.2. Trade-offs in grouped member design

Standardized truss configurations aim to balance performance, manufacturability, and simplicity. Studies indicate that grouping members with similar load demands can maintain efficiency [49]. For example, in trusses with symmetric loading, members like bottom chords or diagonals may experience comparable axial forces, allowing effective grouping without significant efficiency losses. However, factors like truss shape, boundary conditions, and load asymmetry can limit grouping effectiveness. In trusses with irregular geometries or non-uniform loading, members that appear similar may face different forces, making efficient grouping challenging without compromising performance [23].

Research shows that structural mass, symmetry, and connection complexity vary with group numbers, with optimal designs often at intermediate levels, balancing the extremes of fully standardized and fully unique configurations [49]. Later sections will treat group count as a decision variable, namely the (n) the allowed number of HEA profiles, influenced by stakeholder preferences (e.g.,constructability, mass) and optimization constraints such as limits on unique profiles and prioritization of specific design requirements.

3

Literature review

3.1. Grouping strategies for steel structures

Grouping strategies in steel structures aim to reduce the number of unique cross-sectional profiles, enhancing constructability while maintaining structural performance. These strategies assign identical profiles to sets of elements, minimizing fabrication complexity and aligning with practical design constraints [48]. Recent literature highlights advances in automated grouping methods, driven by computational tools and optimization algorithms, which offer efficient alternatives to manual grouping [16].

3.1.1. Review of grouping literature

Early grouping methods relied on manual assignment based on engineering intuition, often grouping elements by load type (e.g., tension vs. compression members) or structural role (e.g., chords vs. diagonals) [46]. Such approaches, while practical, frequently led to overdesign, increasing material use. Advances in computational design have introduced automated grouping techniques, such as combinatorial search algorithms that optimize profile assignments while limiting diversity [49]. For instance, [48] proposed a fully stressed design approach, iteratively grouping elements to minimize weight and profile variety, achieving results comparable to complex optimization methods with fewer computational resources.

Recent studies have explored penalty-based grouping to balance structural efficiency and constructability. Following [16] damped exponential penalties to limit profile transitions are introduced, demonstrating improved designs for truss beams. Similarly, [40] developed multi-objective grouping for 3D steel frames, incorporating column symmetry and bracing configurations to reduce fabrication costs. These advances highlight the shift toward data-driven, automated grouping, leveraging tools like Grasshopper and Karamba3D to evaluate thousands of different structural configurations [10].

3.1.2. Practical constraints in grouping

Grouping strategies must account for practical constraints, including manufacturing limitations, cost implications, and stakeholder preferences [6]. Fabrication facilities often impose restrictions on the number of unique profiles due to inventory and machining constraints, necessitating fewer profile types to reduce costs [23]. Complex joints, resulting from high profile diversity, increase welding or bolting expenses and assembly time, particularly for prefabricated structures [16]. Stakeholder requirements, such as aesthetic symmetry or modular transportability, further constrain grouping, as seen in designs prioritizing uniform chords for visual harmony [40]. For example, [31] emphasized the importance of aligning grouping with industrialized construction workflows to enhance modularity and reduce on-site adjustments. This discrete nature of profile selection, combined with the need to balance multiple objectives like mass, symmetry, and connection simplicity, poses significant computational challenges. Efficient optimization methods are required to navigate large, discrete search spaces while incorporating stakeholder-driven weights to reflect preferences for constructability or aesthetics [32]. Bayesian optimization, particularly the Tree-structured Parzen Estimator (TPE), is well-suited for such tasks due to its ability to model probabilistic relationships in discrete spaces and adapt to weighted multi-objective functions, offering a promising approach for truss optimization [37].

3.2. Optimization methods for discrete structural problems

Structural optimization in civil engineering involves selecting optimal designs from a set of alternatives to meet criteria such as minimizing material use or enhancing constructability [12]. Discrete problems, such as assigning cross-sectional HEA profiles to steel truss elements, require methods that balance multiple objectives, including potential balance of parameters like mass, symmetry, connection degree, and beam continuity [48]. This section reviews five optimization methods implemented in Optuna [35], a hyperparameter optimization framework: the fundamental approach of exhaustive search (EXS), the heuristic method of random sampling (Random-Sampler), and three metaheuristic approaches Non-dominated Sorting Genetic Algorithm II (NSGA-II), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), and Tree-structured Parzen Estimator (TPE). Heuristics employ simple, problem-specific rules to find satisfactory solutions, while metaheuristics use iterative, guided strategies to explore complex search spaces [25]. Among these, TPE is a Bayesian metaheuristic that models the probability distribution of high-performing configurations, offering efficiency in discrete spaces [7]. These methods are evaluated for the 9-element truss benchmark in this study, emphasizing TPE algorithm effectiveness for weighted scalarization of objectives in contrast with alternative optimization algorithms.

3.2.1. Exhaustive search: foundational approach

Exhaustive search, implemented as the BruteForceSampler (EXS) [35], systematically evaluates every possible configuration in a discrete design space to identify the global optimum [20]. In structural engineering, it is applied to tasks like selecting truss element profiles to minimize objectives such as mass while meeting structural requirements [49]. Its core strength is guaranteeing the global optimum, serving as a benchmark to validate scalable methods like TPE in this thesis [24].

The computational complexity of EXS grows exponentially with the number of elements (k) and profiles (n), yielding (n^k) configurations and resulting in an extremely rapid expansion of the search space that renders it computationally infeasible for large-scale problems [9]. Nevertheless, exhaustive search is valuable for smaller problems where resources allow comprehensive exploration, offering transparency that assures stakeholders of solution optimality [44]. In this thesis,exhaustive search is operated for verification of the effectiveness of scalable optimization.

3.2.2. Heuristics

In optimization, heuristics represent a category of algorithms designed to tackle complex problems by prioritizing efficiency over exhaustive computation. Unlike exact methods that systematically explore the entire search space to guarantee an optimal solution, heuristics employ practical, problem-specific strategies to quickly identify satisfactory solutions, often at the expense of optimality. This trade-off makes them particularly valuable in engineering contexts, such as steel structures optimization, where computational resources and time are constrained [32]. Among heuristic approaches, the Random-Sampler stands out for its straightforward implementation [9].

Random-Sampler

The Random-Sampler, implemented in Optuna [35], is a heuristic method that randomly selects configurations from the discrete design space without employing learning or adaptation mechanisms [7]. In truss optimization, Random-Sampler assigns HEA profiles to elements, evaluating objectives such as mass while ensuring structural constraints, such as (UC ≤ 1) [49].

Nevertheless, Random-Sampler lack of guided exploration leads to inefficient sampling, particularly in high-dimensional spaces. The probability of identifying near-optimal configurations decreases exponentially with the number of elements (k) and profiles (n), often requiring an unbounded number of trials to approach optimality. This inefficiency contrasts with TPE model-based approach, which leverages Bayesian optimization to prioritize promising configurations [7] [35]. In this thesis, Random-Sampler serves as a baseline to evaluate the effectiveness of more sophisticated methods like TPE for the 9-element truss benchmark, highlighting the need for guided search strategies in complex discrete optimization tasks.

3.2.3. Metaheuristics

Metaheuristics are iterative, guided optimization strategies designed to explore complex search spaces and find near-optimal solutions without guaranteeing global optimality [20]. Unlike heuristics, which rely on simple, problem-specific rules, metaheuristics employ structured mechanisms to balance exploration and exploitation, making them suitable for high-dimensional, discrete problems such as steel truss optimization [37]. This section evaluates three metaheuristic algorithms implemented in Optuna [35]: Non-dominated Sorting Genetic Algorithm II (NSGA-II), Covariance Matrix Adaptation Evolution Strategy (CMA-ES) and Tree-structured Parzen Estimator (TPE). These are assessed for their ability to optimize problems similar to the 9-element truss benchmark, while balancing mass, symmetry, connection degree, and beam continuity with TPE emphasized for its Bayesian efficiency[8].

NSGA-II (NSGAIISampler)

The Non-dominated Sorting Genetic Algorithm II (NSGA-II), implemented as NSGAIISampler in Optuna [35], is a population-based metaheuristic inspired by natural selection. It evolves a set of solutions through selection, crossover, and mutation, excelling in multi-objective optimization by generating Pareto fronts via non-dominated sorting and crowding distance [9]. NSGA-II supports categorical parameters, such as HEA profile assignments, and typically requires high trial budgets (100 – 10,000) to converge [35]. In structural engineering, it has been applied to problems like dike geometry optimization, balancing cost and stability [24].

However, NSGA-II struggles with topologically driven objectives like symmetry or connection degree, as genetic encodings may fail to capture spatial dependencies in truss design [48]. Small changes in encoding can lead to significant structural variations, slowing convergence [11]. Its stochastic nature demands extensive parameter tuning, and high trial budgets increase computational costs [17]. In this thesis, NSGA-II's limitations with discrete, topological objectives make it less suitable than TPE for the 9-element truss benchmark [11].

CMA-ES (CmaEsSampler)

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES), implemented as CmaEsSampler in Optuna [35], is an evolutionary metaheuristic that adapts a multivariate normal distribution to sample promising configurations. It excels in continuous optimization but supports categorical parameters through encoding, making it applicable to truss profile assignments [31]. CMA-ES typically requires trial budgets of (1,000 – 10,000), reflecting its iterative adaptation of the covariance matrix to focus on high-performing regions [35].

CMA-ES has been used for enignnering optimization, often balancing conflicting objectives [22], however, CMA-ES high trial requirements and focus on continuous spaces make it less efficient for discrete, topological objectives like connection degree or symmetry. Its encoding for categorical variables can introduce complexity, and convergence may be slower for the benchmark model's discrete search space [34]. In this thesis, CMA-ES is considered a robust alternative but is outperformed by TPE lower trial needs and pruning capabilities [35].

TPE (TPESampler)

The Tree-structured Parzen Estimator (TPE), implemented as TPESampler in Optuna [35], is a Bayesian metaheuristic that models the probability distribution of high-performing configurations using kernel density estimators [7]. Follwoing [36], TPE splits past trials into good and poor performing groups, sampling new configurations to maximize expected improvement as represented in Equation 4.8. It supports categorical parameters, pruning of unpromising trials, and low trial budgets (100 – 1,000), making it highly efficient for discrete spaces [35]. In truss optimization, TPE handles mixed objectives (e.g., mass, symmetry, connection degree) by modeling hierarchical constraints, such as profile assignments dependent on spatial constraints [8].

Compared to NSGA-II and CMA-ES, TPE converges faster due to its probabilistic sampling and pruning, requiring fewer evaluations for the 9-element truss 1.95 million configurations [47]. Its implementation in Optuna, with settings like 100 startup trials and seed=42, ensures reproducibility and efficiency (Section 4.5.2). TPE ability to handle weighted scalarization aligns with stakeholder preferences, making it the primary optimizer in this research, validated against EXS for superior performance in discrete, non-smooth landscapes [49].

3.2.4. Comparison of optimization methods and selection

This section evaluates five optimization algorithms: Exhaustive Search (EXS), Random-Sampler, Non-dominated Sorting Genetic Algorithm II (NSGA-II), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), and Tree-structured Parzen Estimator (TPE), implemented within the Optuna framework [35]. The evaluation examines their suitability for optimizing a benchmark 9-element steel truss, which involves assigning HEA profiles to minimize objectives, including structural mass, symmetry, connection degree, and beam continuity, while satisfying constraints such as unity check threshold (UC ≤ 1).

To facilitate a systematic comparison, Table 3.2 consolidates the characteristics of the methods discussed in Sections 3.2.1 to 3.2.3, focusing on support for categorical parameters, pruning capabilities, multi-objective optimization potential, constraint handling mechanisms, and recommended trial budgets.

Table 3.1: Qualitative Comparison of Optimization Methods

Method	Categorical Parameters	Pruning	Multi-Objective	Constraint Handling	Recommended N_{trials}
Exhaustive Search	Yes	No	No	Direct	All possible
Random-Sampler	Yes	No	No	Via objective function	High ($>$10,000)
NSGA-II	Yes	No	Yes	Via genetic mechanisms	100–10,000
CMA-ES	Via encoding	No	No	Via penalty functions	1,000–10,000
TPE	Yes	Yes	Via scalarization	Via objective function	100–1,000

Table 3.2: Qualitative Comparison of Optimization Methods. Note: “Categorical Parameters” indicates direct support for discrete HEA profile assignments. “Pruning” denotes the ability to terminate unpromising trials early. “Multi-Objective” reflects support for multiple objectives, though the study employs weighted scalarization. “Constraint Handling” specifies the mechanism for enforcing UC $\leq$ 1. “Recommended N_{trials}” indicates the trial budgets suggested by Optuna’s sampler documentation [35].

The following analysis details each method’s performance, highlighting their strengths and limitations for the truss optimization problem. EXS ensures global optimality but is computationally infeasible for large search spaces [9]. Random-Sampler provides simplicity but lacks efficiency due to unguided sampling [20]. NSGA-II excels in multi-objective optimization but faces challenges with topological objectives, such as symmetry [11] [49]. CMA-ES, designed for continuous optimization, is less effective for discrete problems [31]. TPE offers superior efficiency through categorical parameter support, pruning, and reduced trial requirements, making it well-suited for the scalarized objective function used in this study [7] [47].

Exhaustive Search (EXS):

EXS, implemented as BruteForceSampler, evaluates all possible configurations , guaranteeing global optimality with exponential time complexity proportional to (n^k) [9]. It supports categorical parameters and directly enforces constraints but lacks pruning capabilities, rendering it suitable only as a validation benchmark rather than a practical optimization tool [48].

Random-Sampler

Random-Sampler employs random configuration sampling, supporting categorical parameters and managing constraints through objective function penalties [8]. Its lack of guided exploration necessitates high trial budgets (>10,000), reducing efficiency in high-dimensional search spaces [35]. It serves as a baseline for comparison.

NSGA-II (NSGAIISampler):

NSGA-II, implemented as NSGAIISampler, excels in multi-objective optimization by generating Pareto fronts, supporting categorical parameters, and handling constraints via genetic mechanisms [11] [17]. Its requirement for 100 – 10,000 trials and difficulties with topological objectives, such as symmetry, limit its efficiency for the scalarized objective function used in this study [35].

CMA-ES (CmaEsSampler):

CMA-ES, implemented as CmaEsSampler, is optimized for continuous spaces, supporting categorical parameters through encoding and constraints via penalty functions [34]. Its demand for (1,000 – 10,000) trials and the complexity introduced by encoding reduce its suitability for discrete truss optimization [35].

TPE (TPESampler):

TPE, implemented as TPESampler, leverages Bayesian optimization with kernel density estimators, supporting categorical parameters, pruning unpromising trials, and handling constraints through objective function penalties [7] [36]. It converges efficiently within (100 – 1,000) trials (Section 5.3) , aligning with the scalarized objective function and stakeholder preferences [47].

The Tree-structured Parzen Estimator (TPE) is designated as the primary optimization algorithm for this study based on its superior performance in addressing the requirements of discrete, constrained structural optimization for the benchmark 9-element truss. This selection is justified by the following attributes:

- Direct support for categorical parameters, enabling efficient HEA profile assignments.
- Pruning capability, which minimizes computational costs for expensive structural evaluations.
- Low trial budget requirement, enhancing efficiency compared to NSGA-II and CMA-ES.
- Compatibility with weighted scalarization, facilitating alignment with stakeholder requirments.

3.3. Benchmarking and validation in optimization studies

Structural engineers typically evaluate optimization algorithms using a range of performance metrics that extend beyond the quality of the final solution. These include convergence speed, the accuracy of achieved objective values, diversity across solution sets, and computational efficiency. Collectively, these indicators help assess whether an algorithm is practically viable and reliable for solving real-world design problems. Such metrics are standard in the evaluation of multi-objective optimization frameworks and frequently used in comparative studies [47]. An often endorsed method for algorithm validation is to use an exhaustive search on small-scale problems as a benchmark for accuracy. Such an approach allows researchers to measure how closely heuristic or surrogate-based methods approximate known optima. Studies by [15] have found that using exhaustive enumeration is useful for assessing how effectively algorithms like genetic algorithms and Bayesian methods perform in specific design areas. The transition from brute-force techniques to probabilistic optimizers, such as the Tree-structured Parzen Estimator (TPE), necessitates benchmarking due to randomness and conditional sampling in stochastic algorithms. In the absence of ground truth validation or performance curves, algorithm TPE outputs might show an absence of interpretability or statistical robustness [9].

Several studies [17] [28] were operating benchmarking with tools like convergence plots, averaging results from several runs, and comparing performance indicators. These practices improve the credibility of findings and provide insights into solution stability and algorithm sensitivity to hyperparameter settings. Performance plots showing the best objective value over trials, for instance, are a standard approach to evaluating convergence behavior.

This research uses an exhaustive search for small-scale, single-objective instances to evaluate the advanced algorithm. A complete enumeration of all viable design variants was generated utilizing a 9-element steel truss benchmark structure featuring a set of five distinct HEA profile types. The optimal solution identified by exhaustive search served as the baseline for evaluating the TPE performance regarding convergence rate, optimization of objectives, and similarity of truss configurations. This baseline technique prioritizes repeatability, interpretability, and method-specific performance diagnostics while complying with the highest standards in structural optimization.

3.4. Research gap

The literature reviewed in the previous sections reveals several unresolved challenges and blind spots in the application of optimization algorithms to structural engineering problems, particularly in the domain of steel truss design. These can be broadly categorized into academic research gaps and methodological shortcomings.

Despite the growing prominence of Bayesian optimization techniques, their application in civil engineering contexts - especially in steel frame or truss design - remains limited. While studies such as [7] [47] have shown the robustness of the Tree-structured Parzen Estimator (TPE) in complex objective landscapes, its adoption in engineering community has lagged behind more conventional heuristics [20]. Similarly, the multi-objective frameworks used in most prior works tend to group objectives of similar structural nature typically continuous, scalarized metrics like weight or deflection. Few studies engage with structurally diverse objectives that span both quantitative and topological concerns, such as symmetry or connection degree [27].

While grouping strategies have been explored in automated design tools [46], these often exclude practical constraints such as connection complexity or continuity in beam layouts. Additionally, most literature does not incorporate stakeholder-informed weighting schemes when balancing design objectives. Instead, uniform weights or a posteriori Pareto ranking are often assumed, limiting the interpretability and real-world applicability of results [48].

Multi-objective optimisation has substantially benefited from genetic algorithms, particularly NSGA-II. Their performance decreases when addressing goals reliant on layout or logical grouping: symmetry, connection quantity, or member continuity. Research by [28] and [49] emphasizes that while effective for continuous problems, these algorithms show insensitivity to topologically structured objectives due to encoding constraints and stochastic search mechanisms. Moreover, benchmarking is frequently conducted on the basis of single performance metrics or convergence speed alone. The use of exhaustive search as a validation tool in civil engineering is limited to only a few papers, as noted by [50], despite its importance in verifying the optimality of results in tractable problem instances. Furthermore, statistical robustness, such as averaging over multiple runs to ensure reliable performance, is rarely emphasized, as highlighted by [31] which undermines the trustworthiness of results in stochastic approaches like Bayesian optimization or genetic algorithms.

Building on these identified gaps, this study proposes a tailored optimization framework based on the Tree-structured Parzen Estimator (TPE) to address the challenges inherent in grouping-based truss design. It introduces a method that accounts for both structural performance and constructability, guided by stakeholder informed weighting.

4 Methodology

4.1. General workflow

This thesis utilizes a systematic methodology, starting with the creation and structural analysis of 2D truss models in Grasshopper software [41], proceeding through the quantification of performance objectives, and concluding with the implementation of optimization algorithms to determine optimal truss configurations. This methodology focuses primarily on a parametric model developed in the Grasshopper environment, utilizing the Karamba3D structural analysis plugin. This model simulates the structural behavior of steel trusses subjected to uniform loading conditions and provides element-specific performance indicators, including the Unity Check (UC) values, which are essential for evaluating structural safety.

For more detailed analysis and optimization, outputs from Grasshopper are converted into structured Excel spreadsheets, which function as inputs for a Python-based parametric model. This secondary model manages the definition and equilibrium of multiple design objectives and employs both exhaustive search and Tree-structured Parzen Estimator (TPE) techniques for optimization. A detailed workflow is illustrated in Figure 4.1, indicating the progression from structural modeling to optimization and result validation.

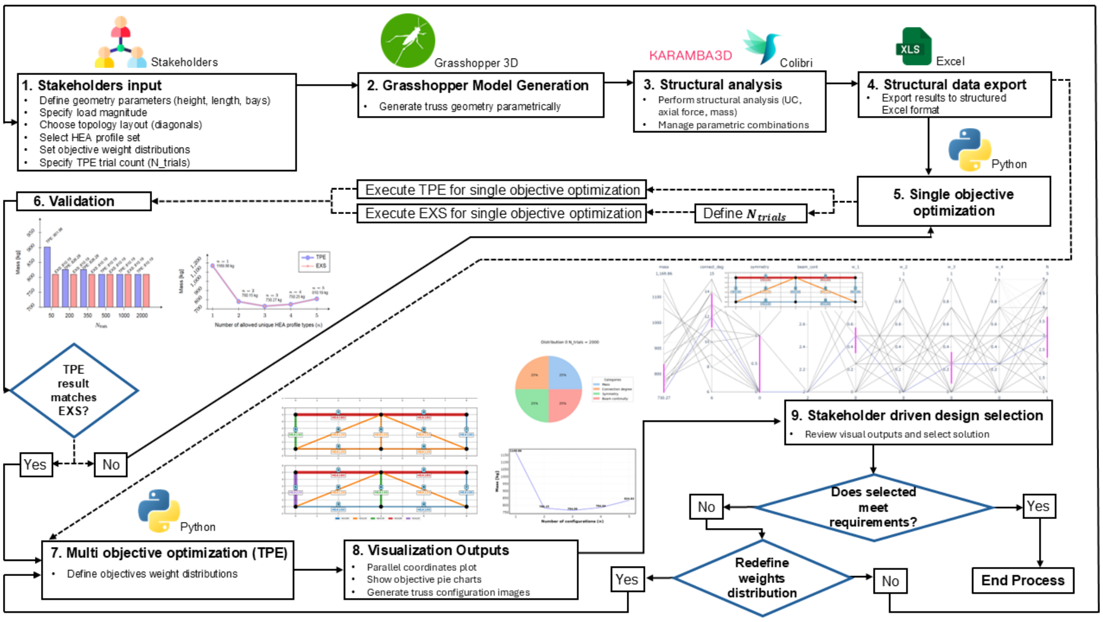

Stakeholders
1. Stakeholders input
Define geometry parameters (height, length, bays)
Specify load magnitude
Choose topology layout (diagonals)
Select HEA profile set
Set objective weight distributions
Specify TPE trial count (N_trials)
Grasshopper 3D
2. Grasshopper Model Generation
Generate truss geometry parametrically
KARAMBA3D
Colibri
3. Structural analysis
Perform structural analysis (UC, axial force, mass)
Manage parametric combinations
Excel
4. Structural data export
Export results to structured Excel format
Python
5. Single objective optimization
Execute TPE for single objective optimization
Execute EXS for single objective optimization
Define N_{trials}
6. Validation
Mass [kg]
N_{trials}
TPE
EXS
Number of allowed unique HEA profile types (n)
TPE result matches EXS?
Yes
No
Distribution 0 N_trials = 2000
Categories
Mass
Connection degree
Symmetry
Beam continuity
mass
connect_deg
symmetry
beam_cont
Number of configurations (n)
Python
7. Multi objective optimization (TPE)
Define objectives weight distributions
8. Visualization Outputs
Parallel coordinates plot
Show objective pie charts
Generate truss configuration images
9. Stakeholder driven design selection
Review visual outputs and select solution
Does selected meet requirements?
No
Yes
Redefine weights distribution
Yes
No
End Process

Figure 4.1: General workflow illustrating the methodology employed in optimizing steel truss grouping

4.2. Truss model generation and parameterization

The structural system analyzed is a two-dimensional rectangular steel truss, consisting of nine members and subjected to a uniformly distributed load 24 kN/m, as specified in the Grasshopper setup (Figure B.1). The truss was modeled using Grasshopper, a visual programming plugin for Rhino [41], enabling the allocation of HEA steel profiles to each element. The structural boundary conditions were established to inhibit global translational and rotational movements at the supports, with pinned joints assumed for all connections, ensuring a statically determinate structure. While lateral bracing was presumed to prevent buckling from being the governing factor, hence allowing to focus primarily under axial force circumstances (tension and compression) without considering bending moments.

The internal force of each element was estimated using Karamba3D [14], a finite element analysis (FEA) plugin integrated into Grasshopper, which computes axial forces and evaluates structural performance. Unity Check (UC) values were determined as the ratio of internal normal force to design resistance of each element, ensuring compliance (UC ≤ 1) threshold. The (UC) assessment presumes a consistent profile assignment across all elements in each iteration, hence streamlining computation and facilitating scalability analysis across various profile combinations.

The structural behavior is influenced by the distribution of profile sizes, however, the optimization model permits mixed-profile configurations by using (UC) values initially derived from single-profile assessments. This introduces a key assumption: although changes in individual element profiles would affect internal force distribution, in reality the optimization phase treats (UC) values as fixed inputs to ensure low computational effort. This simplification was essential to accommodate the computing requirements of assessing a relative extensive design space, with a minimum of ($n = 5$) allowed HEA profiles across ($k = 9$) elements, resulting in 5^9 turning approximately 1.95 million possible combinations.

The Grasshopper model, as shown in Figure B.1, allows for parametric adjustment of the truss geometry, including the truss length (12.00 m), number of bays (4 bays), pattern type (#4), truss depth (0.250 m), and applied load (uniformly distributed load of 24 kN/m). These parameters enable the exploration of different rectangular truss typologies while maintaining a consistent structural framework. The geometry setup feeds into the Karamba3D plugin, which defines the structural components - posts (columns), chords (beams), diagonals, and support points, for finite element analysis (FEA) (Appendix B.1).

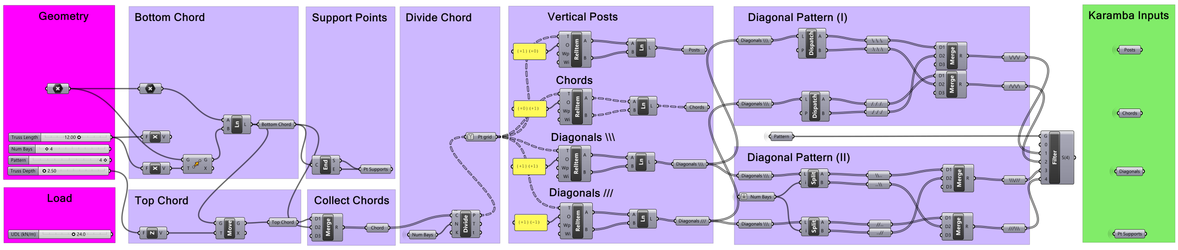


Figure 4.2: Grasshopper setup defining geometry and load input for parametric truss analysis, input into Karamba for structural analysis.

Parallel to the geometry and load inputs, the Grasshopper model incorporates material and profile definitions, as illustrated in Appendix B.2. The material is specified as S355 structural steel with a yield stress of 355 MPa, commonly used in European structural applications due to its balance of strength and ductility. Following from Appendix B.2, a set of five HEA profiles (HEA140, HEA160, HEA200, HEA220, HEA240) are iterated ($n = 5$), allowing for a controlled exploration of profile assignments across the truss elements. However, these profiles are not the main ones used for the benchmark problem, which instead uses a different set of HEA profiles as defined in the experimental setup Chapter 5. These profiles are selected for their symmetry and standardized dimensions, facilitating bolted or welded connections in practical applications [21].

The analysis results are visualized using the Colibri plugin within Grasshopper [43], as shown in Appendix B.3. Colibri enables the aggregation and iteration of design data across multiple configurations, exporting results such as element IDs, node coordinates, (UC) values, and structural mass into CSV files, which are later converted into Excel sheets for further processing. This setup allows for the visualization of axial force distribution and deflection under various loading conditions, as followed from Appendix A.1, which illustrates different trusses typologies' deflection under uniformly distributed loads with different magnitudes varying from 20 to 40 kN/m.

Subsequently, the data generated from Grasshopper was transferred into Excel sheets via the Colibri plugin, specifying each element's ID, associated nodes, node coordinates, (UC) values, and self-weight (structural mass) represented in Chapter 5. This structured data format facilitated seamless integration with the Python-based parametric optimization model, which uses the Tree-structured Parzen Estimator (TPE) for multi-objective optimization.

4.3. Definition of optimization objectives

This thesis's optimization approach is determined by four basic design objectives that combine structural performance and structural complexity — mass, degree of connection, symmetry, and beam continuity. The objectives were chosen based on literature review findings on structural optimization and the practical aspects of modular construction, transportability, and assembly [42].

Each objective is calculated for a given truss configuration, conditional on the element grouping strategy and profile assignments. The mass target is primarily established in engineering design, however the complexity related objectives — connection degree, symmetry, and beam continuity — underscore the requirement for more constructible and standardized solutions.

- **Mass** represents the aggregate self-weight of the structure and has a significant correlation with environmental impact and material expenditure. Lightweight constructions are favored for their sustainability and cost-effectiveness against heavyweight structures.
- **Connection Degree** measures the complexity of joints by measuring the number of unique profiles that intersect at each node. An increased degree of diversity within a node contributes to greater complexity in both fabrication and assembly processes.
- **Symmetry** describes the degree to which the truss configuration is evenly mirrored along its vertical axis. A symmetric layout is typically more straightforward to manufacture, visualize, and assemble.
- **Beam Continuity** applies to the consistency of horizontal members. Configurations utilizing the identical profile across successive horizontal segments are structurally and logistically advantageous.

The following sub-sections provide a detailed discussion of each objective, including their quantification, relevance, and balance through optimization [10].

4.3.1. Mass

Decreasing structural mass is a fundamental goal in most structural design challenges, as it directly relates to material efficiency, environmental impact, and construction costs. In the context of steel truss design, the total mass is the sum of the masses of all individual elements, determined by their length, cross-sectional area, and material density.

From a sustainability perspective, mass minimization aligns with two key priorities in contemporary construction: reducing embodied carbon and transportation costs [42]. Additionally, a lower self-weight reduces stress on supporting systems and foundations [21]. Despite its advantages, prioritizing volume alone can lead to highly customized profile designs, which in turn complicate manufacturing and logistics. Mass minimization is therefore considered one of several goals within a broader multi-objective optimization framework.

In this study, the total structural mass is treated as a primary objective in the optimization process. The parametric model uses predefined HEA steel profiles, each with a known weight per meter. The mass of each structural member is determined by multiplying its length by the cross-sectional area and material density. Lighter HEA profiles are typically favored during optimization, assuming they comply with the structural constraints established by (UC).

The mass objective function f_{M} is defined as:

$$f_{\mathsf{M}} = \sum_{i=1}^{k} \rho A_i L_i \tag{4.1}$$

where:

- f_{M} denotes the total structural mass [kg].
- k is the number of truss elements (in this study, $k = 9$).
- ρ is the material density of S355 structural steel kg/m³.
- A_i is the cross-sectional area [m^2] of the HEA profile assigned to element i.
- L_i is the length [m] of element i, extracted from the geometric definition of the truss.

This objective is minimized under the structural constraint:

$$\mathsf{UC}_i \leq 1 \quad \forall i \in \{1, 2, \ldots, k\} \tag{4.2}$$

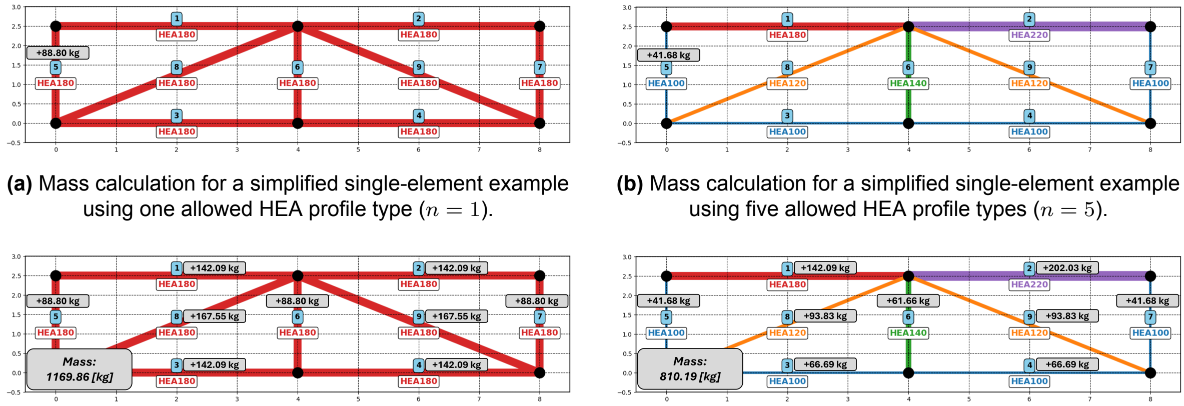
where UC$_i$ is the unity check of element i, ensuring that all members remain within allowable stress limits under loading conditions. Lighter profiles are generally favored during optimization, as long as they satisfy this constraint.

To investigate the influence of design standardization, a parameter (n) is introduced to limit the number of unique HEA profile types allowed in a given solution:

- For $n = 1$, all elements are constrained to share the same profile type (e.g., HEA180).
- For $n = 5$, up to five distinct profiles (e.g., HEA100 to HEA220) can be assigned, enabling more refined sizing.

The impact of this design freedom on structural mass is illustrated in Figure 4.3, which presents a step-by-step representation of the mass calculation procedure:

- Subfigures 4.3a and 4.3b depict a simplified single-element case for $n = 1$ and $n = 5$, respectively.
- Subfigures 4.3c and 4.3d extend the evaluation to the full truss, showing the cumulative mass difference between uniform and diverse profile assignments.



(a) Mass calculation for a simplified single-element example using one allowed HEA profile type ($n = 1$).

(b) Mass calculation for a simplified single-element example using five allowed HEA profile types ($n = 5$).

(c) Mass calculation for the full truss using one allowed HEA profile type ($n = 1$).

(d) Mass calculation for the full truss using five allowed HEA profile types ($n = 5$).

Figure 4.3: Step-by-step representation of the Mass calculation procedure for $n = 1$ and $n = 5$, first applied to a simplified single-element case and then to the full truss.

As evident in Figure 4.3, increasing number of allowed HEA profiles (n) generally leads to a more mass-efficient layout. This formalized calculation process serves as a reference throughout the thesis, with a more detailed elaboration on the mass calculation procedure provided in Appendix F. Next Mass optimization results are analyzed in detail in Chapter 6.

4.3.2. Connection Degree

Connection Degree quantifies the complexity of joint design by measuring the number of distinct profiles converging at each node. In structural and fabrication terms, highly standardized joints where identical or similar elements converge are easier and more cost-effective to assemble [45]. In contrast, joints connecting a large number of dissimilar members tend to increase fabrication difficulty and introduce geometric constraints. Specifically, varying HEA profile dimensions necessitate customized connection details, such as tailored plates or adjusted weld configurations, which elevate fabrication costs and complexity [1]. Besides, aligning dissimilar profiles to minimize nodding eccentricity is challenging, often requiring reinforcement to ensure stable load transfer, further complicating joint design [16].

This structural complexity objective is computed using undirected graph theory [26], where each node represents a connection and edges represent truss elements. This follows the classic vertex degree formulation, where complexity increases with profile heterogeneity at a given node. The increased manufacturing effort and detailed complexity result in penalties for nodes connecting truss components of various profiles.

The truss is represented as an undirected graph $G = (V, E)$, where the set of nodes V corresponds to joints in the truss and the set of edges E corresponds to structural elements. Each edge connects two vertices (nodes) and represents a single truss member.

Let $d(v_j)$ denote the degree of node v_j, defined as the number of unique elements connected to node v_j. The Connection Degree objective function f_{CN} is defined as:

$$f_{CN} = \sum_{j=1}^{m} d(v_j) \tag{4.3}$$

where:

- f_{CN} represents the total connection complexity.
- m is the number of nodes in the truss (here, $m = 6$).
- $d(v_j)$ is the number of edges (i.e., truss elements) incident to node v_j.

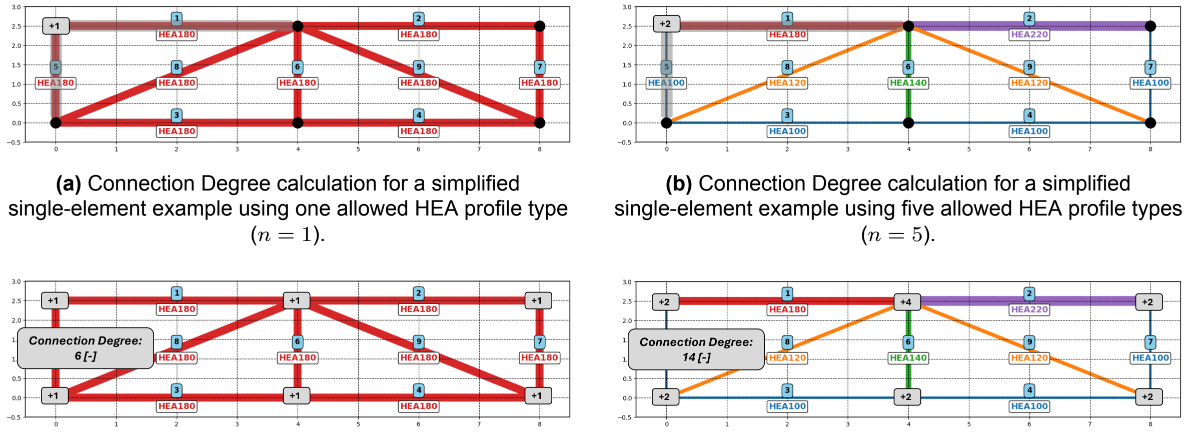
This formulation captures the joint complexity rather than individual element characteristics. Higher values of f_{CN} indicate more connections across nodes, which may correspond to structurally denser regions and more intricate fabrication challenges.

To evaluate the role of profile assignment on connection complexity, the parameter (n) is once again introduced to control the number of distinct HEA profiles allowed in the design:

- For $n = 1$, profile types are consistent, and joints typically merge identical elements.
- For $n = 5$, element diversity increases, leading to more irregular node connectivity patterns.

The impact of this design freedom on connection complexity is illustrated in Figure 4.4, which presents a step-by-step calculation of the Connection Degree objective:

- Subfigures 4.4a and 4.4b apply the procedure to a simplified single-element case with $n = 1$ and $n = 5$, respectively.
- Subfigures 4.4d and 4.4d extend the procedure to the full truss. The total connection degree rises from 6 (standardized configuration case) to 14 (unique configuration case), highlighting how allowing more distinct profiles can increase node complexity.



(a) Connection Degree calculation for a simplified single-element example using one allowed HEA profile type $(n = 1)$.

(b) Connection Degree calculation for a simplified single-element example using five allowed HEA profile types $(n = 5)$.

(c) Connection Degree calculation for the full truss using one allowed HEA profile type $(n = 1)$.

(d) Connection Degree calculation for the full truss using five allowed HEA profile types $(n = 5)$.

Figure 4.4: Step-by-step representation of the Connection Degree calculation procedure for $n = 1$ and $n = 5$, first applied to a simplified single-element case and then to the full truss.

This objective reflects constructability, fabrication clarity, and standardization. The calculation logic is consistently implemented across all design variants and serves as a reference for the evaluation and trade-off analysis in Chapters 6. A more detailed breakdown of the graph formulation and node-edge mapping is provided in Appendix F.

This process is supported by the underlying `get_connection_degree()` implemented in the Python code, which incorporates node positions V and edge attributes. Additionally, the calculation of the connection degree metric is handled using the `create_graph()` program function, presented in Appendix F. This representation is foundational to all structural complexity objectives, including connection degree, symmetry, and beam continuity.

4.3.3. Symmetry

Symmetry is a design preference that improves both visual clarity and structural viability. Symmetrical structures are frequently preferred in civil engineering because of their aesthetic equilibrium, the potential for modular repetition in construction, and the simplicity of design communication.

The symmetry metric is calculated in this thesis by first identifying a central vertical axis, which is referred to as the mirror axis, and subsequently evaluating the quantity of geometrically mirrored element pairs along this axis. The symmetry score is reduced by deviations from this ideal, whereas a perfectly symmetric truss configuration would have all corresponding members mirrored along the centerline.

From a construction perspective, symmetry simplifies fabrication, as emphasized in [48] [49]. This thesis advocates for symmetric configurations due to their simplicity in modular truss assembly and manufacturing efficiency, which enables mirrored elements to utilise shared detailing and production methods. In addition, the on-site installation of symmetric trusses is advantageous because of their generally more predictable load paths and much simpler orientation [19]. Mass efficiency is essential for the optimisation of building processes and the promotion of standardisation, despite the fact that it may not always be in alignment with symmetry.

In the context of truss design, the symmetry objective penalizes deviations from an ideal mirrored geometry relative to a central vertical axis. This axis of symmetry is defined explicitly, and corresponding pairs of mirrored elements are identified. Asymmetry is introduced when a mirrored pair is assigned different profiles, indicating a deviation from the intended symmetric configuration.

Let f_{SYM} denote the symmetry score. For each mirrored pair (e_i, e_j), a mismatch in profile assignment increases f_{SYM} by 1. Perfect symmetry yields a score of 0[-]. Thus, the symmetry objective is given by:

$$f_{SYM} = \sum_{(e_i,e_j)\in P} \delta(e_i, e_j) \tag{4.4}$$

where:

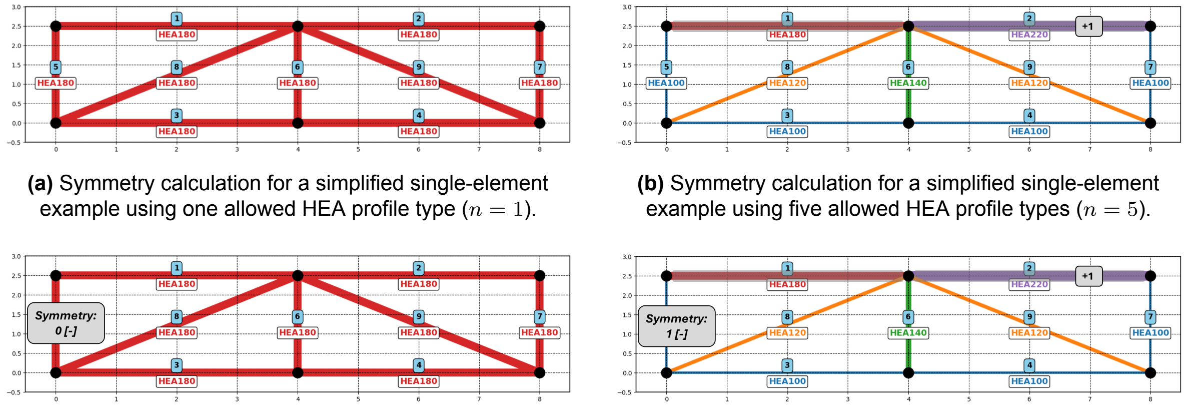
- P is the set of all mirrored element pairs;
- $\delta(e_i, e_j) = 1$ if the profiles assigned to e_i and e_j differ, and 0 otherwise.

The effect of increasing profile diversity on symmetry is controlled by the parameter (n):

- For $n = 1$, all elements use the same profile and perfect symmetry is achieved.
- For $n = 5$, greater profile diversity allows for more tailored configurations but typically results in a higher number of mismatches.

The impact of this design freedom on symmetry is illustrated in Figure 4.5, which presents the symmetry calculation procedure:

- Subfigures 4.5a and 4.5b present a simplified case showing the logic behind profile matching for $n = 1$ and $n = 5$, respectively.
- Subfigures 4.5c and 4.5d show the symmetry results for the full truss: a perfect match ($f_{\text{SYM}} = 0$) for the uniform profile case and a penalty ($f_{\text{SYM}} = 1$) when mirrored elements differ.



(a) Symmetry calculation for a simplified single-element example using one allowed HEA profile type ($n = 1$).

(b) Symmetry calculation for a simplified single-element example using five allowed HEA profile types ($n = 5$).

(c) Symmetry calculation for the full truss using one allowed HEA profile type ($n = 1$).

(d) Symmetry calculation for the full truss using five allowed HEA profile types ($n = 5$).

Figure 4.5: Step-by-step representation of the Symmetry calculation procedure for $n = 1$ and $n = 5$, first applied to a simplified single-element case and then to the full truss.

The symmetry objective contributes to design regularity and aesthetic harmony. It is particularly relevant when prefabrication or modularity is desirable. A more detailed implementation of the symmetry evaluation is available in Appendix F. Similarly to the connection degree, symmetry relies on the graph representation of the truss, created via the `create_graph()` presented in Appendix F. The scoring logic itself is computed using the `get_symmetric_metric()` presented in Appendix F.

4.3.4. Beam Continuity

Beam continuity refers to the uninterrupted use of identical profile types along aligned horizontal members within the truss structure. It is introduced as a practical measure of modularity and construction simplicity, with a particular focus on the fabrication and transportation of horizontal beams. By encouraging profile uniformity along key structural directions, beam continuity supports both the mechanical performance and the efficiency of downstream construction processes.

This objective is especially relevant for structural assemblies that benefit from prefabrication, as continuous horizontal beams can be produced, transported, and installed as single components. This reduces the need for on-site splicing, shortens installation time, and may reduce costs related to material handling [42]. Conversely, fragmented beam continuity due to frequent profile changes introduces logistical challenges and diminishes constructability.

The beam continuity metric is evaluated by analyzing the horizontal elements in each row of the truss and identifying whether the same profile type is used consecutively. When a continuous beam is formed using the same profile across two or more horizontal elements, it is considered a continuity gain. Disruptions in continuity, such as changes in profile type reduce the continuity score.

Let f_{BCN} denote the beam continuity score. The objective is defined such that each segment of continuous horizontal alignment contributes positively to the total:

$$f_{BCN} = \sum_{r=1}^{R} \gamma_r \tag{4.5}$$

where:

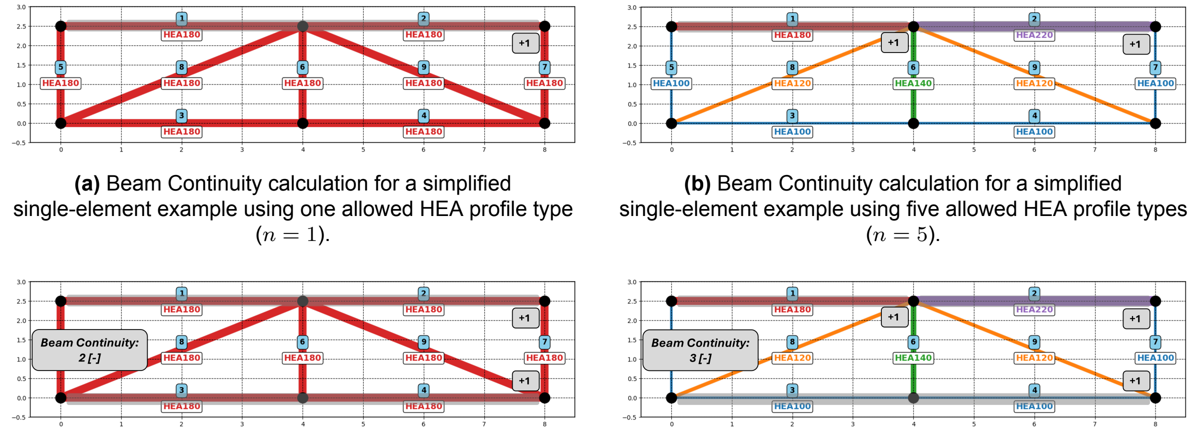
- R is the number of horizontal rows evaluated in the truss,
- γ_r is the number of uninterrupted profile matches found along row r.

The influence of profile diversity on continuity is again governed by the parameter (n):

- For $n = 1$, profile types are consistent along horizontal members, enabling the formation of continuous beams.
- For $n = 5$, element diversity increases, and continuity is frequently disrupted by changes in profile assignment.

The impact of this design freedom on continuity is illustrated in Figure 4.6, which presents a step-by-step calculation procedure:

- Subfigures 4.6a and 4.6b apply the logic to a simplified truss segment under uniform and varied profile conditions.
- Subfigures 4.6c and 4.6d extend the assessment to the full truss, demonstrating how the continuity score improves or deteriorates based on profile assignment.



(a) Beam Continuity calculation for a simplified single-element example using one allowed HEA profile type $(n = 1)$.

(b) Beam Continuity calculation for a simplified single-element example using five allowed HEA profile types $(n = 5)$.

(c) Beam Continuity calculation for the full truss using one allowed HEA profile type $(n = 1)$.

(d) Beam Continuity calculation for the full truss using five allowed HEA profile types $(n = 5)$.

Figure 4.6: Step-by-step representation of the Beam Continuity calculation procedure for $n = 1$ and $n = 5$, first applied to a simplified single-element case and then to the full truss.

This objective reflects constructability, fabrication clarity, and standardisation. The calculation logic is consistently implemented across all design variants and serves as a reference for the evaluation and trade-off analysis in Chapters 6. A more detailed implementation of the continuity metric is available in Appendix F. This process is directly supported by the underlying `create_graph()` function, which incorporates node positions and edge attributes. The continuity score is computed using the `get_beam_continuity_metric()` function presented in Appendix F.

4.4. Stakeholder weight input and objective balancing

In structural design, stakeholder preferences are critical for resolving conflicting objectives, including structural efficiency, constructability, and aesthetic clarity [33]. Stakeholders, such as structural engineers, manufacturers, architects, and clients, frequently emphasize varying criteria: reducing material consumption (mass), simplifying assembly (connection degree), amplifying aesthetic value (symmetry), or improving modularity (beam continuity). To integrate these diverse priorities into the multi-objective optimization framework, this thesis employs a weighted scalarization method, facilitating a systematic examination of trade-offs in rectangular steel truss design.

4.4.1. Weighted scalarization framework

Stakeholder preferences are formalized through a weight vector $\mathcal{W} = \{w_M, w_{CN}, w_{SYM}, w_{BCN}\}$, where each component corresponds to one of the four objectives: mass f_M, connection degree f_{CN}, symmetry f_{SYM}, and beam continuity f_{BCN}. The weights are normalized to satisfy:

$$w_M + w_{CN} + w_{SYM} + w_{BCN} = 1 \tag{4.6}$$

A composite performance weighted score is computed as:

$$score = w_M \cdot f_M + w_{CN} \cdot f_{CN} + w_{SYM} \cdot f_{SYM} + w_{BCN} \cdot f_{BCN} \tag{4.7}$$

where each objective function $f : \mathcal{S} \to \mathbb{R}$ is normalized to the range $[0, 1]$ to ensure comparability. This scalarization is implemented in the multi-objective Tree-structured Parzen Estimator (TPE) algorithms (Algorithms 3 and 4), enabling the ranking of candidate truss configurations based on stakeholder-defined priorities. In contrast, the exhaustive search (EXS) algorithms (Algorithms 1 and 2) evaluate objectives individually, rendering scalarization unnecessary. While scalarization unifies the objective values into a single score, the TPE algorithm explores the trade-offs between all objectives concurrently [47], whereas EXS would require separate iterations per objective, effectively treating them in isolation [20].

4.4.2. Flexibility in weight distributions

The optimization framework supports the evaluation of multiple weight distributions $\mathcal{W}$, ranging from small sets (e.g., 4-6 weights distributions) to extensive grids (e.g., 35 or 177 weights distributions for comprehensive trade-off analysis). This flexibility accommodates diverse stakeholder scenarios, from cost-driven designs to those prioritizing visual or modular clarity. Weight vectors are generated from predefined grids, such as:

- Coarse grid: $\{0.0, 0.25, 0.5, 0.75, 1.0\}$, yielding 35 valid combinations.
- Fine grid: $\{0.0, 0.1, 0.2, 0.25, 0.4, 0.5, 0.6, 0.75, 0.8, 0.9, 1.0\}$, yielding 177 combinations.
- Custom grids: Potentially thousands of combinations for granular exploration.

For the analysed results presented in Chapter 6, three specific weight distributions are evaluated:

- TPE($\mathcal{W}_1$): with distribution $\mathcal{W}_1 = \{0.25, 0.25, 0.25, 0.25\}$, prioritizing all four objectives equally.
- TPE($\mathcal{W}_2$): with distribution $\mathcal{W}_2 = \{0.4, 0.1, 0.4, 0.1\}$, moderately prioritizing *mass* and *symmetry* over *connection degree* and *beam continuity*.
- TPE($\mathcal{W}_3$): with distribution $\mathcal{W}_3 = \{0.1, 0.4, 0.1, 0.4\}$, moderately prioritizing *connection degree* and *beam continuity* over *mass* and *symmetry*.
- TPE($\mathcal{W}_4$): with distribution $\mathcal{W}_4 = \{0.4, 0.05, 0.5, 0.05\}$, heavily prioritizing *mass* and *symmetry* over *connection degree* and *beam continuity*.

These distributions, summarized in Table 6.5, guide the TPE optimization to produce designs aligned with specific stakeholder priorities, as visualized in multi-parallel plots (e.g., Figure 6.14).

The ability to evaluate multiple weight distributions enhances the framework's versatility but introduces significant computational challenges. For a 9-element truss with ($n = 5$) allowed HEA profiles, combining with 177 weight vectors results in a design space exceeding 345 million combinations. Chapter 6 emphasizes computational burden for varying numbers of weight vectors, where Table 6.10b highlighting the infeasibility of exhaustive search for large grids.

4.5. Optimization algorithm implementation

This section outlines the methodologies for Exhaustive Search (EXS) and Tree-structured Parzen Estimator (TPE) used to optimize steel truss configurations. Algorithms address single and multi-objective problem involving mass, connection degree, symmetry, and beam continuity, with EXS serving as a baseline for validation and TPE providing an efficient solution for complex search spaces [35].

4.5.1. Exhaustive search optimization implementation

Exhaustive Search systematically evaluates all possible configurations to identify the global optimum for a given objective. In this study, it is applied to a truss with ($k = 9$) elements, each assignable one of ($n = 5$) allowed HEA profiles, yielding 5^9 turning approximatelly 1.95 million configurations. This method optimizes each objective independently to establish baseline results. For the mass objective, we seek the configuration with the lowest total mass that satisfies the structural constraints (UC ≤ 1). For the structural complexity objectives connection degree, symmetry, and beam continuity all configurations achieving the best possible score for the respective objective are considered, subject to the feasibility condition that no element exceeds the (UC ≤ 1) threshold. Given the discrete nature of these metrics, multiple configurations may exhibit identical optimal scores. In such cases, a secondary selection criterion is applied: among the feasible and score optimal configurations, the one with the lowest total mass is selected. For example, consider two configurations with identical symmetry scores: one operating ($k = 9$) elements with HEA180 profile (lighter) and another operating ($k = 9$) elements with HEA220 profiles (heavier). The HEA180 configuration is preferred due to its lower mass, ensuring material efficiency when designs perform equally on complexity metrics.

However, due to the exhaustive nature of this method, which requires evaluating all possible configurations, it is computationally intensive and not scalable to larger problems or real-time applications. Therefore, while Exhaustive Search provides a valuable benchmark for validation, it is impractical for more complex scenarios, necessitating the use of more efficient optimization algorithms like TPE for larger-scale or multi-objective optimization tasks.

The optimization process proceeds through the following steps:

1. Evaluate the four objective functions for a given configuration.
2. Check the feasibility of the configuration by ensuring no element exceeds the (UC ≤ 1) threshold.
3. Record the best configuration based on the objective score.
4. Repeat for all configurations in the search space.

Exhaustive Search has been implemented in two algorithmic variants:

- Algorithm 1: EXS Search for Mass Optimization
- Algorithm 2: EXS Search for Structural Complexity Objectives Optimization

Algorithm 1 Exhaustive Search for Mass Optimization

```
Require: $\mathcal{P}$, $k$, $UC_{\text{max}}$, $f_M$

Input: Profile set $\mathcal{P} = \{\text{HEA100}, \text{HEA120}, \text{HEA140}, \text{HEA180}, \text{HEA220}\}$, number of elements $k = 9$, maximum Unity Check $UC_{\text{max}} = 1$, mass objective function $f_M : \mathcal{S} \to \mathbb{R}$
Initialize $\mathcal{S} \leftarrow \{s \mid s : \{1, \dots, k\} \to \mathcal{P}\}$    ▷ Search space of all configurations
Initialize $results \leftarrow \emptyset$    ▷ Store results for each $n$
for $n \in \{1, 2, 3, 4, 5\}$ do    ▷ Iterate over number of unique HEA profile types
    $\mathcal{S}_n \leftarrow \{s \in \mathcal{S} \mid |\text{unique}(s)| = n\}$    ▷ Filter configurations with exactly $n$ profiles
    Initialize $best\_config_n \leftarrow \emptyset$, $best\_value_n \leftarrow \infty$    ▷ Track best configurations for batch $n$
    for each $s \in \mathcal{S}_n$ do    ▷ Iterate over configurations in batch
        if $UC(s) \leq UC_{\text{max}}$ then    ▷ Check feasibility
            $value \leftarrow f_M(s)$    ▷ Compute Objective score $M$ [kg]
            if $value < best\_value_n$ then    ▷ Update if lighter
                $best\_config_n \leftarrow s$
                $best\_value_n \leftarrow value$
            end if
        end if
    end for
    $results \leftarrow results \cup \{n, best\_config_n, best\_value_n\}$    ▷ Store batch result
end for
return $results$    ▷ Return DataFrame with $n$, configuration, and objective result
```

Algorithm 2 Exhaustive Search for Structural Complexity Objectives Optimization

```
Require: $\mathcal{P}$, $k$, $UC_{\text{max}}$, $f$, $f_M$

Input: Profile set $\mathcal{P} = \{\text{HEA100}, \text{HEA120}, \text{HEA140}, \text{HEA180}, \text{HEA220}\}$, number of elements $k = 9$, maximum Unity Check $UC_{\text{max}} = 1$, objective function $f : \mathcal{S} \to \mathbb{R}$ (e.g., CN, SYM, BCN), mass objective function $f_M : \mathcal{S} \to \mathbb{R}$
Initialize $\mathcal{S} \leftarrow \{s \mid s : \{1, \dots, k\} \to \mathcal{P}\}$    ▷ Search space of all configurations
Initialize $results \leftarrow \emptyset$    ▷ Store results for each $n$
for $n \in \{1, 2, 3, 4, 5\}$ do    ▷ Iterate over number of unique HEA profile types
    $\mathcal{S}_n \leftarrow \{s \in \mathcal{S} \mid |\text{unique}(s)| = n\}$    ▷ Filter configurations with exactly $n$ profiles
    Initialize $best\_configs_n \leftarrow \emptyset$, $best\_value_n \leftarrow \infty$    ▷ Track best configurations for batch $n$
    for each $s \in \mathcal{S}_n$ do    ▷ Iterate over configurations in batch
        if $UC(s) \leq UC_{\text{max}}$ then    ▷ Check feasibility
            $value \leftarrow f(s)$    ▷ Compute objective score(e.g., $SYM$ [-])
            if $value < best\_value_n$ then    ▷ New best found
                $best\_value_n \leftarrow value$
                $best\_configs_n \leftarrow \{s\}$    ▷ Reset list with new best
            else if $value = best\_value_n$ then    ▷ Equal score, add to list
                $best\_configs_n \leftarrow best\_configs_n \cup \{s\}$
            end if
        end if
    end for
    $best\_config_n \leftarrow \arg\min_{s \in best\_configs_n} f_M(s)$    ▷ Secondary filtering by mass
    $results \leftarrow results \cup \{n, best\_config_n, best\_value_n\}$    ▷ Store batch result
end for
return $results$    ▷ Return DataFrame with $n$, configuration, and objective result
```

4.5.2. TPE optimization implementation

The Tree-structured Parzen Estimator (TPE) is a Bayesian optimization algorithm that efficiently explores large and complex search spaces by modeling the distribution of promising solutions [7]. Unlike

Gaussian Process-based methods, TPE estimates the density of configurations with high and low performance using non-parametric kernel density estimators (KDEs), which makes it particularly suited for discrete and categorical design variables like HEA profile types in steel trusses [8].

TPE operates by splitting past observations into two groups: a good-performing group and a poor-performing group, based on a quantile threshold (e.g., top 20% of previous trials) [36]. It constructs probability density $p(x \mid D^{(l)})$ and $p(x \mid D^{(g)})$ from these groups and then samples new candidates by maximizing the acquisition score, as shown in Equation 4.8:

$$\text{acquisition score} = \frac{p(x \mid D^{(l)})}{p(x \mid D^{(g)})} \tag{4.8}$$

This process favors exploring regions of the search space with a high likelihood of improvement.

In this thesis, TPE is applied to optimize a scalar performance score composed of four normalized objectives: mass, connection degree, symmetry, and beam continuity. TPE efficiently explores the design space by dynamically adapting to the structure of the objective landscape [47]. This TPE algorithmic configuration enables to capture and leverage interdependencies when learning from previous design evaluations.

The TPE optimization process begins with an initial phase of 100 randomly sampled trials, followed by a transition to probabilistic sampling guided by the expected improvement criterion. The Optuna library is used as the implementation framework [35].

The specific sampler configuration includes:

- `multivariate=True`: models interdependencies between variables
- `group=True`: groups variables by logical association
- `n_startup_trials=100`: initial exploration phase
- `n_ei_candidates=100`: number of candidate configurations evaluated per trial
- `seed=42`: fixed seed for reproducibility

The optimization process proceeds through the following steps:

1. Evaluate the four objective functions for a given configuration.
2. Check the feasibility of the configuration by ensuring no element exceeds the (UC ≤ 1) threshold.
3. Normalize and scalarize the results using the stakeholder-defined weight vector.
4. Record the best configuration based on the objective score.
5. Repeat for all configurations in the search space for a fixed number of trials (e.g., N_{trials} from 50 to 2000).

TPE has been implemented in two algorithmic variants:

- Algorithm 3: TPE Search for Mass Optimization
- Algorithm 4: TPE Search for Structural Complexity Objectives Optimization

These variants reflect whether the objective is purely scalarized or relates to one of the structural complexity metrics. When multiple configurations yield the same score - for example, in symmetry or beam continuity, a secondary filtering step based on mass is applied to select the lighter solution.

Algorithm 3 TPE Search for Mass Optimization

Require: $\mathcal{P}$, k, UC_{max}, f_M, N_{trials}, $\mathcal{W}$

```
Input: Profile set $\mathcal{P} = \{\text{HEA100}, \text{HEA120}, \text{HEA140}, \text{HEA180}, \text{HEA220}\}$, number of elements $k = 9$, maximum Unity Check $UC_{\text{max}} = 1$, mass objective function $f_M : \mathcal{S} \to \mathbb{R}$ (total mass in [kg]), number of trials $N_{\text{trials}}$, weight distributions $\mathcal{W} = \{w_M, w_{CN}, w_{SYM}, w_{BCN}\}$
Initialize $\mathcal{S} \leftarrow \{s \mid s : \{1, \ldots, k\} \to \mathcal{P}\}$          ▷ Search space of all configurations
Initialize $results \leftarrow \emptyset$          ▷ Store results for each $n$ and $w$
for $n \in \{1, 2, 3, 4, 5\}$ do          ▷ Iterate over number of unique HEA profile types
    $\mathcal{S}_n \leftarrow \{s \in \mathcal{S} \mid |\text{unique}(s)| = n\}$          ▷ Filter configurations with exactly $n$ profiles
    for each $w \in \mathcal{W}$ do          ▷ Iterate over weight distributions
        Initialize $study \leftarrow$ TPESampler($n_{\text{startup}} = 100$, seed $= 42$)
        Initialize $best\_config_n \leftarrow \emptyset$, $best\_M_n \leftarrow \infty$          ▷ Track best configurations for batch $n$
        for $t = 1$ to $N_{\text{trials}}$ do          ▷ Perform TPE trials
            $s \leftarrow$ sample_config($study, \mathcal{S}_n$)
            if $UC(s) \leq UC_{\text{max}}$ then          ▷ Check feasibility
                $M, CN, SYM, BCN \leftarrow f_{\text{all}}(s)$
                $score \leftarrow w_M \cdot \text{normalize}(M) + w_{CN} \cdot \text{normalize}(CN) + w_{SYM} \cdot \text{normalize}(SYM) + w_{BCN} \cdot \text{normalize}(BCN)$          ▷ Weighted score
                if $w_M = 1$ and $w_{CN} = w_{SYM} = w_{BCN} = 0$ then          ▷ Prioritize M
                    if $M < best\_value_n$ then          ▷ Update if lighter
                        $best\_config_n \leftarrow s$
                        $best\_value_n \leftarrow M$
                    end if
                end if
                update($study, s, (M, CN, SYM, BCN)$)          ▷ Update TPE model
            end if
        end for
        $results \leftarrow results \cup \{n, \mathcal{W}, best\_config_n, best\_value_n\}$          ▷ Store batch result
    end for
end for
return $results$          ▷ Return DataFrame with $n$, weights, configuration, and objective result
```

Algorithm 4 TPE Search for Structural Complexity Objectives Optimization

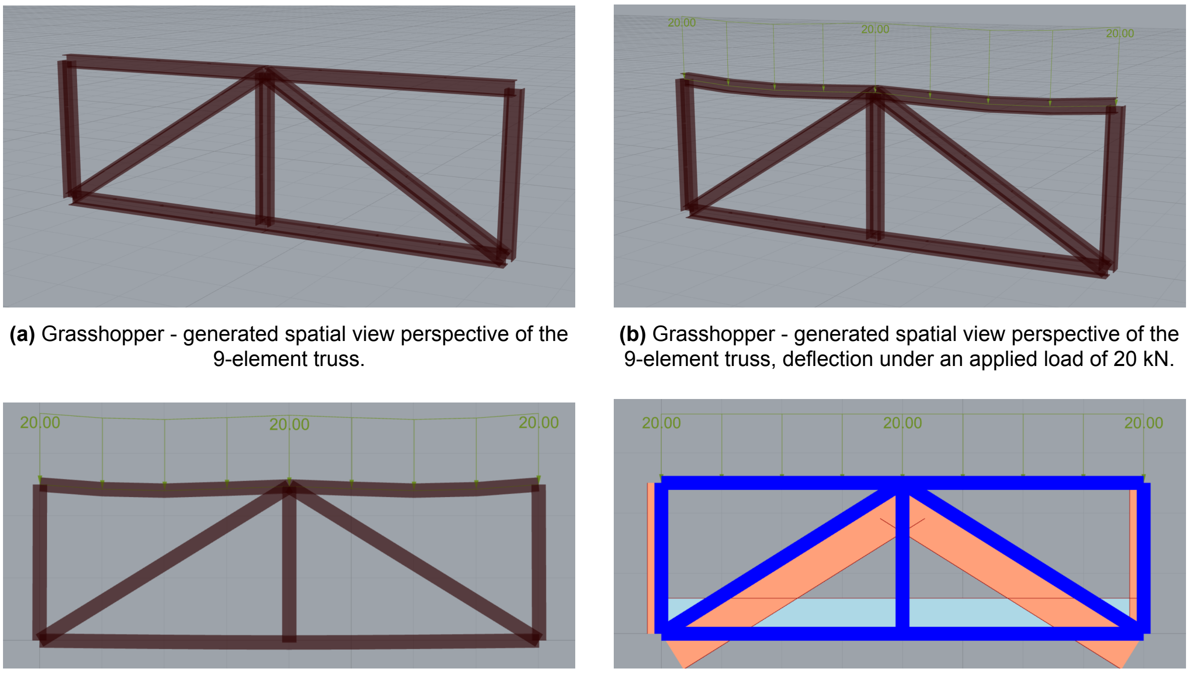
```
Require: $\mathcal{P}$, $k$, $UC_{\text{max}}$, $f$, $f_M$, $N_{\text{trials}}$, $\mathcal{W}$

Input: Profile set $\mathcal{P} = \{\text{HEA100}, \text{HEA120}, \text{HEA140}, \text{HEA180}, \text{HEA220}\}$, number of elements $k = 9$, maximum Unity Check $UC_{\text{max}} = 1$, objective function $f : \mathcal{S} \to \mathbb{R}$ (e.g., CN, SYM, BCN), mass objective function $f_M : \mathcal{S} \to \mathbb{R}$, number of trials $N_{\text{trials}}$, weight distributions $\mathcal{W} = \{w_M, w_{CN}, w_{SYM}, w_{BCN}\}$
Initialize $\mathcal{S} \leftarrow \{s \mid s : \{1, \ldots, k\} \to \mathcal{P}\}$    ▷ Search space of all configurations
Initialize $results \leftarrow \emptyset$    ▷ Store results for each $n$ and $w$
for $n \in \{1, 2, 3, 4, 5\}$ do    ▷ Iterate over number of unique HEA profile types
    $\mathcal{S}_n \leftarrow \{s \in \mathcal{S} \mid |\text{unique}(s)| = n\}$    ▷ Filter configurations with exactly $n$ profiles
    for each $w \in \mathcal{W}$ do    ▷ Iterate over weight distributions
        Initialize $study \leftarrow \text{TPESampler}(n_{\text{startup}} = 100, \text{seed} = 42)$
        Initialize $best\_configs_n \leftarrow \emptyset$, $best\_value_n \leftarrow \infty$    ▷ Track best configurations for batch $n$
        for $t = 1$ to $N_{\text{trials}}$ do    ▷ Perform TPE trials
            $s \leftarrow \text{sample\_config}(study, \mathcal{S}_n)$
            if $UC(s) \leq UC_{\text{max}}$ then    ▷ Check feasibility
                $M, CN, SYM, BCN \leftarrow f_{\text{all}}(s)$
                $score \leftarrow w_M \cdot \text{normalize}(M) + w_{CN} \cdot \text{normalize}(CN) + w_{SYM} \cdot \text{normalize}(SYM) + w_{BCN} \cdot \text{normalize}(BCN)$    ▷ Weighted score
                if $f = \mathit{CN}$ and $w_{CN} = 1$ and $w_M = w_{SYM} = w_{BCN} = 0$ then    ▷ Prioritize CN
                    if $CN < best\_value_n$ then    ▷ New best found
                        $best\_value_n \leftarrow CN$
                        $best\_configs_n \leftarrow \{s\}$    ▷ Reset list with new best
                    else if $CN = best\_value_n$ then    ▷ Equal score, add to list
                        $best\_configs_n \leftarrow best\_configs_n \cup \{s\}$
                    end if
                else if $f = \mathit{SYM}$ and $w_{SYM} = 1$ and $w_M = w_{CN} = w_{BCN} = 0$ then    ▷ Prioritize SYM
                    if $SYM < best\_value_n$ then    ▷ New best found
                        $best\_value_n \leftarrow SYM$
                        $best\_configs_n \leftarrow \{s\}$    ▷ Reset list with new best
                    else if $SYM = best\_value_n$ then    ▷ Equal score, add to list
                        $best\_configs_n \leftarrow best\_configs_n \cup \{s\}$
                    end if
                else if $f = \mathit{BCN}$ and $w_{BCN} = 1$ and $w_M = w_{CN} = w_{SYM} = 0$ then    ▷ Prioritize BCN
                    if $BCN < best\_value_n$ then    ▷ New best found
                        $best\_value_n \leftarrow BCN$
                        $best\_configs_n \leftarrow \{s\}$    ▷ Reset list with new best
                    else if $BCN = best\_value_n$ then    ▷ Equal score, add to list
                        $best\_configs_n \leftarrow best\_configs_n \cup \{s\}$
                    end if
                end if
                update$(study, s, (M, CN, SYM, BCN))$    ▷ Update TPE model
            end if
        end for
        $best\_config_n \leftarrow \arg\min_{s \in best\_configs_n} f_M(s)$    ▷ Secondary filtering by mass
        $results \leftarrow results \cup \{n, \mathcal{W}, best\_config_n, best\_value_n\}$    ▷ Store batch result
    end for
end for
return $results$    ▷ Return DataFrame with $n$, weights, configuration, and objective result
```

A key strength of the TPE algorithm lies in its capacity to handle diverse objective types, including continuous (e.g., mass) and discrete or topological metrics (e.g., connection degree, symmetry, beam continuity). It scales effectively to large design spaces and adapts to trade-off shifts driven by varying weight distributions. This adaptability makes TPE the preferred algorithm for the multi-objective optimization phase of the study.

5 Experimental setup and Validation

5.1. Introduction to experimental setup

This chapter presents computational experiments designed to evaluate best methods for standardised truss grouping within acceptable technological constraints. This paper primarily focusses on the performance of the model-based optimisation method, Tree-structured Parzen Estimator (TPE), compared to a standard exhaustive search baseline. The benchmark parameters are selected as a test case from a simplified rectangular truss including nine parts (see Figure A.5). This geometry demonstrates a balance between representativeness and computational efficiency by embodying typical configurations utilised in modular pedestrian and light vehicle bridges. In the Dutch context, it is occasionally termed a ”square truss” (vierkant vakwerk).



(a) Grasshopper - generated spatial view perspective of the 9-element truss.

(b) Grasshopper - generated spatial view perspective of the 9-element truss, deflection under an applied load of 20 kN.

(c) Grasshopper - generated front view perspective of the 9-element truss under a 20kN load, illustrating structural deflection.

(d) Grasshopper - generated front view perspective of the 9-element truss under a 20kN load, illustrating axial force distribution.

Figure 5.1: Benchmark problem Typology 1: Grasshopper - generated parametric truss with 9 elements, shown under a 20kN load with visualizations of deflection and axial force distribution from multiple perspectives.

Each optimisation algorithm is assessed based on its ability to identify appropriate truss configurations under various constraints. More precisely, where (n) is the number of distinct HEA profiles permitted in the design, TPE and exhaustive search are evaluated over five profile grouping constraints (n from 1 to 5). Four performance parameters are evaluated for optimisation performance: mass, connection degree, symmetry, and beam continuity.

The benchmark model, optimisation objectives, exhaustive search methodology, TPE application, validation results, and stakeholder-focused visualisation tools are elaborated upon in the following subchapters.

5.2. Benchmark model definition

The benchmark model employed in this study is a two-dimensional rectangular truss consisting of $(k = 9)$ elements. It was preferred for its equilibrium between geometric clarity, realistic loading behavior, and traceable optimization search space, providing it appropriate for both exhaustive and TPE-based search.

The geometry and structural simulation were generated using Rhino and Grasshopper combined, with structural performance analyzed with the Karamba3D plugin. The model is conceptualized as statically deterministic and planar, featuring pinned joints and adequate lateral bracing to negate bending effects. Consequently, the structure is evaluated based on the assumption of pure axial force, as permitted by [21] which allows for the exclusion of moment effects in trusses exhibiting optimal joint behavior and bracing.

Material and Load Assumptions:

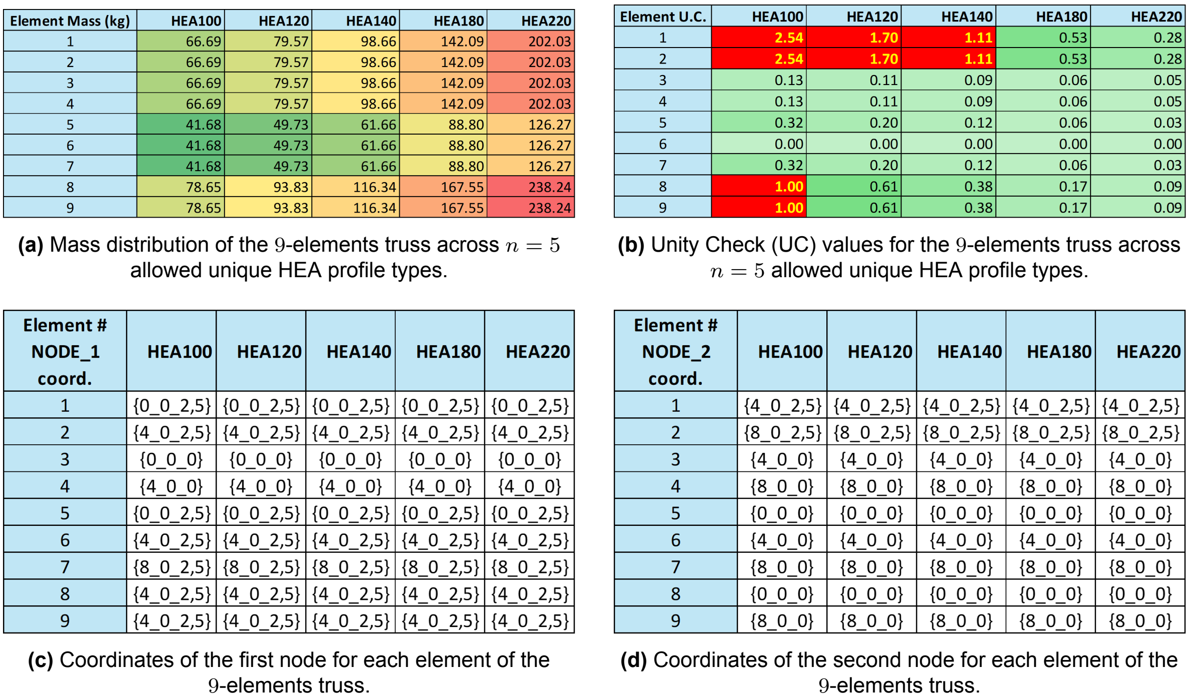
- Material: Frequently utilized in European practice, S355 structural steel has a Young's modulus of approximately 210 GPa.
- Profiles Set: HEA100, HEA120, HEA140, HEA180, HEA220 – selected based on structural relevance, fabrication availability, and prior relevance in research on modular systems.
- Loading: The truss is subjected to a uniformly distributed load of 20 kN/m, simulating service loads for pedestrian or cyclist bridges. This load level is a realistic abstraction, enabling consistent performance comparisons across profile types and grouping configurations.

Following the simulation phase, performance data including unity check (UC) values, mass, and nodal coordinates were extracted from Karamba3D and exported from Grasshopper into structured Excel files. These spreadsheets form the foundational dataset for the optimization framework.

The dataset supports four key information tables:

- Subfigure A.1a presents the mass values computed for each design configuration.
- Subfigure A.1b displays the (UC) values, highlighting the structural feasibility of each configuration.
- Subfigures A.1c and A.1d contain the coordinates of the first and second nodes of each truss element, respectively, defining the structural topology.

This structured data was essential for pre-processing and allowed the following key operations to be performed as a coherent preparatory step, rather than as disjointed processing tasks. Feasibility filtering was first applied to eliminate all design combinations where (UC > 1.0) for any element. This step significantly reduced the profile assignment space from 5^9 turning approximatelly 1.95 million configurations to a tractable subset of 200,000 valid solutions. Next, the cleaned dataset was used to construct the complete design space of all feasible combinations, where each of the $(k = 9)$ truss elements could be assigned one of five HEA profiles. The number of allowed distinct profiles (n) ranged from 1 (fully standardized) to 5 (fully unique), and each combination had to satisfy feasibility constraints based on the (UC).



Element Mass (kg)	HEA100	HEA120	HEA140	HEA180	HEA220
1	66.69	79.57	98.66	142.09	202.03
2	66.69	79.57	98.66	142.09	202.03
3	66.69	79.57	98.66	142.09	202.03
4	66.69	79.57	98.66	142.09	202.03
5	41.68	49.73	61.66	88.80	126.27
6	41.68	49.73	61.66	88.80	126.27
7	41.68	49.73	61.66	88.80	126.27
8	78.65	93.83	116.34	167.55	238.24
9	78.65	93.83	116.34	167.55	238.24

(a) Mass distribution of the 9-elements truss across $n = 5$ allowed unique HEA profile types.

Element U.C.	HEA100	HEA120	HEA140	HEA180	HEA220
1	2.54	1.70	1.11	0.53	0.28
2	2.54	1.70	1.11	0.53	0.28
3	0.13	0.11	0.09	0.06	0.05
4	0.13	0.11	0.09	0.06	0.05
5	0.32	0.20	0.12	0.06	0.03
6	0.00	0.00	0.00	0.00	0.00
7	0.32	0.20	0.12	0.06	0.03
8	1.00	0.61	0.38	0.17	0.09
9	1.00	0.61	0.38	0.17	0.09

(b) Unity Check (UC) values for the 9-elements truss across $n = 5$ allowed unique HEA profile types.

Element # NODE_1 coord.	HEA100	HEA120	HEA140	HEA180	HEA220
1	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}
2	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}
3	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}
4	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}
5	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}
6	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}
7	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}
8	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}
9	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}

(c) Coordinates of the first node for each element of the 9-elements truss.

Element # NODE_2 coord.	HEA100	HEA120	HEA140	HEA180	HEA220
1	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}
2	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}
3	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}
4	{8_0_0}	{8_0_0}	{8_0_0}	{8_0_0}	{8_0_0}
5	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}
6	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}
7	{8_0_0}	{8_0_0}	{8_0_0}	{8_0_0}	{8_0_0}
8	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}
9	{8_0_0}	{8_0_0}	{8_0_0}	{8_0_0}	{8_0_0}

(d) Coordinates of the second node for each element of the 9-elements truss.

Figure 5.2: Excel tables for a parametric truss with 9-elements and $n = 5$ allowed unique HEA profile types.

5.3. Objective functions and metrics

Targeting the optimisation procedure in this thesis reflects four key objective functions, each indicating a core engineering trade-off between material efficiency, constructability, and structural simplicity. Formulated for minimisation, the four metrics discussed in this thesis, namely — mass, connection degree, symmetry, and beam continuity — are evaluated across different truss typologies presented with all required data in Appendix A. These performance objectives were selected not only for their quantifiability but also for their alignment with the most common stakeholder requirements, particularly regarding structural performance and fabrication feasibility.

In the context of this study, these objectives form the basis of both the single and multi-objective optimization tasks described in subsequent sections. Their mathematical formulation reflects a recurring trade-off between standardization (low values for (n) implying reduced complexity, simpler joints) and uniqueness (high values for (n) implying tailored structural performance, better calibrated (UC) ratios). The challenge of balancing these metrics becomes especially evident in the multi-objective TPE discussed in Chapter 6.

5.4. Exhaustive search baseline

An exhaustive search strategy was built to serve as a baseline reference for evaluating the performance of the TPE optimisation method. This method systematically assesses all potential profile assignment combinations throughout the truss elements.

For each value of (n) (i.e., the number of unique profiles allowed (n from 1 to 5), the best-performing configuration was selected for each objective individually. The optimal configuration was defined as the one that minimized the target metric, such as mass, connection degree, symmetry, or beam continuity. In scenarios when multiple configurations achieved the same scores for the target objective, an additional filter based on mass was used, resulting in a selection of the lightest feasible configuration. It is essential to understand that secondary filtering is unfeasible in the TPE configuration due to its probabilistic characteristics and sampling-based framework, underscoring a critical differentiation examined in the next section.

This approach allows the exhaustive search to serve as a reliable performance baseline, especially in the single-objective setting. The mass objective, for example, yielded an optimal value of 730.27 kg when ($n = 3$), demonstrating that the use of exactly three distinct profile types provided the best trade-off between structural adequacy (UC ≤ 1) and material efficiency. Detailed numerical results of this analysis are presented in the corresponding Excel tables in Appendix A and plots in Chapter 6.

Despite its simplicity and completeness, the scalability of exhaustive search is significantly constrained, as evidenced in this study. As the quantity of elements (k) with available profiles (n) increases, the potential configurations grow exponentially, following the exponential relation (n^k). With ($n = 5$) and ($k = 9$), this results in approximately 1.95 million combinations. The quick expansion makes exhaustive search enumeration computationally unfeasible for bigger truss models, as further examined in Chapter 6. With an increase in the number of elements to ($k = 17$), the design space rises to 5^{17}, or 7.6 billion configurations as shown in Figure 6.10b. Although with the sorting, such a space would need days or even weeks of computation, leaving the approach impractical for applications beyond benchmark problems. This necessitates the implementation of alternative approaches such as TPE, which can attain near-optimal solutions with significantly fewer iterations. Detailed computationl framework outputs validating TPE convergence against the exhaustive search baseline for these objectives are provided in Appendix D.

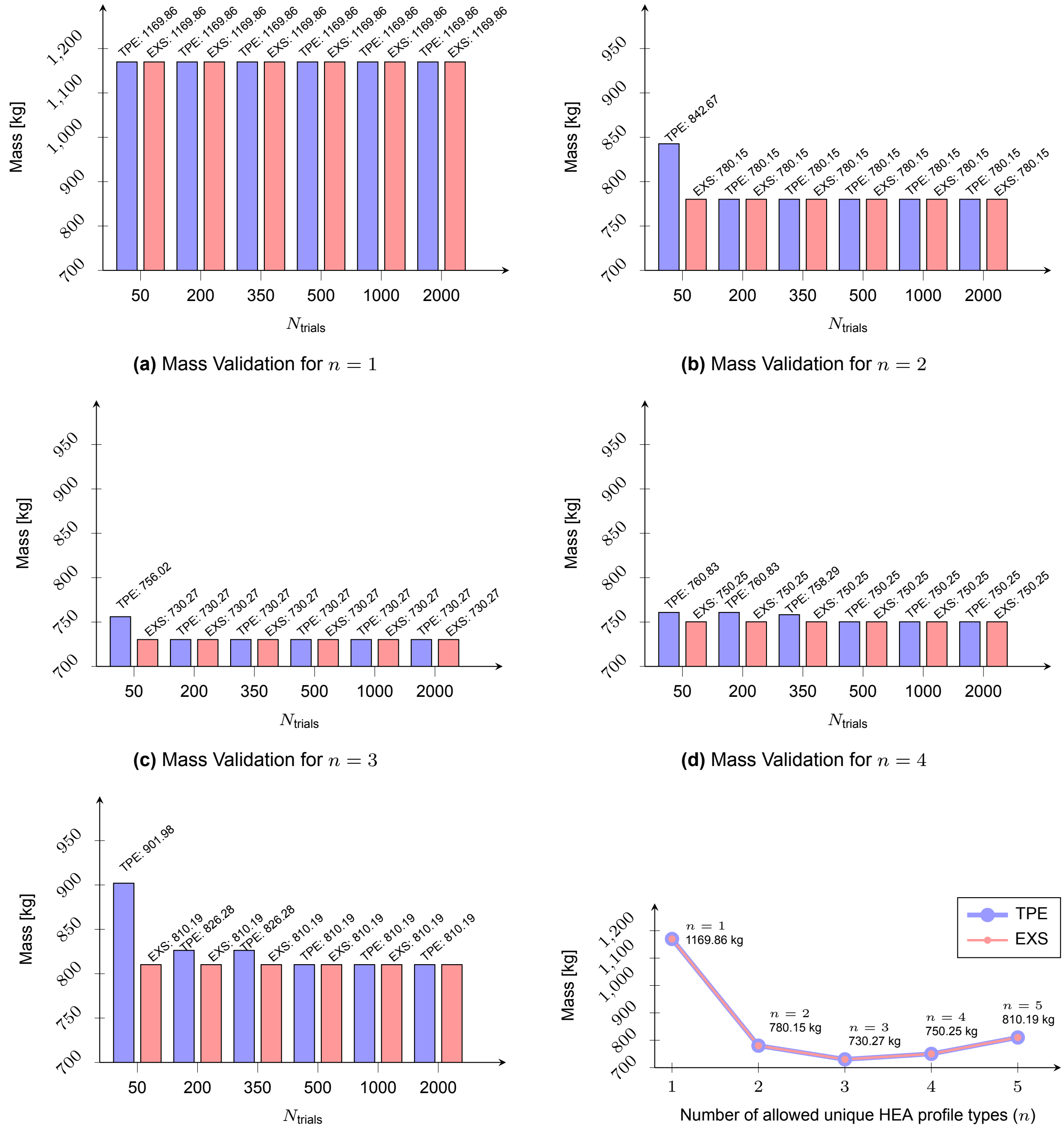
5.5. Validation results: TPE vs Exhaustive Search

This section presents a comparative analysis between the Tree-structured Parzen Estimator (TPE) and exhaustive search, focusing on convergence behavior, computational efficiency, and solution quality for the 9-element truss benchmark. The validation was conducted for each of the four objectives: mass, connection degree, symmetry, and beam continuity, with TPE configured for single-objective optimization in each case.

Exhaustive search served as the ground-truth baseline, identifying the optimal solution out of approximately 1.95 million configurations. However, this space was pre-filtered by eliminating all designs where any member's (UC) exceeded threshold, reducing the valid design set to approximately 200,000 combinations. For each value of n (i.e., number of unique profiles, with (n) equal from 1 to 5), the best-scoring design was selected per objective. When multiple configurations scored equally, a secondary filtering process was applied to minimize mass. This second-level filtering was only possible under exhaustive search and is not feasible in the TPE framework (as follows from Algorithms 1 & 2 from Section 4.5.1 and Algorithms 3 & 4 from Section 4.5.2), highlighting a methodological contrast between the two approaches In terms of post-processing flexibility, which may significantly affect stakeholder usability and fairness in algorithmic comparison particularly in scenarios involving symmetry, modularity, or practical constructability concerns.

Mass Objective: convergence comparison

Figure 5.3 presents the convergence behavior for the mass objective, plotting the objective value against the number of trials (N_{trials}). The TPE method demonstrates reliable convergence toward the exhaustive optimum as the number of trials increases. Early trials (50 – 200) exhibit high variance, however convergence becomes stable and near optimal from 500 trials. Notably, the most complex cases specifically for ($n = 3$) and ($n = 4$) required approximately 350 trials to fully converge to results comparable with the exhaustive search, indicating a higher combinatorial difficulty at these intermediate values of (n). This behavior establishes TPE as a computationally viable alternative when exhaustive search is infeasible, though convergence speed is strongly influenced by problem dimensionality.



(a) Mass Validation for $n = 1$

(b) Mass Validation for $n = 2$

(c) Mass Validation for $n = 3$

(d) Mass Validation for $n = 4$

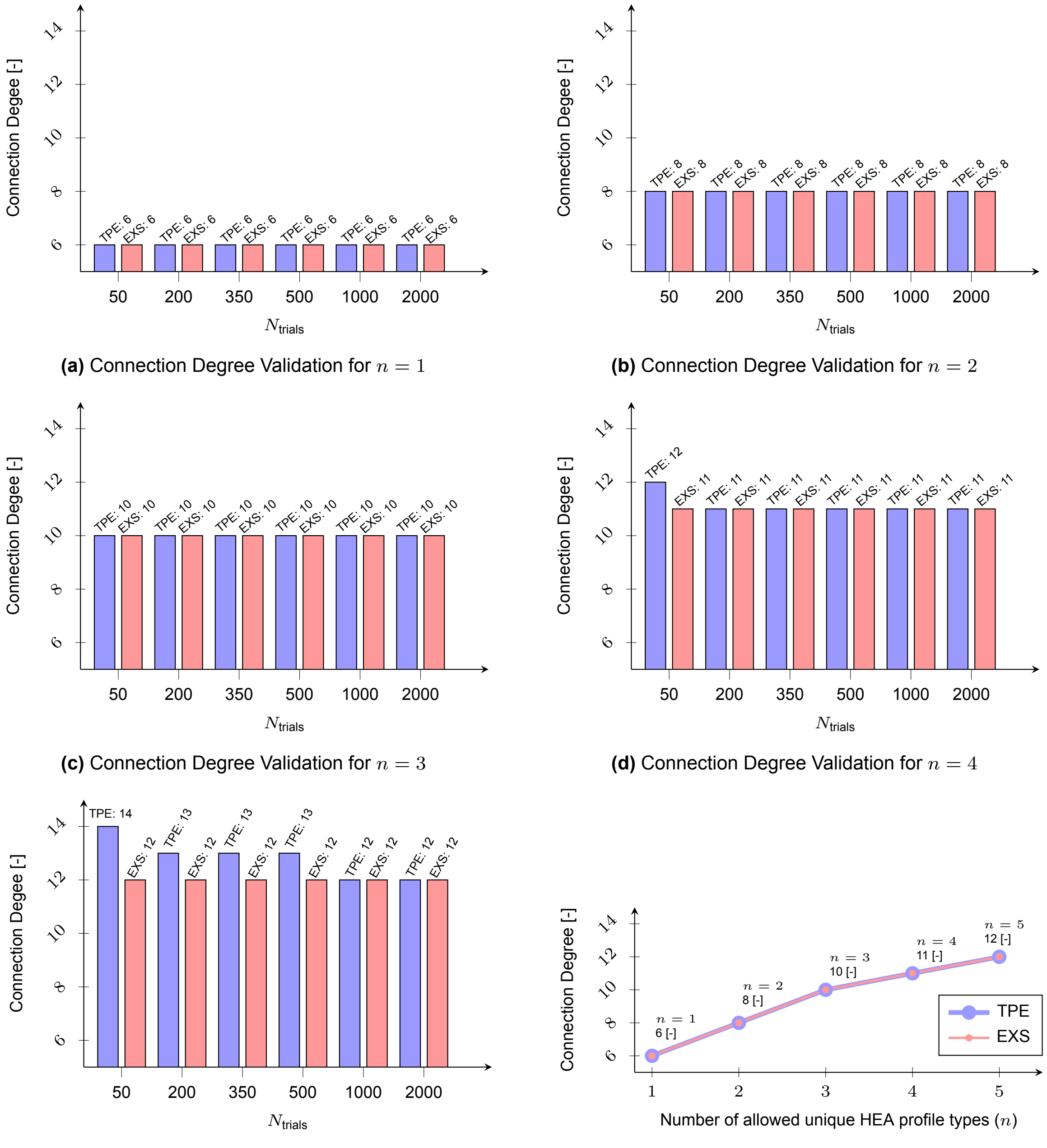
(e) Mass Validation for $n = 5$

(f) TPE and EXS convergence for $N_{\text{trials}} = 500$

Figure 5.3: Validation of the Tree-structured Parzen Estimator (TPE) through convergence of Mass to Exhaustive Search (EXS) results for varying N_{trials} with allowed unique HEA profile types (n).

Connection Degree: convergence comparison

Figure 5.4 illustrate the convergence patterns for the connection degree and beam continuity objectives, both of which exhibit significant variability at lower trial counts. These objectives are particularly sensitive to topological regularity, making early convergence challenging for the probabilistic TPE approach. While some configurations achieved near-optimal results with as few as 50 trials, especially for lower values of (n) , the final and most difficult converging case $(n = 5)$ required up to 500 trials to align with the exhaustive search results. This underscores the uneven convergence behavior across different profile counts and reinforces the necessity of adequate trial budgets to ensure robustness in TPE-based optimization, particularly for structurally intricate objectives.



(a) Connection Degree Validation for $n = 1$

(b) Connection Degree Validation for $n = 2$

(c) Connection Degree Validation for $n = 3$

(d) Connection Degree Validation for $n = 4$

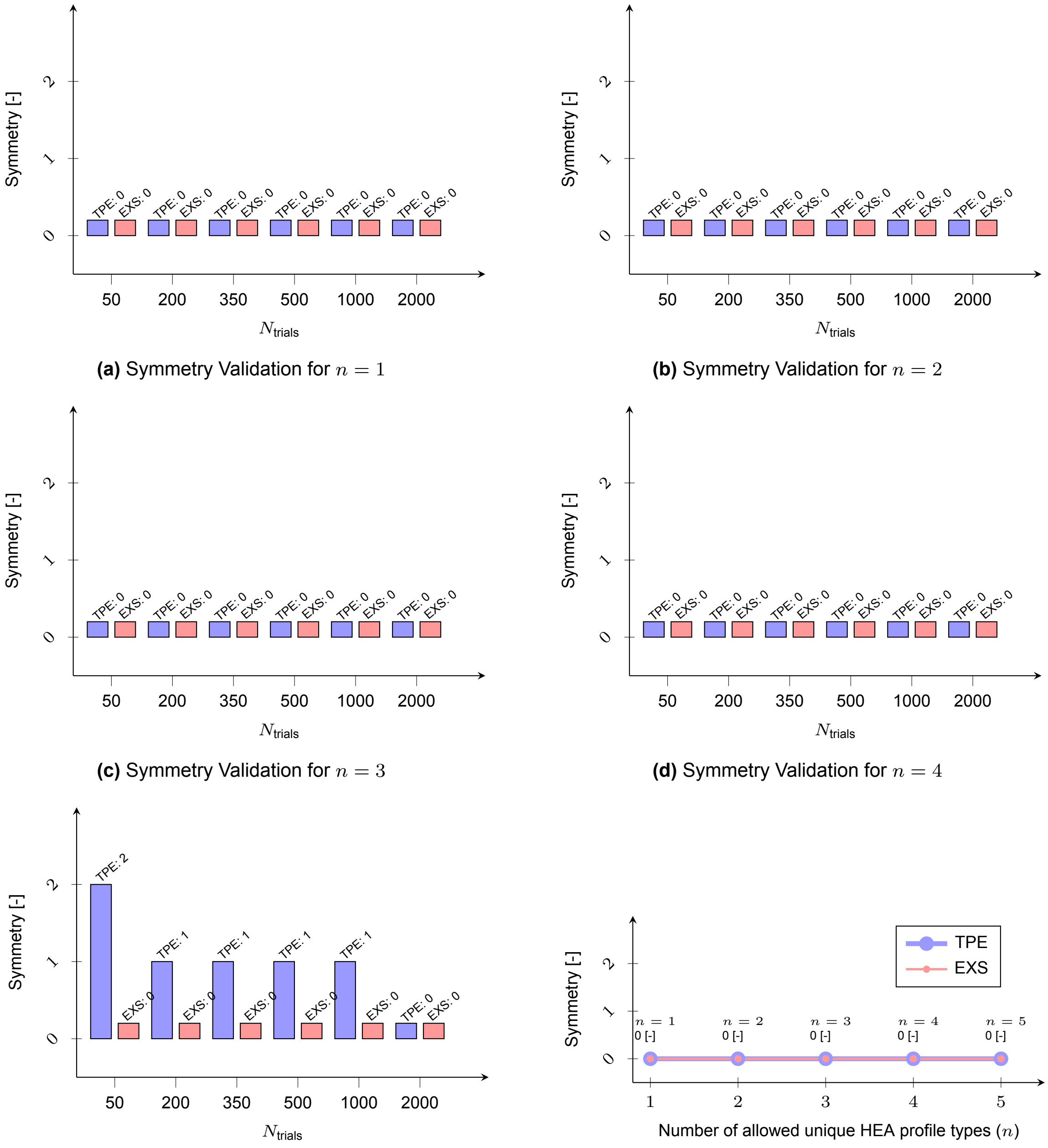
(e) Connection Degree Validation for $n = 5$

(f) TPE and EXS convergence for $N_{\text{trials}} = 1000$

Figure 5.4: Validation of the Tree-structured Parzen Estimator (TPE) through convergence of Connection Degree to Exhaustive Search (EXS) results for varying N_{trials} with allowed unique HEA profile types (n).

Symmetry: convergence comparison

Figure 5.5 focuses on the symmetry objective, which exhibited the most extended convergence horizon among all four objectives. While values of $(n = 1)$ through $(n = 4)$ converged relatively early in the optimization process, the most challenging case $(n = 5)$ required the full 2000 trials to achieve convergence comparable to the exhaustive search. This marks the highest number of trials needed across all objectives, highlighting both the complexity of preserving symmetry in the design space and the sensitivity of this objective to probabilistic search dynamics. The findings emphasize that convergence for symmetry is not only objective-dependent but also highly influenced by the dimensionality and structural constraints associated with higher values of (n).



(a) Symmetry Validation for $n = 1$

(b) Symmetry Validation for $n = 2$

(c) Symmetry Validation for $n = 3$

(d) Symmetry Validation for $n = 4$

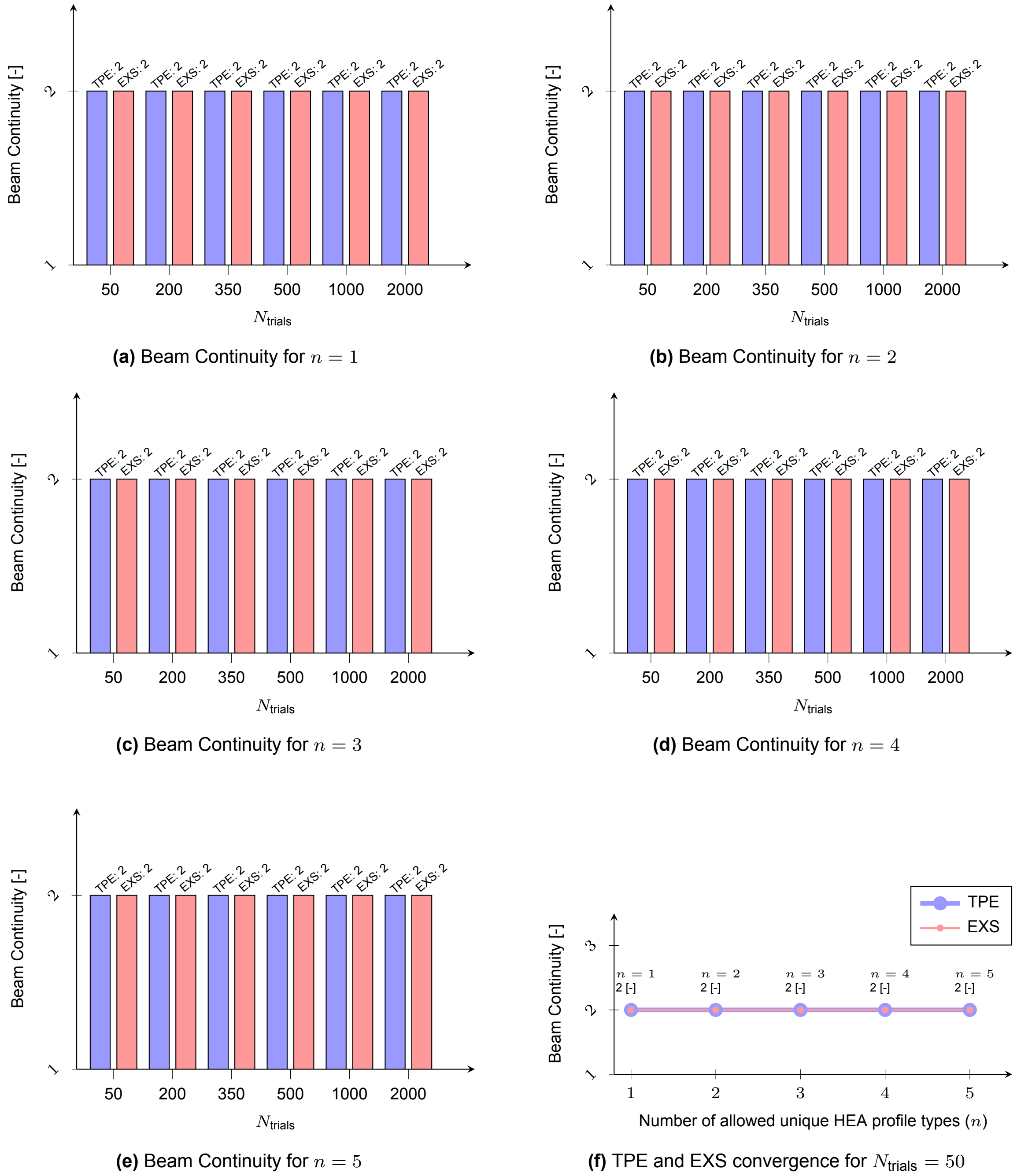
(e) Symmetry Validation for $n = 5$

(f) TPE and EXS convergence for $N_{\text{trials}} = 2000$

Figure 5.5: Validation of the Tree-structured Parzen Estimator (TPE) through convergence of Symmetry to Exhaustive Search (EXS) results for varying N_{trials} with allowed unique HEA profile types (n).

Beam continuity: convergence comparison

Figure 5.6 presents the convergence pattern for the beam continuity objective, which demonstrated the fastest and most consistent convergence across all evaluated cases. Remarkably, all values of (n from 1 to 5) reached convergence within the first 50 trials, making this the least demanding objective in terms of computational effort. This result suggests that the solution space for beam continuity is either less complex or more readily navigable by the TPE algorithm, allowing optimal configurations to be identified rapidly. The robustness and low variance observed at early stages further underscore the efficiency of TPE when applied to objectives with more localized or less ambiguous structural constraints.



(a) Beam Continuity for $n = 1$

(b) Beam Continuity for $n = 2$

(c) Beam Continuity for $n = 3$

(d) Beam Continuity for $n = 4$

(e) Beam Continuity for $n = 5$

(f) TPE and EXS convergence for $N_{\text{trials}} = 50$

Figure 5.6: Validation of the Tree-structured Parzen Estimator (TPE) through convergence of Beam Continuity to Exhaustive Search (EXS) results for varying N_{trials} with allowed unique HEA profile types (n).

Convergence behavior across objectives

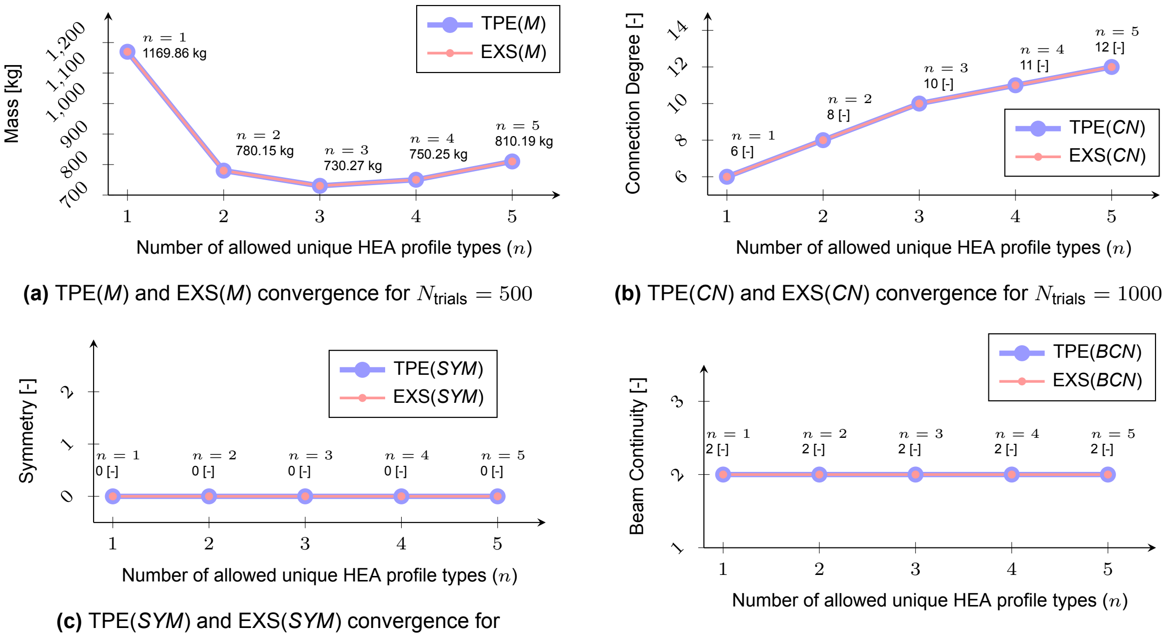
The convergence analysis highlights distinct optimization challenges and trial needs across four objectives. *BCN* proved most tractable, with all (n) converging consistently at just 50 trials, indicating a relatively simple or well-behaved solution space. *CN*, with most (n) converging early, but $(n = 5)$ needing up to 500 trials to match the exhaustive baseline. *M* displayed moderate complexity, with convergence typically stabilizing between 1000 and 2000 trials, though $(n = 3)$ and $(n = 4)$ extended this to around 350 trials. The most demanding was the *SYM* objective, where with $(n = 5)$ requiring 2000 trials, the highest observed. These differences underscore the importance of objective-specific and dimension-aware trial budgeting in probabilistic optimization like TPE, as convergence behavior is heavily depends on both the nature of the objective and the structural complexity from increasing (n).

6 Results and discussion

This chapter presents and discusses the experimental results of the proposed optimization algorithms, starting with Exhaustive Search (EXS) and continuing with the Tree-structured Parzen Estimator (TPE), based on the benchmark problem of a 9-element, structurally determinate 2D rectangular steel truss as detailed in Chapter 5. The evaluation focuses on balancing four performance objectives: structural mass, symmetry, connection degree, and beam continuity, as defined in Chapter 4. Section 6.1 highlights the performance of the TPE optimization algorithm compared to Exhaustive Search for the optimization of a single objective. The chapter then explores TPE performance in single versus multi-objective weight distribution scenarios: TPE($\mathcal{W}_1$) with uniformly distributed weights among all four objectives (6.2.1), TPE($\mathcal{W}_2$) prioritizing mass and symmetry (6.2.2), TPE($\mathcal{W}_3$) prioritizing connection degree and beam continuity (6.2.3), and TPE($\mathcal{W}_4$) heavily prioritizing symmetry and mass (6.2.3). Further analysis investigates the impact of the number of trials, the number of allowed HEA profiles (n), and weight distributions on the computational time and the accuracy of the findings (6.3). Next, the chapter emphasizes the application of multi-parallel plots to identify and visualize different truss configurations based on specific requirements (6.4). Finally, the chapter assesses the structural outcomes and their implications for stakeholder-driven design (6.5).

6.1. Comparison: TPE vs Exhaustive search single objective

Building on the experimental framework outlined in Chapter 5, this section evaluates the performance of the Tree-structured Parzen Estimator (TPE) and Exhaustive Search (EXS) algorithms in optimizing a single objective, mass (*M*), symmetry (*SYM*), connection degree (*CN*), or beam continuity (*BCN*) for a 9-element steel truss benchmark. Result illustrated in Figure 6.1, demonstrate convergence behavior and configuration differences across the objectives. As established in Chapter 5, TPE achieves convergence at ($N_{\text{trials}} = 2000$) for all (n from 1 to 5), aligning with EXS on primary objective values. Despite this alignment, the algorithms' distinct mechanisms yield variations in truss configurations, particularly in secondary objectives, which are analyzed in the subsections below.



(a) TPE(*M*) and EXS(*M*) convergence for $N_{\text{trials}} = 500$

(b) TPE(*CN*) and EXS(*CN*) convergence for $N_{\text{trials}} = 1000$

(c) TPE(*SYM*) and EXS(*SYM*) convergence for $N_{\text{trials}} = 2000$

(d) TPE(*BCN*) and EXS(*BCN*) convergence for $N_{\text{trials}} = 50$

Figure 6.1: Comparison of TPE and EXS single Optimization across Objectives.

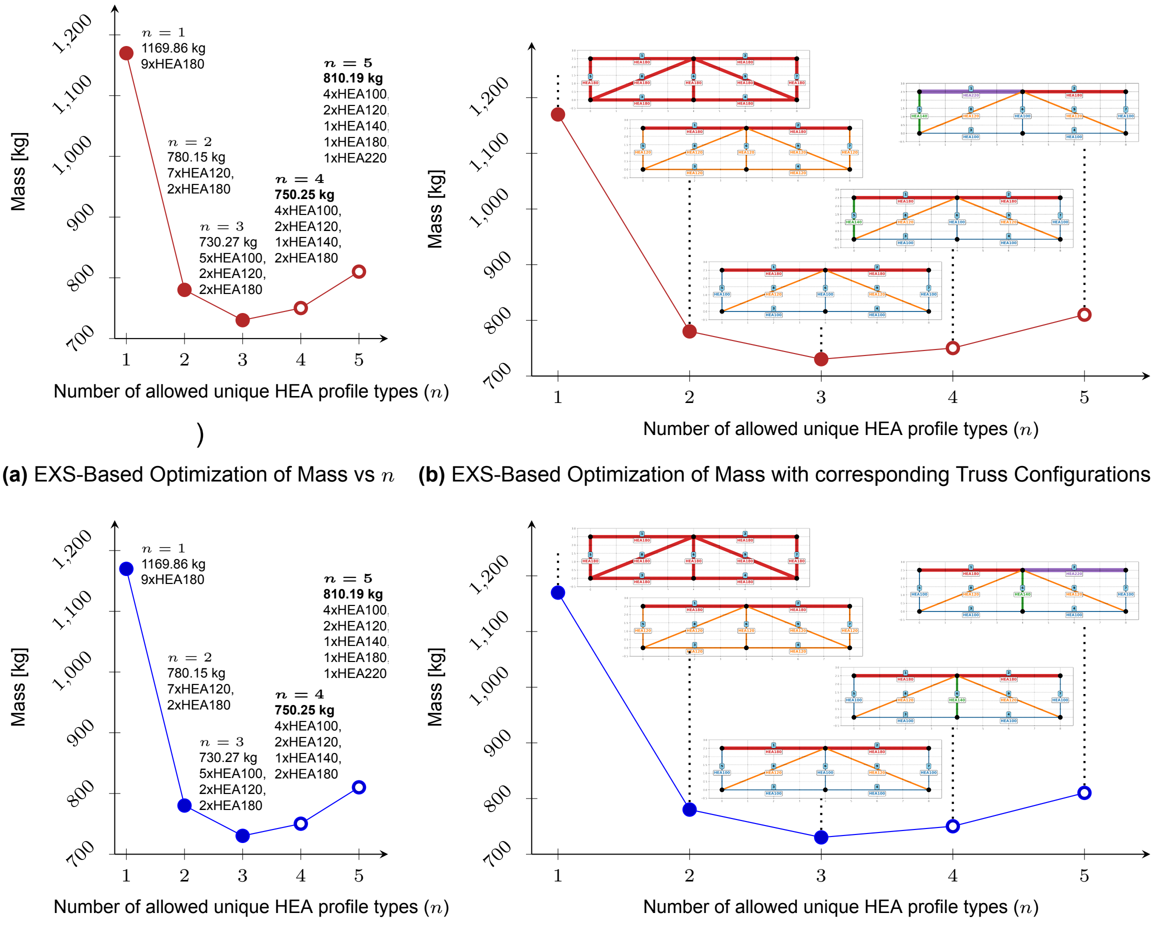
6.1.1. Comparison: TPE(M) vs EXS(M)

For mass optimization, TPE(*M*) and EXS(*M*) achieve identical minimal mass values, as reported in Table 6.1. However, with optimization focused solely on mass (secondary objectives assigned zero weight), the resulting truss configurations diverge, impacting metrics such as symmetry. For ($n = 4$), TPE(*M*) yields a symmetry score of 0[-] (fully symmetric), while EXS(*M*) produces a score of 1[-]. Similarly, at ($n = 5$), TPE(*M*) achieves a symmetry score of 1[-], compared to EXS(*M*) symmetry score of 2[-]. These differences, illustrated in Figiure 6.2 stem from the algorithms' selection strategies. EXS(*M*) evaluates all configurations systematically, selecting the first with minimal mass, often overlooking secondary objectives. Conversely, TPE(*M*) Bayesian approach, utilizing stochastic sampling guided by expected improvement, explores the design space more extensively, occasionally identifying configurations with enhanced secondary performance [7].

Table 6.1: Evaluation of Exhaustive Search EXS(*M*) vs Tree-structured Parzen Estimator TPE(*M*) for Mass Optimization across grouping levels $n = 1$ to 5, $N_{\text{trials}} = 2000$.

		$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	$\mathcal{W}$	**EXS(*M*)**	**TPE(*M*)**	Δ	**EXS(*M*)**	**TPE(*M*)**	Δ	**EXS(*M*)**	**TPE(*M*)**	Δ	**EXS(*M*)**	**TPE(*M*)**	Δ	**EXS(*M*)**	**TPE(*M*)**	Δ
Mass	**1**	**1169.86**	**1169.86**	**=**	**780.15**	**780.15**	**=**	**730.27**	**730.27**	**=**	**750.25**	**750.25**	**=**	**810.19**	**810.19**	**=**
Connection Degree	0	6	6	=	9	9	=	12	12	=	13	13	=	14	14	=
Symmetry	0	0	0	=	0	0	=	0	0	=	1	0	↓	2	1	↓
Beam Continuity	0	2	2	=	2	2	=	2	2	=	2	2	=	3	3	=

The divergence arises from EXS(*M*) streamlined process, which terminates upon identifying the minimal mass, as detailed in its pseudo-code (Chapter 4). For mass optimization, EXS(*M*) employs a straightforward iterative process, stopping at the first configuration achieving the optimal mass. In contrast, for structural complexity objectives like symmetry, EXS(*SYM*) incorporates an additional iterative step. For example, when multiple configurations share the same symmetry score but differ in mass (e.g., using light versus heavy HEA profiles), a secondary iteration selects the lightest configuration. This additional layer is absent in EXS(*M*), limiting its exploration and causing it to miss symmetric configurations that TPE(*M*) captures, a limitation evident in the results. This suggests TPE(*M*) potential to balance primary and secondary objectives, even in single-objective scenarios.



(a) EXS-Based Optimization of Mass vs n **(b)** EXS-Based Optimization of Mass with corresponding Truss Configurations

(c) TPE-Based Optimization of Mass vs n **(d)** TPE-Based Optimization of Mass with corresponding Truss Configurations

Figure 6.2: Comparison of Exhaustive Search (EXS) and Tree-structured Parzen Estimator (TPE) based optimization of mass versus the number of allowed unique HEA profile types (n) for $N_{\text{trials}} = 2000$.

6.1.2. Comparison: TPE(CN) vs EXS(CN)

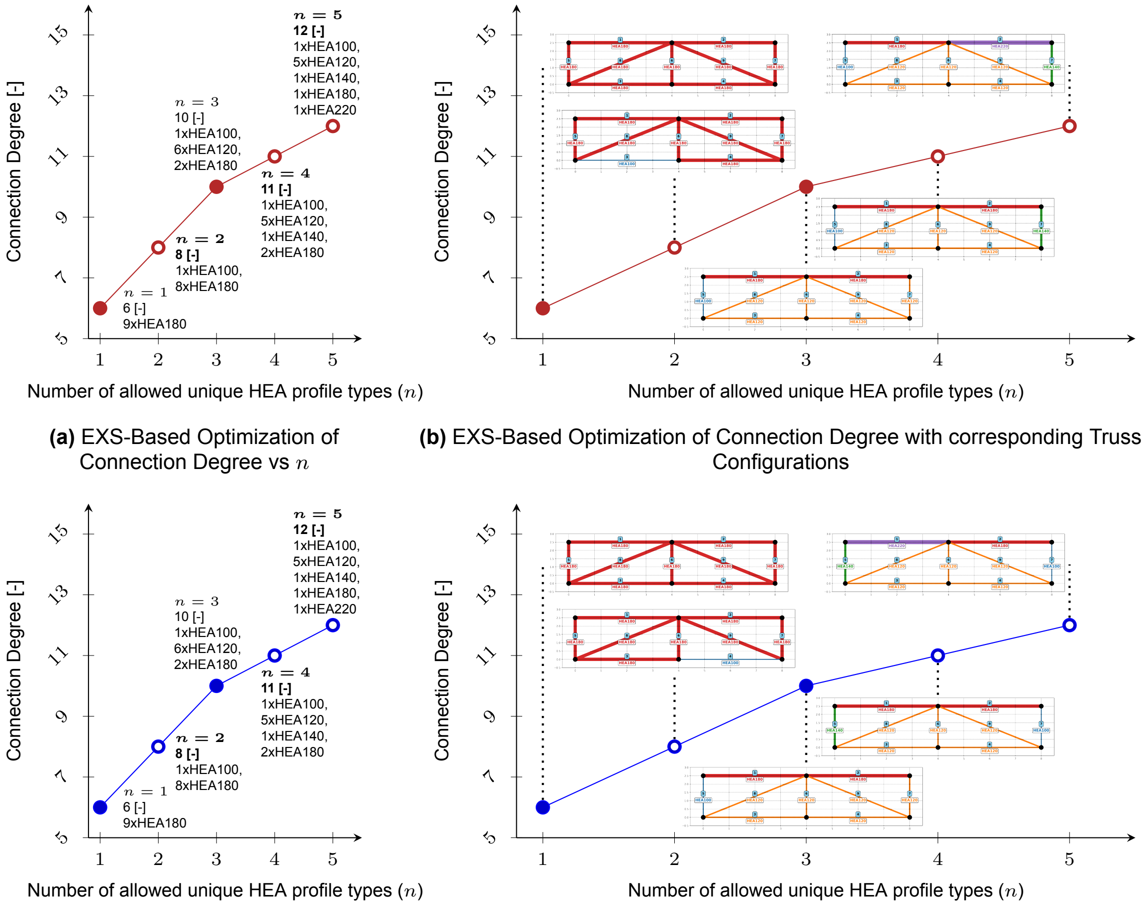
Connection degree optimization reveals a notable case where TPE(*CN*) and EXS(*CN*) converge on the primary objective and achieve identical secondary objective values (Table 6.2). This consistency, initially surprising, results from the constrained search space of the connection degree objective, particularly at higher (n), as detailed in Chapter 4. Defined by the number of distinct element types connected at a node, this objective significantly limits configuration variability [9] [47]. For ($n = 1$), where a single HEA profile is assigned to all nine elements, both TPE(*CN*) and EXS(*CN*) achieve a connection degree score of 6[-] (assuming a single profile type across nodes). Theoretically, TPE(*CN*) could select heavier profiles (e.g., HEA220 instead of HEA180) for all elements, maintaining the same connection degree score while increasing mass, since mass has zero weight in the weighted sum objective function. However, TPE(*CN*) consistently aligns with EXS(*CN*), producing identical mass and other secondary objective values.

This alignment is attributed to two key factors. First, the connection degree metric, being arrangement-driven, significantly reduces the number of feasible configurations, especially at higher (n) values. For ($n = 1$), where a single profile type is used, all elements must share the same HEA profile (e.g., HEA180), inherently fixing the mass and other secondary objectives, forcing TPE(*CN*) and EXS(*CN*) to select identical configurations. Second, TPE(*CN*) sampling strategy, as a Bayesian optimization method, models the probability distribution of high-performing configurations using a tree-structured Parzen estimator [8]. When ($w_M = 0$), TPE(*CN*) may still favor configurations that align with the initial or most frequently sampled solutions, which, in this case, resemble EXS(*CN*) outcomes due to the constrained search space. For higher values of (n), the search space remains limited, further enforcing consistency. Figure 6.3 illustrates these configurations, showing no differences in configurations identified by TPE(*CN*) and EXS(*CN*). This highlights the influence of objective-specific constraints on optimization, suggesting that connection degree's limited search space minimizes secondary objective divergence, even when weights permit flexibility.

Table 6.2: Evaluation of Exhaustive Search EXS(*CN*) vs Tree-structured Parzen Estimator TPE(*CN*) for Connection Degree Optimization across grouping levels $n = 1$ to 5, $N_{\text{trials}} = 2000$.

		$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w	**EXS(*CN*)**	**TPE(*CN*)**	Δ	**EXS(*CN*)**	**TPE(*CN*)**	Δ	**EXS(*CN*)**	**TPE(*CN*)**	Δ	**EXS(*CN*)**	**TPE(*CN*)**	Δ	**EXS(*CN*)**	**TPE(*CN*)**	Δ
Mass	0	1169.86	1169.86	=	1094.46	1094.46	=	772.11	772.11	=	784.04	784.04	=	843.98	843.98	=
Connection Degree	**1**	**6**	**6**	**=**	**8**	**8**	**=**	**10**	**10**	**=**	**11**	**11**	**=**	**12**	**12**	**=**
Symmetry	0	0	0	=	1	1	=	1	1	=	1	1	=	2	2	=
Beam Continuity	0	2	2	=	3	3	=	2	2	=	2	2	=	3	3	=

For higher values of (n), the search space remains limited, further enforcing consistency. Figure 6.3 illustrates these configurations, showing no differences in configurations identified by TPE(*CN*) and EXS(*CN*). This highlights the influence of objective-specific constraints on optimization, suggesting that connection degree's limited search space minimizes secondary objective divergence, even when weights permit flexibility.



(a) EXS-Based Optimization of Connection Degree vs n

(b) EXS-Based Optimization of Connection Degree with corresponding Truss Configurations

(c) TPE-Based Optimization of Connection Degree vs n

(d) TPE-Based Optimization of Connection Degree with corresponding Truss Configurations

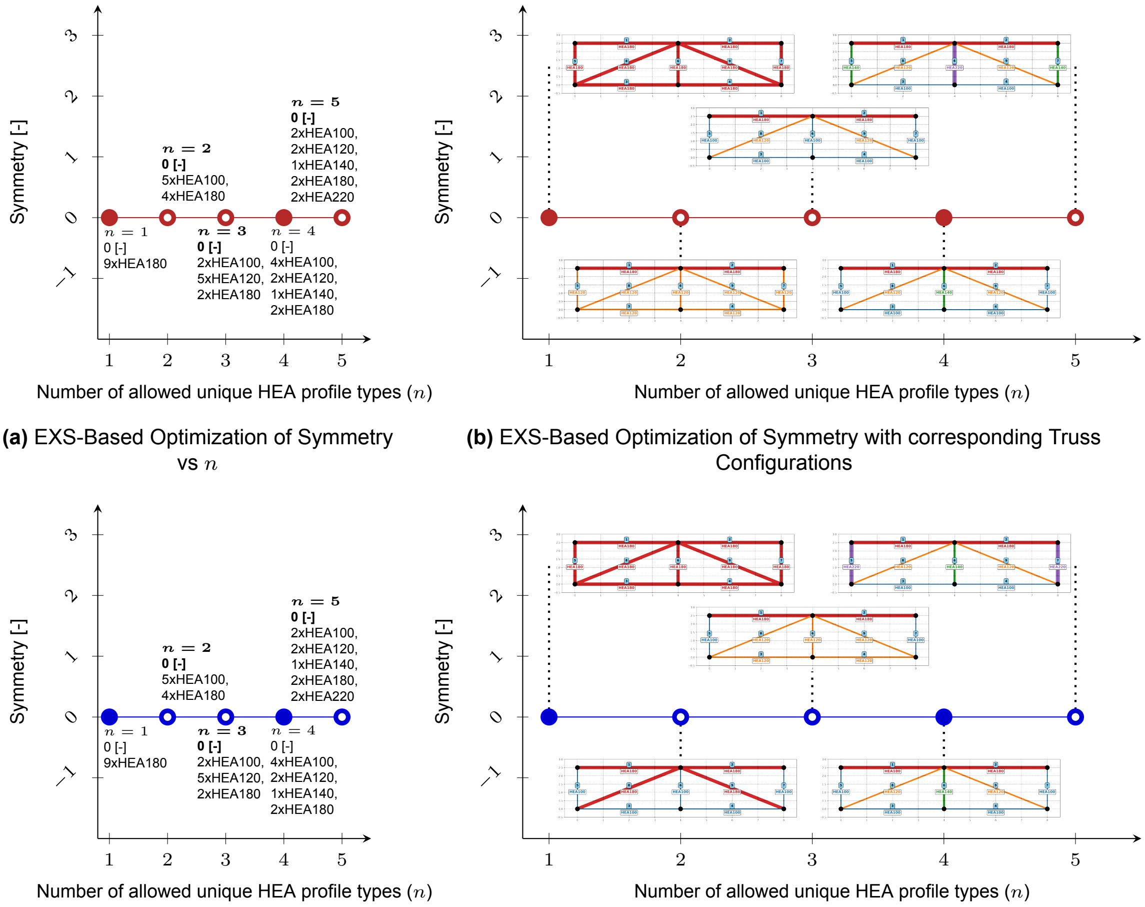
Figure 6.3: Comparison of Exhaustive Search (EXS) and Tree-structured Parzen Estimator (TPE) based optimization of connection degree versus the number of allowed unique HEA profile types (n) for $N_{\text{trials}} = 2000$.

6.1.3. Comparison: TPE(SYM) vs EXS(SYM)

In symmetry optimization, TPE(*SYM*) converges with EXS(*SYM*) on the primary objective (Table 6.3), yet disparities emerge in secondary objectives such as mass and connection degree. For ($n = 2$), TPE(*SYM*) yields configurations with higher mass and connection degree than EXS(*SYM*), due to its exclusive focus on symmetry (other objectives weighted at zero). This prioritization stems from TPE(*SYM*) weighted sum approach, where zero-weighted objectives are disregarded, leading to configurations optimized solely for symmetry. Slightly increasing weights for secondary objectives could align TPE(*SYM*) outcomes with EXS(*SYM*), suggesting a potential for multi-objective optimization, as explored in later sections. Figure 6.4 illustrates these differences for ($n = 2$), ($n = 3$), and ($n = 5$), confirming convergence on symmetry but highlighting varied secondary performance. These findings underscore TPE(*SYM*) flexibility in single-objective optimization, potentially enhancing secondary metrics when weights are adjusted, unlike EXS(*SYM*) deterministic selection.

Table 6.3: Evaluation of Exhaustive Search EXS(*SYM*) vs Tree-structured Parzen Estimator TPE(*SYM*) for Symmetry Optimization across grouping levels $n = 1$ to 5, $N_{\text{trials}} = 2000$.

		$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w	**EXS(*SYM*)**	**TPE(*SYM*)**	Δ	**EXS(*SYM*)**	**TPE(*SYM*)**	Δ	**EXS(*SYM*)**	**TPE(*SYM*)**	Δ	**EXS(*SYM*)**	**TPE(*SYM*)**	Δ	**EXS(*SYM*)**	**TPE(*SYM*)**	Δ
Mass	0	1169.86	1169.86	=	780.2	877.71	↑	730.3	764.06	↑	750.25	750.25	=	854.8	919.41	↑
Connection Degree	0	6	6	=	9	11	↑	12	11	↓	13	13	=	15	15	=
Symmetry	**1**	**0**	**0**	**=**	**0**	**0**	**=**	**0**	**0**	**=**	**0**	**0**	**=**	**0**	**0**	**=**
Beam Continuity	0	2	2	=	2	2	=	2	2	=	2	2	=	2	2	=



(a) EXS-Based Optimization of Symmetry vs n

(b) EXS-Based Optimization of Symmetry with corresponding Truss Configurations

(c) TPE-Based Optimization of Symmetry vs n

(d) TPE-Based Optimization of Symmetry with corresponding Truss Configurations

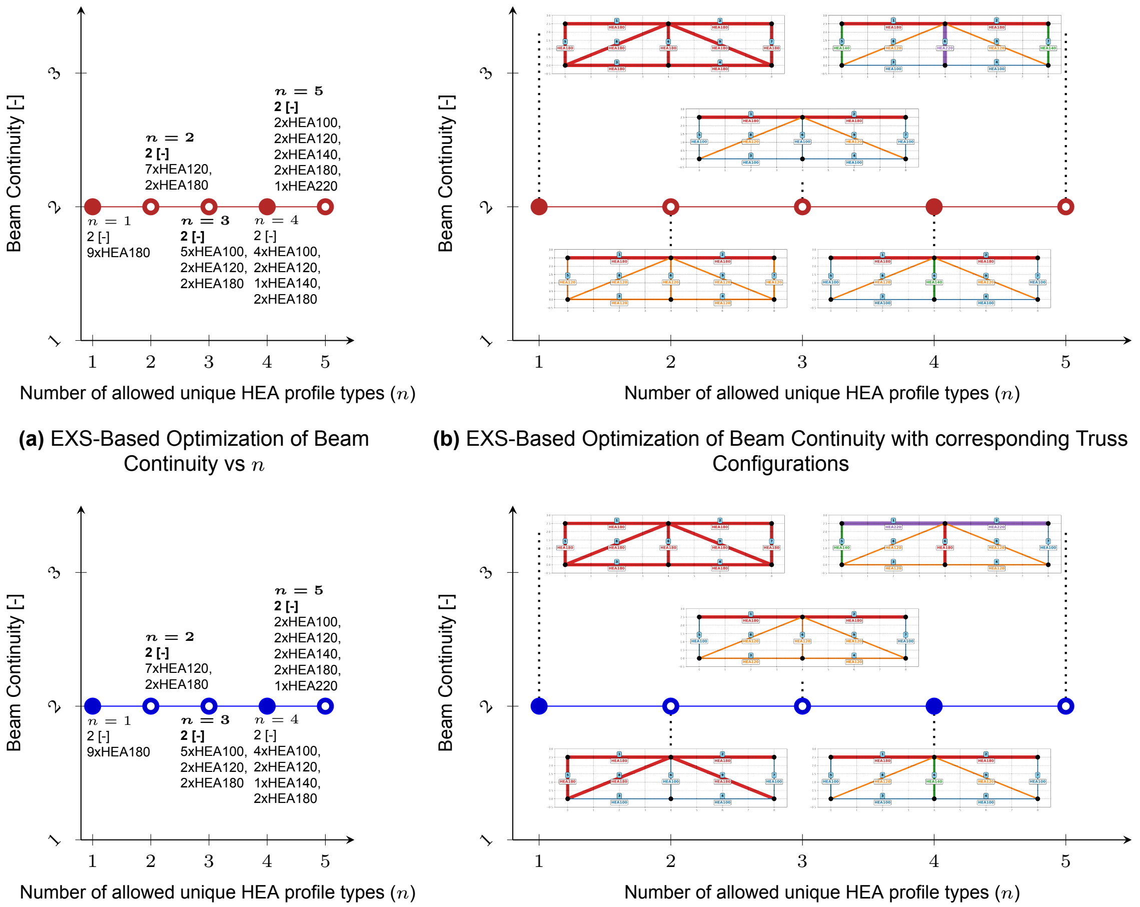
Figure 6.4: Comparison of Exhaustive Search (EXS) and Tree-structured Parzen Estimator (TPE) based optimization of symmetry versus the number of allowed unique HEA profile types (n) for $N_{\text{trials}} = 2000$.

6.1.4. Comparison: TPE(BCN) vs EXS(BCN)

For beam continuity, TPE(*BCN*) converges with EXS(*BCN*) on the primary objective (Table 6.4), yet identifies distinct truss configurations with varying secondary objective values. For $(n = 2)$, $(n = 3)$, and $(n = 5)$, TPE(*BCN*) configurations exhibit slight improvements or deteriorations in secondary metrics compared to EXS(*BCN*), driven by its stochastic sampling strategy [7] [8]. This exploratory diversity enables TPE(*BCN*) to sample configurations probabilistically, unlike EXS(*BCN*) deterministic selection. Figure 6.5 visualizes these distinctions, highlighting TPE(*BCN*) ability to uncover varied secondary performance while maintaining primary objective convergence.

Table 6.4: Evaluation of Exhaustive Search EXS(*BCN*) vs Tree-structured Parzen Estimator TPE(*BCN*) for Beam Continuity Optimization across grouping levels $n = 1$ to 5, $N_{\text{trials}} = 2000$.

		$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	$\mathcal{W}$	**EXS(*BCN*)**	**TPE(*BCN*)**	Δ	**EXS(*BCN*)**	**TPE(*BCN*)**	Δ	**EXS(*BCN*)**	**TPE(*BCN*)**	Δ	**EXS(*BCN*)**	**TPE(*BCN*)**	Δ	**EXS(*BCN*)**	**TPE(*BCN*)**	Δ
Mass	0	1169.86	1169.86	=	780.2	924.83	↑	730.3	764.06	↑	750.25	750.25	=	854.8	943	↑
Connection Degree	0	6	6	=	9	10	↑	12	11	↓	13	13	=	15	13	↓
Symmetry	0	0	0	=	0	1	↑	0	0	=	0	0	=	0	1	↑
Beam Continuity	**1**	**2**	**2**	**=**	**2**	**2**	**=**	**2**	**2**	**=**	**2**	**2**	**=**	**2**	**2**	**=**



(a) EXS-Based Optimization of Beam Continuity vs n

(b) EXS-Based Optimization of Beam Continuity with corresponding Truss Configurations

(c) TPE-Based Optimization of Beam Continuity vs n

(d) TPE-Based Optimization of Beam Continuity with corresponding Truss Configurations

Figure 6.5: Comparison of Exhaustive Search (EXS) and Tree-structured Parzen Estimator (TPE) based optimization of beam continuity versus the number of allowed unique HEA profile types (n) for $N_{\text{trials}} = 2000$.

6.2. TPE performance in single vs multi-objective scenarios

As described in the previous subchapter, one of the notable characteristics and inherent limitations of the TPE algorithm is its stochastic nature. When applied to single-objective optimization using the weighted sum approach, the TPE algorithm tends to overlook evident improvements, such as mass reductions while maintaining equivalent symmetry results, as demonstrated in the comparison of EXS(*SYM*) with TPE(*SYM*) in Section . This limitation stems from TPE stochastic sampling, which, under a weighted sum with zero weights for secondary objectives, neglects their optimization, potentially missing configurations that balance multiple objectives, as noted in Chapter 4. Consequently, this section evaluates the performance of single-objective TPE optimization against multi-objective TPE optimization, employing four distinct weight distributions, as presented in Table 6.5.

Table 6.5: Comparison of different weights combination for TPE distributions in parametric truss optimization.

	w_M	w_{CN}	w_{SYM}	w_{BCN}
TPE($\mathcal{W}_1$)	0.25	0.25	0.25	0.25
TPE($\mathcal{W}_2$)	0.4	0.1	0.4	0.1
TPE($\mathcal{W}_3$)	0.1	0.4	0.1	0.4
TPE($\mathcal{W}_4$)	0.4	0.05	0.5	0.05

To determine the most appropriate single-objective TPE optimizations for comparison with multi-objective TPE configurations, a strategic approach was adopted. Rather than posing this as a question, the analysis leverages a systematic methodology to select single-objective counterparts that maximize observed differences, thereby highlighting the impact of multi-objective weighting. A comprehensive comparative analysis was conducted for TPE($\mathcal{W}_1$), TPE($\mathcal{W}_2$), TPE($\mathcal{W}_3$), and TPE($\mathcal{W}_4$) against all single-objective TPE optimizations. Appendix E.1 encompasses detailed sections comparing each multi-objective TPE configuration with single-objective TPE for mass, connection degree, symmetry, and beam continuity, quantifying differences in objective values across (n from 1 to 5) with ($N_{\text{trials}} = 2000$) to identify the single-objective TPE yielding the most significant deviations. These comparisons, including visualizations of the respective truss configurations, are fully documented in Appendix E.1, providing a robust foundation for the analysis. This ensures a thorough evaluation of how weight distributions influence truss design outcomes.

As a result, Table 6.12 reveals the greatest number of differences between single-objective and multi-objective TPE configurations. It was observed that TPE($\mathcal{W}_1$), which equally prioritizes all objectives, exhibited the most differences with single-objective TPE for connection degree, recording 12 differences. Similarly, TPE($\mathcal{W}_2$), emphasizing mass and symmetry, showed the highest number of differences with single-objective TPE for connection degree. For analytical precision, TPE($\mathcal{W}_2$) was compared with the second-highest difference, beam continuity TPE, which recorded 6 differences, representing the second single-objective TPE with significant deviations. TPE($\mathcal{W}_3$), prioritizing connection degree and beam continuity, unsurprisingly exhibited the most differences with mass optimization. Finally, TPE($\mathcal{W}_4$), a unique case heavily prioritizing symmetry, recorded the most differences with connection degree but was compared with symmetry single-objective TPE to underscore the limitation of single-objective TPE, which omits logical solutions by disregarding objectives with zero weight. This selection strategy highlights the impact of multi-objective weighting on search space exploration, particularly for connection degree, which significantly constrains the design space due to its sensitivity.

Table 6.6: Number of Differences (Increases or Decreases) between Single-Objective TPEs and TPE($\mathcal{W}_1$), TPE($\mathcal{W}_2$), TPE($\mathcal{W}_3$), TPE($\mathcal{W}_4$) Configurations Across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

TPE (Single Objective)	$\mathcal{W}_1$	ΔTPE($\mathcal{W}_1$)	$\mathcal{W}_2$	ΔTPE($\mathcal{W}_2$)	$\mathcal{W}_3$	ΔTPE($\mathcal{W}_3$)	$\mathcal{W}_4$	ΔTPE($\mathcal{W}_4$)
Mass	0.25	7	0.4	4	0.1	11	0.4	0
Connection Degree	0.25	12	0.1	12	0.4	7	0.05	13
Symmetry	0.25	8	0.4	5	0.1	9	0.5	8
Beam Continuity	0.25	8	0.1	6	0.4	9	0.05	8

All TPE results in this section were generated with ($N_{\text{trials}} = 2000$), ensuring robust convergence for both single and multi-objective optimizations, as validated in Chapter 5. The subsections below (6.2.1 to 6.2.4) systematically analyze each multi-objective TPE configuration against its selected single-objective counterpart, elucidating trade-offs and the influence of weight distributions on truss configurations.

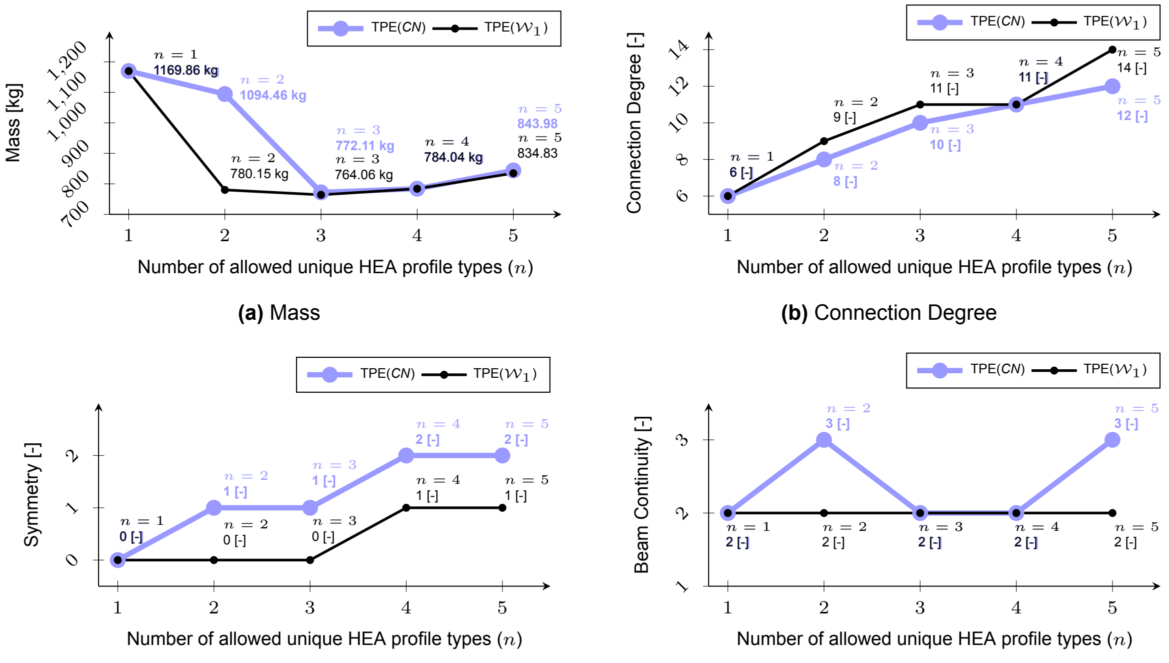
6.2.1. TPE($\mathcal{W}_1$) performance vs single objective TPE optimization scenarios

TPE($\mathcal{W}_1$), which uniformly distributes weights across all four objectives, is compared with single-objective TPE for connection degree TPE(*CN*). Given TPE probabilistic nature, TPE($\mathcal{W}_1$) is expected to achieve a more balanced performance across objectives. However, an initial examination of TPE($\mathcal{W}_1$) versus TPE(*CN*) for connection degree, as depicted in Subfigure 6.6b, reveals nuanced differences. Specifically, for connection degree, TPE(*CN*) outperforms TPE($\mathcal{W}_1$) slightly at ($n = 2$), ($n = 3$), and ($n = 5$), while performing equivalently at other n values. This marginal inferiority in connection degree suggests that TPE($\mathcal{W}_1$) sacrifices some performance in this objective. However, a broader analysis of TPE($\mathcal{W}_1$) performance across the secondary objectives of TPE(*CN*) — mass, symmetry, and beam continuity demonstrates its superior overall balance, rendering the connection degree difference negligible.

Table 6.7: Comparison of Tree-structured Parzen Estimator of Connection Degree TPE(*CN*) vs TPE($\mathcal{W}_1$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w(**TPE(*CN*)**)	$\mathcal{W}_1$	**TPE(*CN*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_1$)**	Δ
Mass	0	0.25	1169.86	1169.86	=	1094.46	780.15	↓	772.11	764.06	↓	784.04	784.04	=	843.98	834.83	↓
Connection Degree	1	0.25	6	6	=	8	9	↑	10	11	↑	11	11	=	12	14	↑
Symmetry	0	0.25	0	0	=	1	0	↓	1	0	↓	2	1	↓	2	1	↓
Beam Continuity	0	0.25	2	2	=	3	2	↓	2	2	=	2	2	=	3	2	↓

For instance, at ($n = 2$), as shown in Subfigure 6.6a, TPE($\mathcal{W}_1$) achieves a mass of 780 kg, approximately 300 kg lighter than TPE(*CN*) 1094 kg. Similarly, Subfigures 6.6c and 6.6d for ($n = 2$) show TPE($\mathcal{W}_1$) outperforming TPE(*CN*) in symmetry and beam continuity. These subfigures, part of Figure 6.6, depict TPE($\mathcal{W}_1$) objective function represented with black line consistently below TPE(*CN*) objective function represented with bluish line, signifying enhanced secondary objective performance. While not unexpected, this improvement in mass, symmetry, and beam continuity for TPE($\mathcal{W}_1$) comes at the cost of a minor increase in connection degree (e.g., 9[-] for TPE($\mathcal{W}_1$) vs. 8[-] for TPE(*CN*) at ($n = 2$)), which is relatively insignificant in the context of overall truss performance.



(a) Mass

(b) Connection Degree

(c) Symmetry

(d) Beam Continuity

Figure 6.6: Comparison of TPE($\mathcal{W}_1$) and TPE(*CN*) Optimization across Objectives for $N_{\text{trials}} = 2000$.

This analysis highlights TPE($\mathcal{W}_1$) capacity to leverage non-zero weights for a broader search space exploration, unlike TPE(*CN*) constrained focus on connection degree. A structured evaluation approach, first comparing primary objectives and then assessing secondary objectives reveals TPE($\mathcal{W}_1$) consistent superiority, as its performance curves typically lie below those of TPE(*CN*) for secondary objectives. This comparative logic is applied in the following subsections.

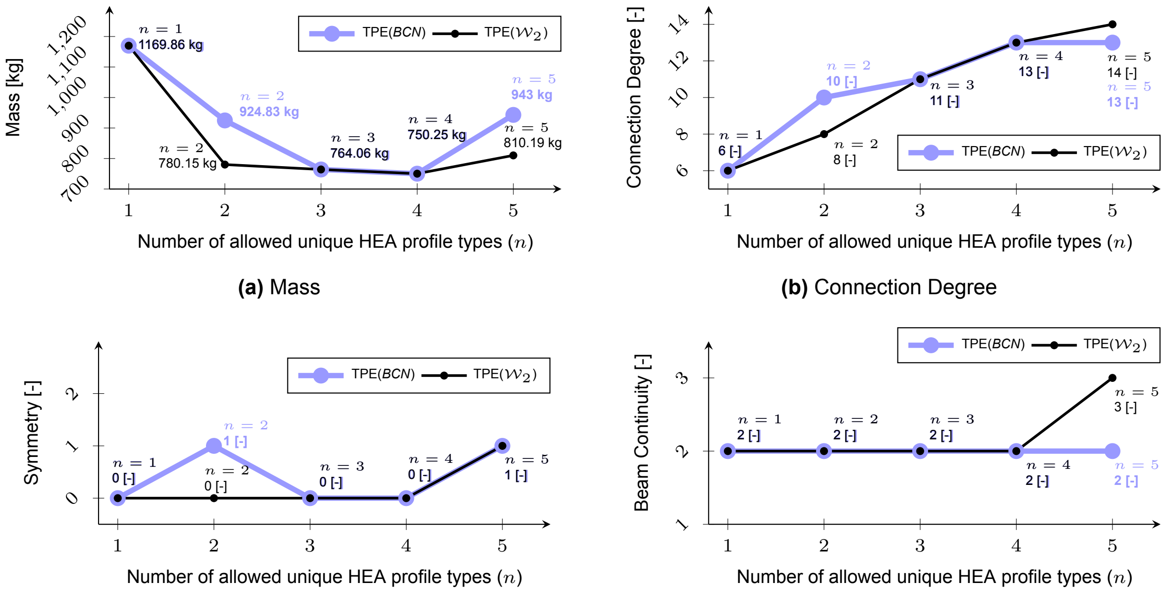
6.2.2. TPE($\mathcal{W}_2$) performance vs single objective TPE optimization scenarios

TPE($\mathcal{W}_2$), which prioritizes mass and symmetry as indicated in Table E.8 is evaluated against single-objective TPE for beam continuity (TPE(*BCN*)). Applying the analytical framework from the previous subsection, TPE($\mathcal{W}_2$) matches TPE(*BCN*) beam continuity performance for most (n) values, except at ($n = 5$), where TPE($\mathcal{W}_2$) records a slightly higher score (3[-] vs. 2[-] for TPE(*BCN*)). Concurrently, TPE($\mathcal{W}_2$) significantly reduces mass at ($n = 5$), achieving 810 kg compared to 943 kg for TPE(*BCN*). Notably, at ($n = 2$), TPE($\mathcal{W}_2$) equals TPE(*BCN*) in beam continuity while improving secondary objectives—mass, symmetry, and connection degree. Another compelling observation occurs at ($n = 3$) and ($n = 4$), where TPE($\mathcal{W}_2$) and TPE(*BCN*) yield identical results, underscoring the influence of weight distributions in aligning multi-objective and single-objective outcomes.

Table 6.8: Comparison of Tree-structured Parzen Estimator of Beam Continuity TPE(*BCN*) vs TPE($\mathcal{W}_2$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w(**TPE(*BCN*)**)	$\mathcal{W}_2$	**TPE(*BCN*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_2$)**	Δ
Mass	0	0.4	1169.86	1169.86	=	924.83	780.15	↓	764.06	764.06	=	750.25	750.25	=	943	810.19	↓
Connection Degree	0	0.1	6	6	=	10	8	↓	11	11	=	13	13	=	13	14	↑
Symmetry	0	0.4	0	0	=	1	0	↓	0	0	=	0	0	=	1	1	=
Beam Continuity	1	0.1	2	2	=	2	2	=	2	2	=	2	2	=	2	3	↑

This equivalence at ($n = 3$) and ($n = 4$) suggests that TPE($\mathcal{W}_2$) emphasis on mass and symmetry converges with TPE(*BCN*) optimal profiles within the 9-element truss (Figure 6.7).These findings highlight the value of varied weight distributions in expanding the optimization search space, a topic explored further in following subsections. TPE($\mathcal{W}_2$) ability to maintain or enhance beam continuity while improving secondary objectives exemplifies the advantages of multi-objective optimization, as non-zero weights facilitate exploration of configurations overlooked by TPE(*BCN*).



(a) Mass

(b) Connection Degree

(c) Symmetry

(d) Beam Continuity

Figure 6.7: Comparison of TPE($\mathcal{W}_2$) and TPE(*BCN*) Optimization across Objectives for $N_{\text{trials}} = 2000$.

6.2.3. TPE($\mathcal{W}_3$) performance vs single objective TPE optimization scenarios

This comparison examines TPE($\mathcal{W}_3$), which assigns weights of 0.4 to connection degree and beam continuity and 0.1 to mass and symmetry, against single-objective TPE for mass (TPE(*M*)). As expected, TPE(*M*) outperforms TPE($\mathcal{W}_3$) in mass optimization due to TPE($\mathcal{W}_3$) low mass weight. However, for all

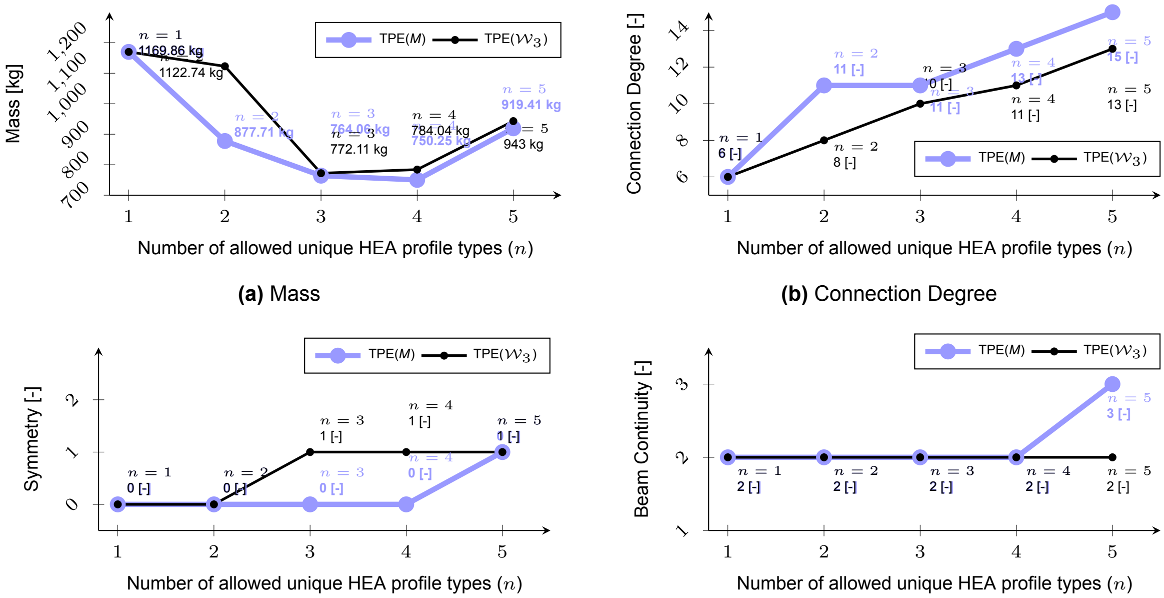
(n) values except ($n = 2$), TPE($\mathcal{W}_3$) mass results are remarkably close to those of TPE(*M*). In contrast, connection degree and beam continuity, TPE($\mathcal{W}_3$) prioritized objectives, consistently surpass TPE(*M*) performance. Yet, for mass and symmetry, TPE($\mathcal{W}_3$) yields inferior results compared to TPE(*M*).

Table 6.9: Comparison of Tree-structured Parzen Estimator of Mass TPE(*M*) vs TPE($\mathcal{W}_3$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w(**TPE(*M*)**)	$\mathcal{W}_3$	**TPE(*M*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_3$)**	Δ
Mass	1	0.1	1169.86	1169.86	=	780.15	1122.74	↑	730.27	772.11	↑	750.25	784.04	↑	810.19	943	↑
Connection Degree	0	0.4	6	6	=	9	8	↓	12	10	↓	13	11	↓	14	13	↓
Symmetry	0	0.1	0	0	=	0	0	=	0	1	↑	0	1	↑	1	1	=
Beam Continuity	0	0.4	2	2	=	2	2	=	2	2	=	2	2	=	3	2	↓

A particularly intriguing finding is TPE(*M*) symmetry performance, where, at ($n = 3$) and ($n = 4$), it surpasses TPE($\mathcal{W}_3$), despite symmetry being a secondary objective for TPE(*M*) and a less prioritized one for TPE($\mathcal{W}_3$). This suggests that prioritizing connection degree exerts significant influence on other objectives, reflecting its sensitivity in constricting the search space. This aligns with Chapter 4, which notes connection degree's arrangement-driven nature reduces configuration variability. Consequently, TPE($\mathcal{W}_3$) achieves superior connection degree results at ($n = 3$) and ($n = 4$) compared to TPE(*M*), but at the expense of increased symmetry. Similarly, beam continuity, a prioritized objective for TPE($\mathcal{W}_3$), outperforms TPE(*M*), though TPE(*M*) beam continuity score deviates only slightly at ($n = 5$).

These results indicate that prioritizing connection degree in weight distributions may be suboptimal, as it imposes substantial constraints on the computational space, adversely affecting other objectives. Notably, TPE(*M*), focused solely on mass, remains competitive in connection degree and beam continuity compared to TPE($\mathcal{W}_3$), despite the emphasis on these objectives.



(a) Mass

(b) Connection Degree

(c) Symmetry

(d) Beam Continuity

Figure 6.8: Comparison of TPE($\mathcal{W}_3$) and TPE(*M*) Optimization across Objectives for $N_{\text{trials}} = 2000$.

6.2.4. TPE($\mathcal{W}_4$) performance vs single objective TPE optimization scenarios

This subsection evaluates the performance of TPE($\mathcal{W}_4$), characterized by a weight distribution heavily favoring symmetry ($w_{SYM} = 0.5$) alongside weights for mass ($w_M = 0.4$), connection degree (w_{CN} $= 0.05$), and beam continuity ($w_{BCN} = 0.05$), respectively, within the multi-objective optimization framework. A notable preliminary finding warrants consideration: comprehensive analysis reveals that TPE($\mathcal{W}_4$) converges entirely with single-objective TPE for mass (TPE(*M*)) across all objectives and n values, yielding identical truss configurations. This convergence, observed in Appendix E.4 in Table

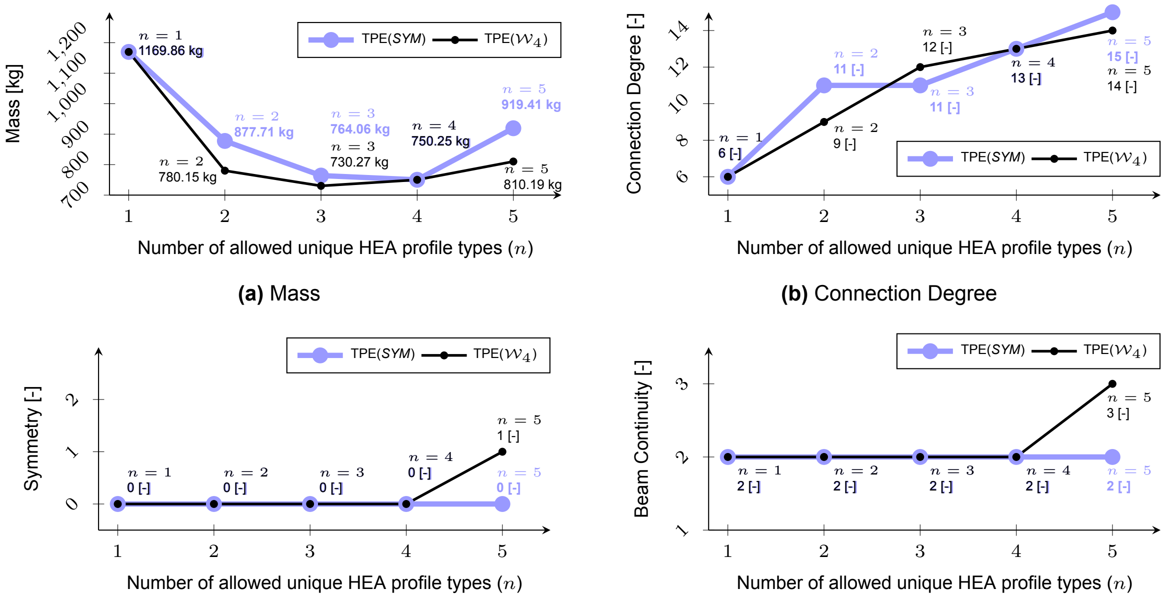
E.13, implies a meaningful comparison between TPE($\mathcal{W}_4$) and TPE(*M*), as their outcomes are indistinguishable, likely due to the significant mass weight in TPE($\mathcal{W}_4$) aligning with TPE(*M*) optimization focus. Consequently, to elucidate the implications of TPE($\mathcal{W}_4$) symmetry prioritization, this analysis shifts to a comparison with single-objective TPE for symmetry (TPE(*SYM*)), selected due to symmetry's dominant weight in TPE($\mathcal{W}_4$) and its substantial differences relative to single-objective TPE, second only to connection degree, as evidenced in Table 6.12.

Following Table 6.12, TPE($\mathcal{W}_4$) is thus compared against TPE(*SYM*) to explore the nuanced effects of multi-objective optimization when symmetry is heavily prioritized. This comparison reveals that TPE(*SYM*) consistently achieves an optimal symmetry score of 0[-] across all (n) values, reflecting its singular focus on symmetry. In contrast, TPE($\mathcal{W}_4$) matches this performance for (n from 1 to 4) but records a symmetry score of 1[-] at ($n = 5$), influenced by the competing priorities of other objectives. Nevertheless, at ($n = 5$), TPE($\mathcal{W}_4$) significantly enhances mass performance, reducing it by approximately 120 kg from 919 kg for TPE(*SYM*) to 810 kg, as indicated in Table E.15. This improvement, however, is accompanied by minor increases in symmetry and beam continuity for TPE($\mathcal{W}_4$).

Table 6.10: Comparison of Tree-structured Parzen Estimator of Symmetry TPE(*SYM*) vs TPE($\mathcal{W}_4$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w(**TPE(*SYM*)**)	$\mathcal{W}_4$	**TPE(*SYM*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_4$)**	Δ
Mass	0	0.4	1169.86	1169.86	=	877.71	780.15	↓	764.06	730.27	↓	750.25	750.25	=	919.41	810.19	↓
Connection Degree	0	0.05	6	6	=	11	9	↓	11	12	↑	13	13	=	15	14	↓
Symmetry	1	0.5	0	0	=	0	0	=	0	0	=	0	0	=	0	1	↑
Beam Continuity	0	0.05	2	2	=	2	2	=	2	2	=	2	2	=	2	3	↑

The non-zero weights assigned to secondary objectives in TPE($\mathcal{W}_4$) facilitate a broader exploration of the design space, enabling configurations that optimize mass while incurring only marginal symmetry degradation, unlike TPE(*SYM*) exclusive emphasis on symmetry, which constrains its scope. This underscores the dependency of TPE performance on weight distributions, as TPE($\mathcal{W}_4$) balanced weighting mitigates the limitations inherent in single-objective optimization.



(a) Mass

(b) Connection Degree

(c) Symmetry

(d) Beam Continuity

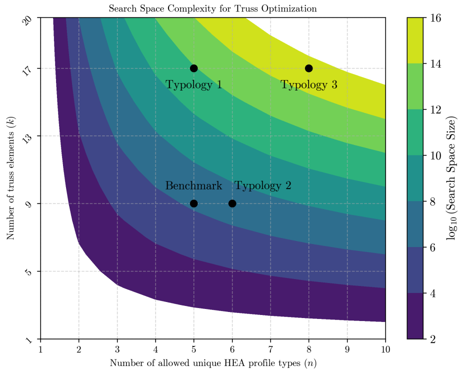
Figure 6.9: Comparison of TPE($\mathcal{W}_4$) and TPE(*SYM*) Optimization across Objectives for $N_{\text{trials}} = 2000$.

6.3. Influence of optimization parameters

This section evaluates the impact of key parameters on the computational efficiency and solution quality of the Tree-structured Parzen Estimator (TPE) algorithm applied to steel truss optimization. These parameters include the number of truss elements (k) the number of allowed HEA profiles (n), the number of trials ($N_{\text{trials}} = 2000$), and the number of weight distributions in multi-objective optimization. The analysis integrates theoretical insights from optimization literature with empirical observations from the benchmark problem, as detailed in Chapter 3, to provide a comprehensive understanding of parameter influences.

6.3.1. Search space complexity

The dimensionality of the search space in truss optimization is determined by the number of truss elements (k) and the number of allowed HEA profiles (n), resulting in a total of (n^k) possible configurations. For the benchmark problem, characterized by ($k = 9$) and ($n = 5$), the search space encompasses approximately 1.95 million configurations (5^9). In contrast, larger typologies, such as Typology 3 with ($k = 17$) and ($n = 8$), exhibit an exponentially larger search space of $8^{17} \approx 2.25 \times 10^{15}$ configurations. Notably, a comparative analysis between the benchmark problem and Typology 2 reveals that, despite their identical number of truss elements ($k = 9$), the two scenarios differ solely in the number of HEA different profiles analyzed (($n = 5$) for the benchmark versus ($n = 6$) for Typology 2). This minor increment of one additional HEA profile significantly alters the search space, as evidenced in Table 6.10. Specifically, the search space expands approximately five times from 1.95 million configurations in the benchmark problem to 10.08 million in Typology 2, underscoring the exponential sensitivity to changes in (n). These values are comprehensively detailed in Table 6.10, which quantifies the search space sizes across the benchmark problem and Typologies 1–3, accompanied by their logarithmic scales $\log_{10}(n^k)$ to effectively illustrate the exponential growth. Complementing this, the contour plot in Figure 6.10 provides a visual representation of this growth as a function of (n) and (k), delineating regions where the search space magnitude renders exhaustive search methods computationally infeasible. Consequently, this exponential expansion precludes the practicality of exhaustive search for larger truss configurations, necessitating the adoption of efficient optimization algorithms such as the Tree-structured Parzen Estimator (TPE). TPE leverages Bayesian optimization to strategically prioritize high-performing configurations, thereby mitigating the computational burden [9]. The heightened complexity associated with larger truss configurations amplifies computational demands, necessitating meticulous parameter tuning to achieve an optimal balance between exploration and resource constraints, as highlighted by [47].



(a) Contour plot illustrating the logarithmic search space size ($\log_{10}(n^k)$) as a function of the number of HEA profiles (n) and truss elements (k).

Problem	n	k	Search Space Size (n^k)	$\log_{10}(n^k)$
Benchmark	5	9	1,953,125	6.29
Typology 1	5	17	762,939,453,125	11.88
Typology 2	6	9	10,077,696	7.00
Typology 3	8	17	2,251,799,813,685,248	15.35

(b) Search space sizes for truss optimization problems, detailing the number of HEA profiles (n), truss elements (k), total configurations (n^k), and logarithmic search space size ($\log_{10}(n^k)$).

Figure 6.10: Visualization of Search Space Complexity in Truss Optimization Problems. The contour plot (a) shows the exponential growth of the search space with respect to the number of HEA profiles (n) and truss elements (k), while the table (b) provides exact values for the benchmark and typologies.

6.3.2. Amount of trials N_{trials}

The extent of search space exploration in the Tree-structured Parzen Estimator (TPE) is governed by the number of trials (N_{trials}), a parameter that directly influences both the quality of solutions and the computational expenditure. Higher values of (N_{trials}) increase the probability of identifying optimal or near-optimal solutions by facilitating more extensive sampling of the search space, however this comes at the cost of a linear increase in computational time. Validation experiments conducted on the benchmark problem, defined by ($k = 9$) truss elements and ($n = 5$) HEA profiles, demonstrated that ($N_{\text{trials}} = 2000$) was sufficient to achieve convergence comparable to that of Exhaustive Search (EXS), as detailed in Chapter 5. In contrast, for larger search spaces, such as those encountered in Typology 3 with ($k = 17$) and ($n = 8$), the determination of an appropriate (N_{trials}) poses a significant challenge due to the expansive configuration space, which comprises $8^{17} \approx 2.25 \times 10^{15}$ configurations as noted in the Section 6.3.1. To address this, heuristic strategies, such as monitoring convergence through iterative plotting of the best objective value against trial number, provide a pragmatic approach for selecting (N_{trials}), with optimization terminated when incremental improvements reach a plateau [7]. In practical applications, adjustments to (N_{trials}) were made for larger problems, capitalizing on enhanced computational resources, such as the Blue Delft computing cluster, to ensure run times remained feasible within a 24-hour limit [18]. This adaptive approach aligns with best practices in Bayesian optimization, where trial counts must balance solution quality with computational efficiency, particularly in high-dimensional structural optimization problems [36].

6.3.3. Amount of weight distributions

In multi-objective optimization, the granularity of the approximated Pareto front is dictated by the number of weight distributions, with each distribution necessitating an independent Tree-structured Parzen Estimator (TPE) optimization run. Consequently, the computational time exhibits a linear relationship with the number of distributions, as each additional run proportionally increases the overall processing duration. This study evaluated configurations comprising 4, 35, and 177 weight distributions to systematically investigate the trade-offs among objectives, as outlined in Chapter 4. A greater number of distributions yields a more refined representation of the Pareto front, thereby enhancing the resolution of trade-off analyses, however, this refinement substantially elevates the computational demand. The selection of the number of distributions is profoundly influenced by stakeholder preferences, clearly delineated requirements may permit a reduced number of distributions, whereas scenarios characterized by uncertainty necessitate a larger set to comprehensively capture a diverse spectrum of trade-offs. This linear scaling underscores the imperative for judicious allocation of computational resources to effectively balance the depth of trade-off analysis with practical time constraints.

6.3.4. Computational constraints and practical considerations

Practical limitations, including available computational power and time, critically shape parameter selection. To ensure feasibility, analyses were designed to complete within 8 hours on a personal computer, aligning with typical overnight processing, or up to 24 hours on the Blue Delft computing cluster for larger problems (e.g., Typology 3) [18]. For instance, in the benchmark problem, (N_{trials}) and 4 distributions were manageable within 8 hours. For larger trusses with increased (k), (n), (N_{trials}) (Table 6.11), and distributions (e.g., 177 distributions), the Blue Delft cluster enabled extended runs while adhering to the 24-hour limit [18].

Table 6.11: Impact of Optimization Parameters on TPE Performance

Parameter	Impact on Search Space	Impact on Computational Time	Impact on Solution Quality
Number of Elements (k)	Exponential (n^k)	Increases with evaluation complexity	Larger k requires more exploration
Number of Profiles (n)	Exponential (n^k)	Increases with search space size	More profiles increase diversity
Number of Trials (N_{trials})	None	Linear increase	Higher N_{trials} improves quality
Number of Distributions	None	Linear increase	More distributions enhance Pareto resolution

This strategic use of computational resources underscores the need to balance parameter settings with stakeholder-driven design requirements, where clarity in preferences can reduce the number of distributions, thereby optimizing computational efficiency [47]. The interplay of truss elements, HEA profiles, trials, and weight distributions profoundly affects the computational efficiency and solution quality of TPE-based optimization. The exponential growth of the search space with (m) and (n) necessitates efficient sampling, while (N_{trials}) and distributions linearly impact computational time, requiring careful calibration to achieve robust solutions within practical constraints. By leveraging heuristic convergence monitoring and high-performance computing resources, such as the Blue Delft cluster, this study balanced precision and feasibility, aligning with the stakeholder-driven design objectives.

6.4. Multi-Plot visualization analysis

As established in Chapter 5, multi-parallel plots are a highly effective tool for dynamically visualizing the trade-offs in multi-objective optimization of steel truss configurations. This section elaborates on their utility in representing the optimization results for the 9-element benchmark truss, focusing on four objectives, mass, connection degree, symmetry, and beam continuity alongside their stakeholder-defined weights and the number of allowed HEA profiles (n).

The multi-parallel plot consists of multiple axes to capture the optimization landscape comprehensively. It includes four axes for the objective values (mass, connection degree, symmetry, and beam continuity), four axes for their corresponding weights $\{w_M, w_{CN}, w_{SYM}, w_{BCN}\}$, and one axis for ($n$ from 1 to 5). Each line in the plot represents a unique truss configuration, connecting values across these axes to illustrate how a specific combination of objectives, weights, and n corresponds to a feasible design. The plot covers the entire search space defined by the weight distributions considered, with more distributions resulting in a denser plot that includes additional lines representing diverse configurations. This scalability enhances the granularity of trade-off analysis, as demonstrated in Figure 6.11 and Figure 6.12, where Figure 6.11 shows the plot for 35 weight distributions, providing a coarse representation of the trade-off space, while Figure 6.12shows the plot for 177 weight distributions, offers a finer resolution, capturing a broader range of truss configurations.

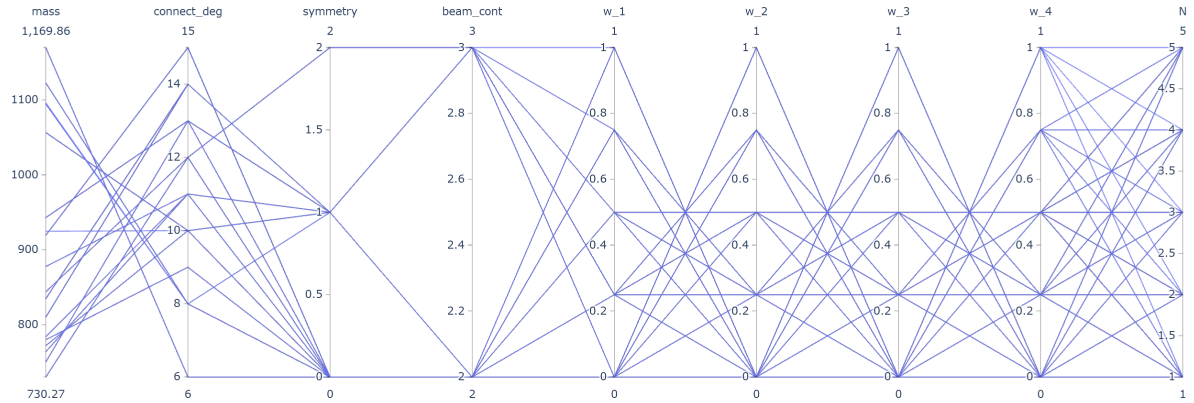


Figure 6.11: Multiparallel plot benchmark problme, $N_{\text{trials}} = 2000$, 35 distributions.

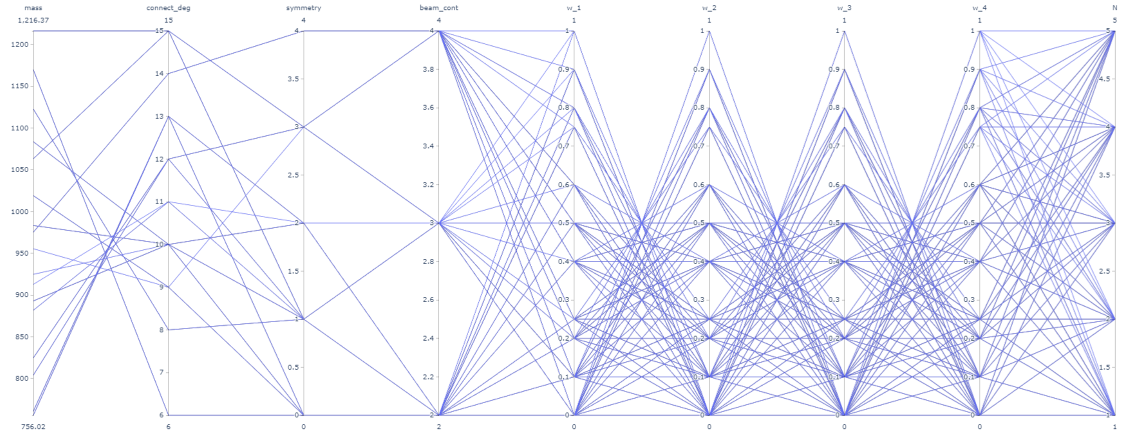


Figure 6.12: Multiparallel plot benchmark problme, $N_{\text{trials}} = 20$, 177 distributions.

A notable feature of the multi-parallel plot is the geometric pattern observed in the weight axes, which may seem counterintuitive given the probabilistic nature of the Tree-structured Parzen Estimator (TPE) optimization. This pattern arises from the predefined weight distributions used in the optimization, as detailed in Chapter 4. The study employs two sets of distributions: a coarse grid with 35 combinations derived from weight values $\{0.0, 0.25, 0.5, 0.75, 1.0\}$, and a finer grid with 177 combinations from $\{0.0, 0.1, 0.2, 0.25, 0.4, 0.5, 0.6, 0.75, 0.8, 0.9, 1.0\}$. These discrete values, constrained by the normalization condition ($w_M + w_{CN} + w_{SYM} + w_{BCN} = 1$), produce a regular geometric arrangement. For example, when ($w_M = 1$) , the other weights are zero, resulting in lines connecting the maximum value on the (w_M) axis to zero on the other weight axes, creating a structured visual pattern.

The dynamic interactivity of the multi-parallel plot is a key advantage, enabling stakeholders to retrieve specific truss configurations from the optimization database by defining boundary intervals on the axes. By selecting ranges for objectives, weights, or nmber of allowed HEA profiles (n), stakeholders can filter the plot to highlight configurations that meet these criteria, instantly visualizing the corresponding truss designs. This functionality, implemented using interactive visualization tools such as Plotly [39], allows real-time exploration of the trade-off space. For instance, binding the mass axis to a specific interval and restricting n to a particular value retrieves all configurations satisfying these constraints, displaying their detailed parameters, such as HEA profile assignments and objective values. This interactivity enhances stakeholder engagement by providing an intuitive interface to navigate complex design spaces, aligning with the thesis's objective of facilitating informed decision-making.

To illustrate, consider a scenario where stakeholders prioritize a mass range of [730.27, 850] kg, a symmetry score of [0, 1], ($n = 3$), and specify weights of ($w_M = 0.4$) and ($w_{SYM} = 0.2$) for mass and symmetry, respectively. By binding these axes in the multi-parallel plot, the tool filters and highlights all truss configurations meeting these constraints, displaying their connections across other axes (e.g., connection degree, beam continuity, and weights). This process, visualized in Figure 6.13.

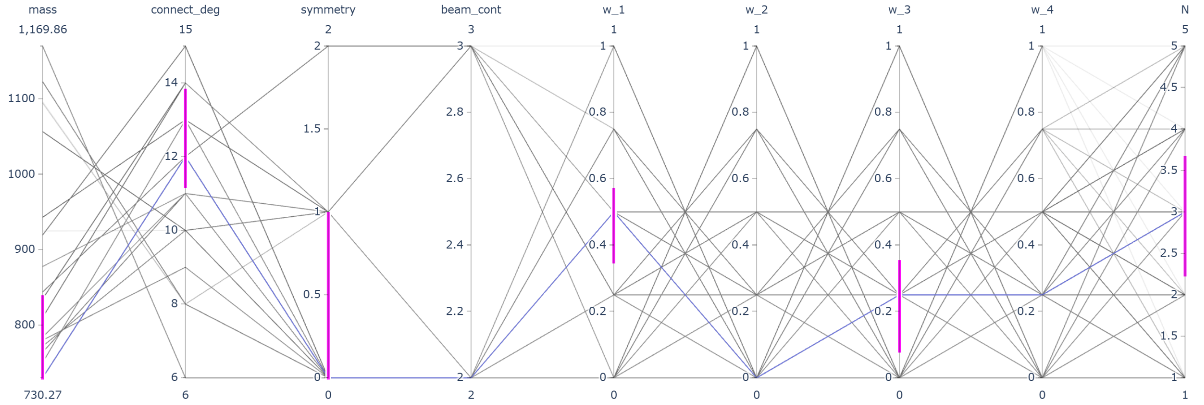


Figure 6.13: Multiparallel plot benchmark problme, $N_{\text{trials}} = 2000$, 35 distributions - boundary conditions

Retrieving one or multiple configurations from the database, presenting their structural parameters and visual representations of the found truss configuration based on the imposed boundaries on the obejctive's axes (Figure 6.14). This example demonstrates the plot's ability to translate stakeholder preferences into actionable design solutions, enhancing transparency and decision-making efficiency.

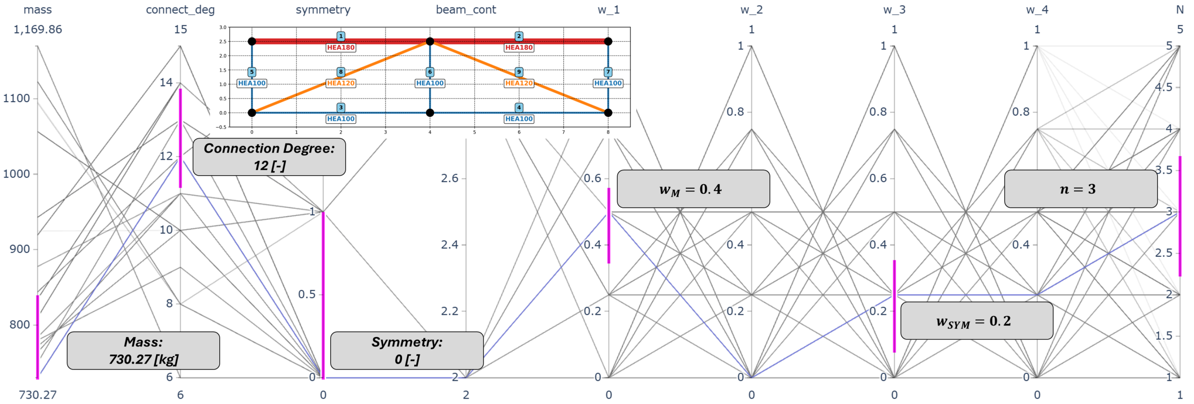
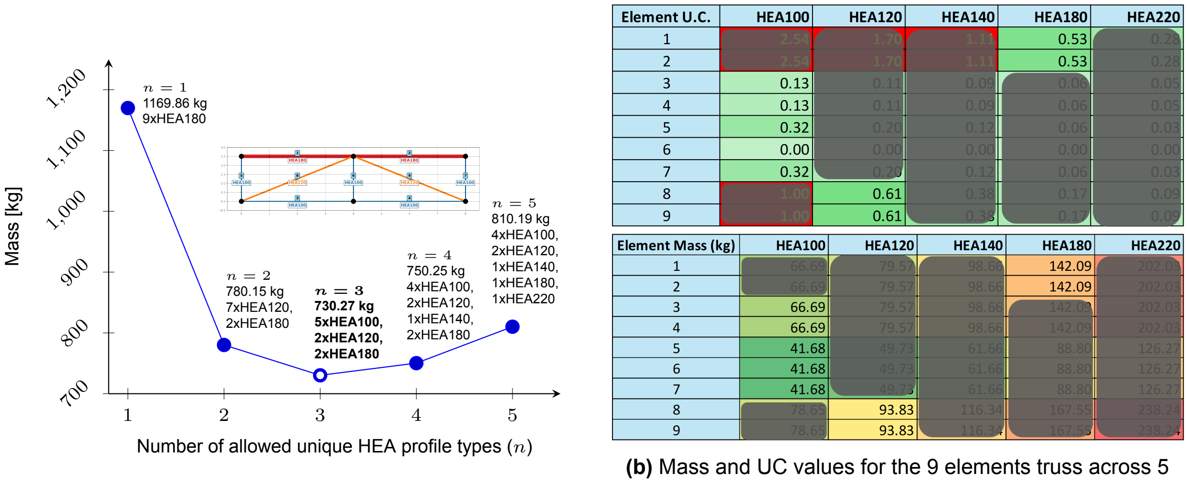


Figure 6.14: Multiparallel plot benchmark problme, $N_{\text{trials}} = 2000$, 35 distributions - determining a potential suitable truss configuration

6.5. Interpretation of structural outcomes

Throughout this thesis, the focus was on optimizing two conflicting objectives: mass, which becomes more optimal with greater uniqueness in the model since each element can be precisely tailored to the applied load, thus featuring less overdesign, versus structural complexity objectives, including connection degree, symmetry, and beam continuity, which become more optimal with standardization of the truss elements since the more standard elements are used in the truss configuration, the simpler the structural implementation becomes [7]. An interesting observation arose during the analysis, given that the trusses analyzed were structurally determinate, rectangular, and symmetric in topology (not to be confused with symmetry in element profiles), they have the feature of distributing loads equivalently through all symmetric elements, causing repetition in some elements [49]. This can be seen in Figure 6.15, where, while optimizing for mass and expecting that the best mass truss configuration would occur at the highest value of (n) (in this case, $(n = 5)$), the most optimal mass for the truss configuration was actually achieved at $(n = 3)$. It can be observed that the minimum of the function is located at $(n = 3)$ and corresponds to the lowest mass identified for this truss, 730.27 kg, . This unexpected result indicates that the symmetric topology naturally groups elements under similar loading conditions, reducing the need for excessive profile diversity. For a detailed representation of the groupings and corresponding configurations for $(n = 1)$ and $(n = 2)$, derived following the same procedure, refer to Appendix C.

As can be followed from Subfigure 6.15b, examination of the (UC) values for each element reveals a distinct formation of three groups of elements. In this case, the most logical selection to determine the lowest mass is to select the first valid (UC) values after those that are invalid. This approach identifies suitable HEA profiles with the best-balanced (UC) values available overall (which, in other words, means minimizing overdesign). For the optimization of the mass objective, elements 1 and 2 are assigned the profile HEA180, elements 3 to 7 are assigned the profile HEA100, and elements 8 and 9 are assigned the profile HEA120, thus forming three distinct groups of elements. This grouping strategy leverages the truss's load distribution to minimize material use while ensuring structural safety with (UC ≤ 1) [21]. The number of groups indicates the value of (n) for which the minimal mass occurs. When the most optimal truss configuration from the perspective of the mass objective for higher values of (n), such as $(n = 4)$ and $(n = 5)$, is examined, an immediate increase in mass is observed. This is explained by the fact that the system is forced to select four or five groups of HEA profiles, which exceeds the optimal number of groups (three), and thus forces the truss to incorporate less optimal elements from the perspective of (UC) values (still feasible but with a higher rate of overdesign). This highlights a trade-off between profile diversity and efficiency, as excessive groups lead to suboptimal material use.



Element U.C.	HEA100	HEA120	HEA140	HEA180	HEA220
1	2.54	1.70	1.11	0.53	0.28
2	2.54	1.70	1.11	0.53	0.28
3	0.13	0.11	0.09	0.06	0.05
4	0.13	0.11	0.09	0.06	0.05
5	0.32	0.20	0.12	0.06	0.03
6	0.00	0.00	0.00	0.00	0.00
7	0.32	0.20	0.12	0.06	0.03
8	1.00	0.61	0.38	0.17	0.09
9	1.00	0.61	0.38	0.17	0.09

Element Mass (kg)	HEA100	HEA120	HEA140	HEA180	HEA220
1	66.69	79.57	98.66	142.09	202.03
2	66.69	79.57	98.66	142.09	202.03
3	66.69	79.57	98.66	142.09	202.03
4	66.69	79.57	98.66	142.09	202.03
5	41.68	49.73	61.66	88.80	126.27
6	41.68	49.73	61.66	88.80	126.27
7	41.68	49.73	61.66	88.80	126.27
8	78.65	93.83	116.34	167.55	238.24
9	78.65	93.83	116.34	167.55	238.24

(a) TPE-Based Optimization of Mass vs n

(b) Mass and UC values for the 9 elements truss across 5 allowed unique HEA profiles ($n = 3$) *(highlighted)* grouping

Figure 6.15: Comparison of TPE-based mass optimization and constrained mass with UC values for a 9-element truss. ($n = 3$)

This principle is crucial for understanding how to balance different objectives. Following mass optimization, the structural complexity objectives behave differently. As previously noted, structural objectives such as connection degree are influenced by both the position of the different element profiles with respect to each other and their overall mass. Consequently, for structural complexity objectives, the grouping strategy applied in mass optimization does not entirely hold. All objectives related to structural complexity tend to worsen as the allowed number of HEA profiles increases, however, as a baseline, adopting the optimal number of HEA groups for mass ($n = 3$) provides a beneficial starting point to balance these with the remaining structural objectives.

Insights from Table 1 are emphasized next, given the applied distributions where all objectives are uniformly prioritized in TPE($\mathcal{W}_1$), mass and symmetry are prioritized in TPE($\mathcal{W}_2$), and connection degree and beam continuity are prioritized in TPE($\mathcal{W}_3$) these three distributions were selected for analysis to explore a significant aspect. (Note that TPE($\mathcal{W}_4$) was excluded, as its particular bias would make its inclusion unfair; thus, the analysis focuses on TPE($\mathcal{W}_1$), TPE($\mathcal{W}_2$), and TPE($\mathcal{W}_3$). This selection offers a balanced perspective: $\mathcal{W}_1$ provides a centered approach, $\mathcal{W}_2$ emphasizes mass and symmetry with weights of 0.4, and $\mathcal{W}_3$ emphasizes connection degree and beam continuity with weights of 0.4.

Table 6.12: Number of Differences (Increases or Decreases)i between Single-Objective TPEs and TPE($\mathcal{W}_1$), TPE($\mathcal{W}_2$), TPE($\mathcal{W}_3$) Configurations Across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

TPE (Single Objective)	$\mathcal{W}_1$	ΔTPE($\mathcal{W}_1$)	$\mathcal{W}_2$	ΔTPE($\mathcal{W}_2$)	$\mathcal{W}_3$	ΔTPE($\mathcal{W}_3$)	ΔSum
Mass	0.25	7	0.4	4	0.1	11	22
Connection Degree	0.25	12	0.1	12	0.4	7	31
Symmetry	0.25	8	0.4	5	0.1	9	22
Beam Continuity	0.25	8	0.1	6	0.4	9	23

Following Table 1, the number of differences between single-objective TPE and multi-objective TPE configurations reveals that the greatest number of differences arises with the objective of connection degree, as this is one of the most restraining objectives out of all four applied in this study, since it is strongly related to the location of the elements. This sensitivity arises because connection degree depends on the arrangement of profiles at nodes, significantly constraining the design space. Consequently, this objective requires careful treatment and should not be assigned a high weight in the problem, as doing so would strongly restrict the available search space. High weights on connection degree can limit exploration, potentially overlooking balanced solutions [3].

7 Reflection, limitations and future work

This chapter begins with a brief personal reflection on the research process, highlighting key lessons learned, challenges encountered, and strategic decisions made throughout the thesis trajectory. These reflections aim to provide insight into the development of the methodology, the reasoning behind specific design choices, and considerations that would inform a similar study if conducted again. Following this, the chapter evaluates the limitations of the multi-objective optimization framework applied to a 2D, structurally determinate, rectangular steel truss using the Tree-structured Parzen Estimator (TPE). Despite promising results, several challenges restrict the framework's applicability and effectiveness, which are addressed below, followed by actionable future work to enhance its potential.

7.1. Reflection

Time Management and Project Scope

The thesis comprehensively included four truss topologies, multiple weight distributions (including up to 177 combinations), and two optimization algorithms, placing significant demands on time constraints. The extensive analysis of multiple topologies required weeks but produced similar outcomes due to their shared rectangular geometry and uniform loading conditions Appendix A. A more focused scope, concentrating on the 9-element benchmark truss with a reduced set of weight distributions, could have facilitated a deeper investigation of results. To address computational demands, the Blue Delft supercomputer was integrated late in the project, accelerating result generation for larger trusses. However, it provided limited analytical benefits over local computations, which could process the 9-element truss in under a day. Engaging Blue Delft for simplistic problems was inefficient, as standard computers suffice. Blue Delft is better suited to complex problems, such as 3D trusses with non-uniform loading or dynamic analysis. Its late implementation coincided with recognizing uniform topology outcomes due to grouping effects (see Appendix C), but time constraints prevented redirecting efforts toward diverse typologies. Earlier use of Blue Delft could have identified this limitation sooner, allowing exploration of varied truss configurations.

Implementation of Grasshopper for a Simple Truss Configuration

The decision to employ Grasshopper with Karamba3D and Colibri (see Section 4.2) plugins for modeling and analyzing a simplistic the 9-element truss benchmrk was based on the initial expectation that the research would address complex truss configurations requiring advanced computational tools. However, for a statically determinate 2D truss with straightforward axial force analysis, this approach proved overly complex, as a Python-based structural analysis would have been more efficient. The choice arose from early uncertainty regarding the research trajectory, which aimed to develop a model capable of handling diverse truss typologies. Consequently, the setup and integration of sophisticated plugins consumed disproportionate time without yielding proportional benefits given the simplicity of the operated structures, underscoring a mismatch between the tool's capabilities and the problem's simplicity.

Algorithm Selection and the Suitability of TPE

The selection of the Tree-structured Parzen Estimator (TPE) as the primary optimization algorithm was motivated by its effectiveness in discrete, multi-objective optimization of steel truss configurations, particularly for categorical HEA profile assignments. TPE Bayesian framework, enhanced by pruning capabilities, requires (100 – 2,000) trials, significantly fewer than the 1.95 million needed by Exhaustive Search (EXS) (Section 5.5). It efficiently aligns with weighted scalarization of objectives: mass, symmetry, connection degree, and beam continuity reflecting stakeholder preferences. However, comparing TPE to EXS, which ensures global optima for small-scale problems but is impractical for larger ones, and NSGA-II, which excels in Pareto front generation but struggles with topological objectives proved challenging. . TPE slower convergence for symmetry at ($n = 5$), requiring 2,000 trials suggests limitations compared to alternatives like CMA-ES for specific problem types. While TPE was deemed suitable given the thesis's focus on a tailored optimization model rather than algorithm analysis, comprehensive investigation of alternative methods is needed, as other algorithms may offer superior performance. TPE optimization potential remains a topic for future research due to resource constraints on extensive algorithm evaluation.

Multi-Objective Optimization and Scalarization
This study aimed to balance multiple objectives — mass, symmetry, connection degree, and beam continuity using the Tree-structured Parzen Estimator (TPE) with weighted scalarization. The thesis framed this as multi-objective optimization to reflect the intent to address diverse stakeholder priorities, as scalarization aggregates objectives into a single score via a weighted sum. However, this terminology was partially inaccurate, as scalarization simplifies the optimization process by prioritizing a single composite score, limiting exploration of trade-offs across the Pareto front, where solutions optimize multiple objectives simultaneously (Section 4.4). For example, configurations optimized with weight set TPE($\mathcal{W}_2$), prioritizing mass and symmetry (Section 6.2.2) may neglect solutions excelling in connection degree or beam continuity. While true multi-objective methods, such as those generating Pareto fronts, could offer greater trade-off diversity, the project's scope and computational constraints hindered their adoption. Future research could develop methods to better balance trade-offs while preserving computational efficiency.

Connection Degree in Constructability Optimization
At the outset of this research, objectives were deliberately selected for their universal applicability across diverse civil engineering structures, including trusses, frames, and bridges, to ensure the optimization framework's versatility. Among these, the Connection Degree (CN) objective, defined as the sum of node degrees in the truss graph, emerged as a particularly compelling metric due to its topological complexity and relevance to constructability (Section 4.3.2). As detailed in Chapter 6, CN proved the most challenging objective to optimize, necessitating up to 500 trials for convergence with the Tree-structured Parzen Estimator (TPE) at ($n = 5$), reflecting its sensitivity to profile diversity.

The feasibility of including CN was scrutinized, given that the framework exclusively employs HEA profiles, which, despite varying grades, are compatible and pose minimal connection challenges. Nevertheless, CN was retained for two principal reasons. First, its topological nature provides critical insights into joint complexity, enhancing the framework's ability to address constructability concerns. Second, its universal applicability extends beyond HEA profiles to diverse structural systems, where profile variability significantly impacts fabrication. For instance, connecting dissimilar profiles necessitates tailored connection details or adjusted welding configurations, elevating fabrication costs, while ensuring minimal eccentricity often requires additional reinforcement, further complicating joint design (Section 4.3.2). Given potential conflicts arising from high CN values, such as increased fabrication complexity and cost, a deliberate decision was made to maintain CN at relatively low levels. This choice was particularly evident in the TPE($\mathcal{W}_3$) distribution that prioritized connection degree and beam continuity. Consequently, configurations with higher CN values, which could have enhanced structural performance through diverse profile assignments, were deprioritized, resulting in solutions that were not the most structurally optimal (Appendix E). This trade-off underscores the challenge of balancing constructability with structural efficiency, highlighting CN critical role in informing stakeholder-driven design decisions.

The digital optimization via TPE systematically navigated these complexities, outperforming human intuition, which might oversimplify joint design by assuming uniform profiles. This underscores CN value in validating the Python framework's robustness and its potential for broader structural applications. Future research could explore non-uniform topologies or varied profile types to further elucidate CN impact, enhancing its alignment with stakeholder-driven design priorities.

Absence of Direct Stakeholder Engagement
The optimization framework utilized predefined stakeholder preferences, assigning weights to objectives such as mass and symmetry without direct consultation with industry stakeholders. This approach risked overlooking critical priorities, such as fabricators' emphasis on beam continuity for prefabrication efficiency or architects' preference for symmetry for aesthetic coherence as well as other stakeholder-specific considerations not readily apparent to researchers. The absence of engagement with real stakeholders perspectives, compromised the framework's alignment with practical engineering requirements, potentially omitting key factors that could have refined objective selection and weight assignments. This highlights the challenge of ensuring practical applicability in academic research without direct stakeholder input, necessitating future collaboration to enhance the framework's relevance.

Truss Topology Analysis
This thesis investigated four distinct truss topologies: Benchmark, Typology 1, Typology 2, and Typology 3 (Figure G.5), all characterized by a rectangular geometry and a uniformly distributed load, as detailed in Chapter 2. The expectation was that these topologies would yield diverse optimal solutions, potentially forming a comprehensive Pareto front for multi-objective optimization or an exponential decrease in mass for single-objective optimization, as seen in prior literature [9] [32]. However, structural analysis conducted via Grasshopper and Karamba3D revealed that the outputs across these topologies were remarkably similar, contrary to initial expectations.

The uniformity in results stemmed from the consistent rectangular geometry and uniformly distributed load applied at nodes (Appendix A), which limited variability despite differences in the number of elements (9 or 17) and HEA profiles used (HEA100 to HEA220). or single-objective optimization, such as minimizing mass, optimal configurations clustered around specific values of (n), with $(n = 3)$ predominant for the Benchmark and Typology 1, and $(n = 2)$ for Typologies 2 and 3, where diagonal positioning induced compression in the top chord (Section A.2).This clustering indicates that the chosen parameters constrained the solution space, reducing the diversity of outcomes.

This outcome highlights a critical lesson in research design. While analyzing multiple topologies validated the adaptability of the Python code implemented for optimization , it did not significantly enhance the diversity of analytical outcomes. The substantial time invested in evaluating these topologies, while useful for code verification, was not the most efficient use of resources. Future research could explore varied truss geometries (e.g., triangular or arched) and diverse load scenarios to generate more varied results, providing a robust testbed for the computational framework and advancing structural optimization insights.

Human versus Digital Optimization Approaches
An insightful observation arises from the TPE optimization results for different values of (n) , as presented in Section A.1. A human engineer could apply engineering assumptions to determine the exact HEA profiles for each of the 9 elements in the benchmark truss, achieving an optimal configuration up to $(n = 3)$, which corresponds to the lowest mass of 730.27 kg. Nevertheless, for $(n = 4)$ and $(n = 5)$, identifying the optimal truss configuration becomes challenging without calculating the mass for potential variants, as the complexity of profile assignments increases.

Initially, one might question the need to evaluate configurations with $(n = 4)$ or $(n = 5)$, since they exceed the optimal mass achieved at $(n = 3)$ in single-objective optimization for mass. For example, prior studies illustrated an exponential decay in mass towards higher values of (n) , as depicted in Figure A.9. However, due to the uniform rectangular topology and consistently applied uniformly distributed load, such a decay ws. not observed, as the solution space remained constrained across all values of (n). In multi-objective optimization, where objectives like symmetry, connection degree, and beam continuity are considered alongside mass, the application of algorithms like TPE becomes essential. For instance,following from Figure 6.7 comparing TPE($\mathcal{W}_2$) (prioritizing mass and symmetry) with TPE(*BCN*), the optimal configuration for TPE($\mathcal{W}_2$) was found at $(n = 4)$, suggesting that higher (n) values can yield optimal solutions when balancing multiple objectives. This demonstrates the value of digital tools in navigating complex trade-offs, where human intuition alone may struggle to identify optimal configurations for higher (n), thus highlighting the complementary role of computational optimization in multi-objective truss design.

7.2. Model and computational limitations

Bias in grasshopper Unity Check calculations: The Grasshopper software calculates unity checks (UC) for each element based on the assumption that the entire truss is composed of a single HEA profile type. However, the optimization process generates truss configurations incorporating multiple profile types, which can alter force distributions within the truss. Consequently, the (UC) values used may not accurately reflect the true structural performance of these configurations, introducing bias in the selection process. Post-design validation checks could partially mitigate this issue, but they risk excluding potentially optimal solutions not selected due to biased (UC) values. Calculating (UC) for all possible profile combinations prior to optimization is computationally infeasible due to the exponential growth of the search space, as discussed in Chapter 6, where the benchmark problem yields approximately 1.95 million configurations.

Limited scope of objectives: The optimization framework focuses on four objectives—mass, connection degree, symmetry, and beam continuity. A broader range of objectives could enhance its relevance to steel truss design, including minimizing construction and material costs, reducing environmental impact through lower embodied carbon optimizing natural frequencies, and maintaining global structural stability. Additional objectives, such as joint complexity and manufacturability, could further align the framework with practical design requirements. Incorporating these and potentially other objectives would create a more comprehensive framework but would increase computational complexity.

Computational costs: The computational efficiency of the TPE algorithm is heavily influenced by parameters such as the number of trials (N_{trials}), amount of operated weight distributions, truss elements (k), and allowed HEA profiles (n). As detailed in Section 6.6, small increases in these parameters lead to exponential growth in the search space. For instance, the benchmark problem with ($k = 9$) and ($n = 5$) has approximately 1.95 million configurations, while a larger truss like Typology 3 with ($k = 17$) and ($n = 8$) yields $8^{17} \approx 2.25 \times 10^{15}$ configurations. This immense search space poses a significant computational challenge, highlighting the need for further research into methods that reduce TPE computational cost or engage more effective multi-objective optimization algorithms to handle such complexity efficiently.

Applicability to diverse structures: The framework is designed specifically for rectangular, 2D, structurally determinate steel trusses, limiting its applicability to complex structural systems encountered in civil engineering, such as non-symmetric topologies, three-dimensional trusses, or statically indeterminate structures. Extending the framework to these systems would require redefining optimization functions and recalibrating objectives to account for varied structural behaviors. This restriction constrains the framework's utility in addressing the diverse challenges of modern structural design.

Limited exploration of TPE hyperparameters: The study did not explore tuning TPE hyperparameters, such as the exploration-exploitation balance, number of startup trials, or gamma parameter in the expected improvement criterion. Further research into optimizing these hyperparameters could enhance TPE performance for truss design, potentially improving convergence rates and solution quality [9]. This area, left for future investigation, is critical for maximizing the algorithm's effectiveness in structural optimization.

Restriction to HEA profiles: The analysis is confined to HEA profiles, whereas truss designs often incorporate diverse profile types, such as I-sections, rectangular hollow sections, and many more [19]. Adapting the Grasshopper model to handle these profiles would require additional recalibration of parametric calculations, potentially affecting optimization outcomes and necessitating a reevaluation of objective functions.

Consideration of Diverse Material Options: The optimization framework is tailored specifically for steel trusses, with its parametric modeling, structural assessments, and grouping approaches centered on the characteristics of HEA steel profiles. This focus on a single material restricts the framework's applicability to other construction materials, such as concrete or timber. Investigating alternative materials could uncover distinct trade-offs in terms of structural efficiency, ease of construction, environmental sustainability, and fabrication requirements. For instance, timber may introduce constraints related to standardized dimensions or joint configurations, while composites could exhibit directional-dependent properties influencing load paths. Incorporating a wider range of materials would necessitate significant revisions to the parametric model, including adjustments to structural constraints (e.g., UC calculations) and optimization objectives (e.g., mass, structural complexity). Although such an expansion is outside the current study's scope, it offers a valuable opportunity to enhance the framework's versatility and relevance for a broader spectrum of civil engineering applications.

Limited comparison of optimization methods: The study evaluates only EXS and TPE algorithms, omitting a thorough comparison with other multi-objective optimization methods, such as genetic algorithms, greedy algorithm optimization, or other suitable algorithms for similar types of problems . Such a comparison would elucidate TPE relative strengths and trade-offs in solution quality, computational efficiency, and robustness. This was beyond the scope of the current research due to time and resource constraints but is recommended for future studies to thoroughly assess TPE efficacy in structural optimization.

7.3. Future work

This work opens several promising research directions for optimizing steel truss design. A primary focus should be improving the computational efficiency of the TPE algorithm, as its current performance creates a bottleneck. Exploring methods to accelerate TPE, or adopting alternative algorithms that are less computationally intensive without sacrificing result quality, could address this issue, building on the computational challenges discussed in Section Model and computational limitations. For instance, evaluating algorithms like CMA-ES, which may offer faster convergence for specific objectives like symmetry, could enhance performance, as suggested by TPE slower convergence at ($n = 5$).

To optimize analysis, future research should balance simplicity in the analyzed truss typologies, such as the 9-element truss benchmark, with the exploration of larger computational spaces and a broader range of weight distributions. This approach would maintain manageable computational demands while enabling richer insights into diverse configurations, addressing the time management challenges noted in Section 7.1. Additionally, leveraging high-performance computing resources like Blue Delft earlier in the research process could facilitate testing complex configurations, such as 3D trusses or non-uniform loading, and identify limitations in topology outcomes sooner.

Following this, another direction is improving the handling of multiple objectives. Instead of relying solely on weighted scalarization, which limits trade-off exploration, developing true multi-objective optimization methods, such as those generating Pareto fronts, could better balance objectives like mass, symmetry, and connection degree. This would provide a broader range of solutions reflecting diverse stakeholder priorities. A potential solution is to develop a single cost function representing all objectives, though this is challenging due to trade-off complexities, which could streamline the process and reduce computational demands [3].

To further enhance the framework, its robustness should be improved to accommodate diverse steel truss topologies, such as statically indeterminate or 3D structures. Exploring varied geometries, such as triangular or arched trusses, and diverse load scenarios would generate more diverse optimization outcomes, overcoming the uniformity observed in rectangular topologies. This would require redefining models and objectives to handle varied structural behaviors.

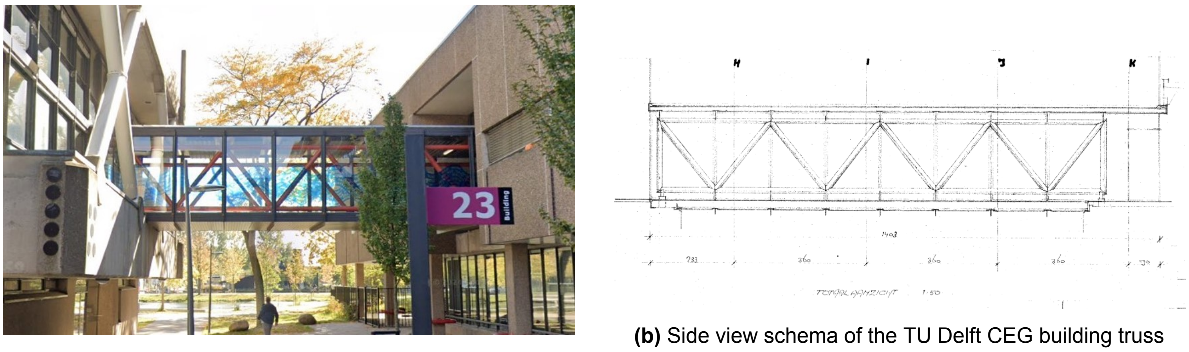
For simpler truss configurations, such as the 2D statically determinate truss studied, future work should prioritize efficient tools like Python-based structural analysis over complex platforms like Grasshopper with Karamba3D. This might reduce setup time and align tool complexity with problem simplicity.

Additionally, tuning TPE hyperparameters, such as the number of trials or gamma parameter, can be complex. Automating this process would improve adaptability across different truss problems, directly tackling the limitation in Section 7.2.

Expanding on the comparisons in Chapter 6, a broader evaluation with other multi-objective optimization methods, such as genetic algorithms or particle swarm optimization, would provide deeper insights into TPE strengths and weaknesses. This involves applying various algorithms to similar problems and analyzing the results.

To enhance practical relevance, future research should involve direct collaboration with industry stakeholders, such as fabricators and architects, to refine objective selection and weight assignments. This would address the absence of stakeholder engagement noted in Section 7.1, ensuring the framework aligns with real-world engineering priorities, such as prefabrication efficiency or aesthetic coherence.

Finally, applying the framework to real-world examples like the TU Delft CEG building truss, as shown in Figure 7.1, represents an initial step toward practical implementation. This example highlights the potential to validate the framework's utility, but since this truss uses diverse profiles, unlike the HEA profiles in our study, the framework must first be adapted to handle different profile types.



(a) TU Delft CG building rectangular truss bridge.

(b) Side view schema of the TU Delft CEG building truss bridge.

Figure 7.1: The TU Delft CEG building truss bridge, depicted in both a color photograph and a side view schematic, serves as a potential real-world case for applying the proposed optimization framework.

8 Conclusion

This research has explored the application of the Tree-structured Parzen Estimator (TPE) algorithm to optimize grouping strategies in a 9-element, 2D rectangular steel truss, with three additional similar typologies to validate findings (Figure G.5). The study addressed the challenge of balancing multiple, often conflicting objectives—structural mass, symmetry, connection complexity, and beam continuity—to support stakeholders in making informed and transparent design decisions. Through extensive experimentation, TPE showed promising effectiveness in navigating complex trade-off spaces, though constrained by weighted scalarization and untuned hyperparameters. The integration of visualization tools, such as multi-parallel plots, enhanced the interpretability of optimized designs, potentially empowering stakeholders to select configurations aligned with their priorities. The findings contribute to advancing structural engineering optimization, offering a promising framework for 2D truss design, and opening potential for further development in exploring more diverse truss typologies and advanced visualization techniques.

In the following section, we provide answers to the research questions:

To what extent can grouping strategies in steel truss structures be optimized to balance multiple, often conflicting objectives, there by enabling stakeholders to make informed and transparent design decisions?

The research suggests that TPE effectively optimizes grouping strategies for 2D rectangular truss typologies by efficiently exploring the design space and balancing conflicting objectives. Leveraging its Bayesian optimization approach, TPE identified configurations that minimize mass while maintaining acceptable levels of symmetry, connection complexity, and beam continuity. Multi-parallel plots (Figure 6.11) provided an interactive visualization of trade-offs, potentially enabling stakeholders to filter and select designs transparently, thus supporting informed decision-making.

RQ1: How do grouping configurations with a fixed number of allowed HEA profiles (n) affect structural mass, symmetry, connection complexity, and beam continuity?

Grouping configurations with a fixed number of HEA profiles (n) critically influence structural outcomes for the studied 2D rectangular truss typologies. In each case, there exists an optimal (n) that minimizes structural mass while balancing other objectives. For the benchmark 9-element truss, this optimal ($n = 3$), achieving a minimal mass of approximately 730.27 kg, though unity check (UC) calculation biases may affect precision, as validated through single-objective TPE and multi-objective TPE with symmetry prioritization (TPE($\mathcal{W}_4$)) (Figure E.13). This configuration effectively grouped elements into a set of three profile types (HEA180, HEA100, HEA120) based on load distribution, as confirmed by unity check (UC) analysis, noting potential inaccuracies. Increasing (n) beyond the optimal value led to higher mass due to suboptimal profile assignments, whereas decreasing (n) enhanced standardization but resulted in increased mass due to overdesign. Similar patterns were observed in other typologies, where careful selection of (n) is crucial for mass efficiency. Symmetry and beam continuity generally improved with lower (n) due to increased standardization, whereas connection complexity, dependent on the spatial arrangement of elements and their mass distribution, remained a challenging objective, particularly for higher (n), due to its sensitivity to positional and mass-related constraints.

RQ2: What trade-offs emerge between standardization (e.g. symmetry) and uniqueness (e.g. mass) when different objectives are prioritized?

The study confirmed that prioritizing uniqueness, such as mass minimization, results in configurations with lower standardization, leading to reduced symmetry and increased connection complexity. Conversely, emphasizing standardization through objectives like symmetry produces heavier but more constructible structures. TPE adequately navigated these trade-offs, generating a set of solutions reflecting varying priority settings, limited by scalarization's focus on a single composite score, as visualized in multi-parallel plots (Figure 6.13). These plots illustrated how different weight distributions influence the balance between lightweight, unique designs and standardized, constructible configurations.

RQ3: In which core aspects of steel truss optimization does the Tree-structured Parzen Estimator (TPE) offer advantages over simple heuristics, metaheuristics like NSGA-II, and exhaustive search for identifying high-quality groupings in single and multi-objective scenarios?

In identifying the core aspects of steel truss optimization where TPE offers advantages over simple heuristics, metaheuristics like NSGA-II, and exhaustive search for high-quality groupings in single and multi-objective scenarios, TPE excels in computational efficiency, scalability, and suitability for discrete objectives, particularly due to its sample efficiency, which reduces the number of evaluations needed for expensive structural simulations. Based on the literature review (Chapter 3) and study findings, TPE was selected for its strong handling of discrete variables and weighted scalarization, though limited experimental comparisons with other metaheuristics and untuned hyperparameters constrain claims of broad superiority. Unlike NSGA-II, which literature suggests struggles with discrete, topologically driven objectives like connection degree due to complex parameter tuning, TPE efficiently models hierarchical constraints using Bayesian optimization. Compared to simple heuristics like RandomSampler, which lack guided exploration and perform poorly in high-dimensional spaces, TPE offers broader exploration. Exhaustive search (EXS) was used to validate TPE findings, confirming identical minimal mass at ($n = 3$), but its computational infeasibility for large problems (e.g., 1.95 million configurations for the benchmark with search spaces growing exponentially as shown in Figure 6.10) underscores TPE scalability. Moreover, TPE ease of implementation, facilitated by libraries like Optuna [35], and its ability to handle complex objectives like connection degree dependent on both positional context and weight minimization make it a suitable alternative for steel truss optimization, particularly for discrete multi-objective tasks, though further experimental comparisons with other metaheuristics are needed.

RQ4: In what ways can stakeholder-defined weights, reflecting their objectives or design preferences, influence the resulting optimal truss layouts?

Stakeholder-defined weights significantly shape optimal truss layouts by prioritizing specific objectives. Higher weights on mass yield lighter structures, while weights on symmetry favor standardized, constructible designs. TPE flexibility facilitates alignment with these preferences, producing a range of solutions, though limited by scalarization's focus. However, balancing weights is critical, as highly sensitive objectives like connection degree can dominate optimization when overemphasized. Excessive weighting may constrain the search space, reducing solution diversity and potentially degrading other objectives, such as mass or symmetry, necessitating careful calibration to maintain a comprehensive exploration of feasible configurations.

RQ5: In what ways can visual or computational tools enhance the interpretability of optimized truss designs and assist in navigating complex trade-off spaces?

The research focused on multi-parallel plots as the primary visualization tool to enhance the interpretability of optimized truss designs. Implemented using tools like Plotly [39], these interactive plots enabled visualization of trade-offs across mass, symmetry, connection complexity, and beam continuity for various (n) and weight distributions (Figure 6.14). By enabling filtering based on specific criteria, multi-parallel plots facilitated navigation through complex trade-off spaces, potentially making it easier to identify configurations that meet stakeholder needs, pending practical validation. While other visualization methods were not explored, the effectiveness of multi-parallel plots underscores their value in structural optimization contexts.

Final Remarks:

This research demonstrates the potential of the Tree-structured Parzen Estimator (TPE) as a highly suitable tool for optimizing 2D rectangular steel truss designs, which are widely used in construction for their ability to provide robust structural support with efficient material use. TPE, a Bayesian optimization algorithm, offers a flexible and sample-efficient approach to this challenge, enabling engineers to explore a complex design space effectively. Steel trusses, particularly those with a 2D rectangular configuration, require careful optimization to balance competing objectives such as minimizing weight, maximizing strength, and simplifying fabrication processes. By applying TPE within this scope, the study showcases its ability to identify high-performing truss designs with fewer computational evaluations compared to traditional methods like grid or random searches.

The flexibility of TPE lies in its capacity to adapt to various design variables and constraints, such as selecting from standardized steel sections or managing multi-objective trade-offs. Its sample efficiency is equally critical, as each truss design evaluation often involves time-intensive simulations, such as finite element analysis. By intelligently modeling the probability of improvement, TPE focuses on promising configurations, reducing the computational burden and making optimization practical for real-world applications. The insights gained from this research not only validate TPE effectiveness for 2D rectangular trusses but also establish a foundation for future advancements in structural engineering. These findings could inspire the application of TPE to more intricate truss geometries or other optimization challenges in the field, potentially enhancing design processes across diverse construction projects.

A standout feature of this work is the incorporation of the practical visualization tool, the multi-parallel plot, which significantly enhances the decision-making process. This plot, also known as a parallel coordinates plot, allows stakeholders to visualize high-dimensional data by representing design variables and objectives on parallel axes. For truss optimization, this means stakeholders can explore trade-offs such as mass versus structural complexity objectives like symmetry or connection degree across a range of design options. By interacting with this visualization, they can filter and select configurations that align with specific priorities, whether that’s minimizing structural mass, enhancing aesthetic symmetry, or ensuring compliance with structural standards. This capability potentially empowers stakeholders, from engineers to project managers, to make well-informed design decisions based on a clear and comprehensive understanding of the optimization outcomes, pending practical validation.

In essence, this research positions TPE as a powerful, adaptable tool for optimizing 2D rectangular steel truss designs, delivering a method that is both efficient and versatile. The combination of TPE optimization prowess and intuitive visualization tools like multi-parallel plots paves the way for more effective and collaborative design processes in structural engineering.

References

[1] José Abad González. *Design, analysis and Application of Innovative Connections for Optimized Steel Trusses*. Tech. rep. TU Delft, 2020.

[2] Mena Abdelnour and Volkmar Zabel. "Identification of closely spaced modes in space truss structures as base for the determination of member normal forces". In: *Journal of Physics: Conference Series*. Vol. 2647. 19. Institute of Physics, 2024. DOI: 10.1088/1742-6596/2647/19/192014.

[3] J. S. Arora and A. K. Govils. *An Efficient Method for Optimal Structural Design by Substructuring*. Tech. rep. 1977, pp. 507–515.

[4] Hilton Back et al. *Connections in steel structures*. 2016. ISBN: 9781564240194.

[5] Helio J.C. Barbosa, Afonso C.C. Lemonge, and Carlos C.H. Borges. "A genetic algorithm encoding for cardinality constraints and automatic variable linking in structural optimization". In: *Engineering Structures* 30.12 (Dec. 2008), pp. 3708–3723. ISSN: 01410296. DOI: 10.1016/j.engstruct.2008.06.014.

[6] Lauren L. Beghini et al. "Connecting architecture and engineering through structural topology optimization". In: *Engineering Structures* 59 (Feb. 2014), pp. 716–726. ISSN: 01410296. DOI: 10.1016/j.engstruct.2013.10.032.

[7] J. Bergstra, D. Yamins, and D. Cox. *Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures*. Tech. rep. 2013.

[8] James Bergstra et al. *Algorithms for Hyper-Parameter Optimization*. Tech. rep. 2010.

[9] Michel Bierlaire. *Optimization: Principles and Algorithms*. 2018. ISBN: 978-2-88915-279-7.

[10] Juliana Triches Boscardin, Victor Yepes, and Moacir Kripka. "Optimization of reinforced concrete building frames with automated grouping of columns". In: *Automation in Construction* 104 (Aug. 2019), pp. 331–340. ISSN: 09265805. DOI: 10.1016/j.autcon.2019.04.024.

[11] K. Cerek and J. Grabe. "Numerical simulation and optimization of dike geometry using multi-objective evolutionary algorithm NSGA-II". In: (2023). DOI: 10.53243/NUMGE2023-80. URL: https://doi.org/10.53243/NUMGE2023-80.

[12] W. F. Chen. "Structural engineering: Seeing the big picture". In: *KSCE Journal of Civil Engineering* 12.1 (Jan. 2008), pp. 25–29. ISSN: 1226-7988. DOI: 10.1007/s12205-008-8025-7.

[13] Chris van der Ploeg and Jelle Roks. *Optimized 3D reinforcement in Complex Shapes Using Dynamo*. Tech. rep. ABT bv., 2016. URL: https://nl.linkedin.com/in/jelle-roks-71344a29.

[14] Clemens Preisinger. *Karamba3D – Parametric Structural Engineering*. 2025. URL: https://karamba3d.com/.

[15] Bruno J. Afonso Costa et al. "Modal Analysis for the Rehabilitation Assessment of the Luiz I Bridge". In: *Journal of Bridge Engineering* 19.12 (Dec. 2014). ISSN: 1084-0702. DOI: 10.1061/(asce)be.1943-5592.0000632.

[16] Raffaele Cucuzza et al. "Constructability-based design approach for steel structures: From truss beams to real-world inspired industrial buildings". In: *Automation in Construction* 166 (Oct. 2024). ISSN: 09265805. DOI: 10.1016/j.autcon.2024.105630.

[17] Kalyanmoy D. "An effcient constraint handling method for genetic algorithms". In: (). URL: www.elsevier.com/locate/cma.

[18] *DelftBlue Supercomputer – User Documentation (TU Delft High Performance Computing Center)*. URL: https://doc.dhpc.tudelft.nl/delftblue/.

[19] Dr D. B. Moore, Mr. D. G. Brown, and Dr R. J. Pope. *Handbook of Structural Steelwork Eurocode Edition - TATA STEEL*. 2013. ISBN: 9781850730651. URL: www.steelconstruction.info.

[20] Jeff Erickson. *Algorithms*. 2019. ISBN: 978-1792-64483-2.

[21] *Eurocode 3 - Design of steel structures - Part 1-1: General rules and rules for buildings*. 2020.

[22] Nikolaus Hansen. "The CMA Evolution Strategy: A Tutorial". In: (Apr. 2016). URL: `http://arxiv.org/abs/1604.00772`.

[23] R. Hibbeler and Kai Beng Yap. *Structural Analysis*. Pearson Education, Inc., 2018, p. 720. ISBN: 9780134610672.

[24] Yoshihiro Kanno. "Global optimization of trusses with constraints on number of different cross-sections: a mixed-integer second-order cone programming approach". In: *Computational Optimization and Applications* 63.1 (Jan. 2016), pp. 203–236. ISSN: 15732894. DOI: `10.1007/s10589-015-9766-0`.

[25] Ali Kaveh and Taha Bakhshpoori. *Metaheuristics: Outlines, MATLAB Codes and Examples*. Springer International Publishing, Mar. 2019, pp. 1–190. ISBN: 9783030040673. DOI: `10.1007/978-3-030-04067-3`.

[26] Ali Kaveh, Hossein Rahami, and Iman Shojaei. *Studies in Systems, Decision and Control 290 Swift Analysis of Civil Engineering Structures Using Graph Theory Methods*. Tech. rep. URL: `http://www.springer.com/series/13304`.

[27] S. Kazemzadeh Azad and O. Hasançebi. "Computationally efficient discrete sizing of steel frames via guided stochastic search heuristic". In: *Computers and Structures* 156 (Aug. 2015), pp. 12–28. ISSN: 00457949. DOI: `10.1016/j.compstruc.2015.04.009`.

[28] Bin Li and Honglei Wang. "Multi-objective sparrow search algorithm: A novel algorithm for solving complex multi-objective optimisation problems". In: *Expert Systems with Applications* 210 (Dec. 2022). ISSN: 09574174. DOI: `10.1016/j.eswa.2022.118414`.

[29] Luís Simões da Silva, Rui Simões, and Helena Gervásio. *Eurocode 3: Design of Steel Structures*. Tech. rep. 2013.

[30] Mostafa Mashayekhi, Eysa Salajegheh, and Milad Dehghani. "Topology optimization of double and triple layer grid structures using a modified gravitational harmony search algorithm with efficient member grouping strategy". In: *Computers and Structures* 172 (Aug. 2016), pp. 40–58. ISSN: 00457949. DOI: `10.1016/j.compstruc.2016.05.008`.

[31] Alexandre Mathern et al. "Multi-objective constrained Bayesian optimization for structural design". In: *Structural and Multidisciplinary Optimization* 63.2 (Feb. 2021), pp. 689–701. ISSN: 16151488. DOI: `10.1007/s00158-020-02720-2`.

[32] Linfeng Mei and Qian Wang. *Structural optimization in civil engineering: A literature review*. Feb. 2021. DOI: `10.3390/buildings11020066`.

[33] Hope Kasuka Mwakamui and Keneilwe Ntshwene. "Stakeholders Influence on Construction Project Success". In: *Journal of Civil Engineering and Urbanism* 14.3s (Sept. 2024), pp. 206–211. ISSN: 22520430. DOI: `10.54203/jceu.2024.21`. URL: `https://ojceu.com/main/attachments/article/104/JCEU14(3s)206-211,2024.pdf`.

[34] Masahiro Nomura et al. *Warm Starting CMA-ES for Hyperparameter Optimization*. Tech. rep. 2021. URL: `www.aaai.org`.

[35] Optuna Team. *Optuna Documentation – Samplers*. 2025. URL: `https://optuna.readthedocs.io/en/stable/reference/samplers/index.html`.

[36] Yoshihiko Ozaki et al. "Multiobjective tree-structured parzen estimator for computationally expensive optimization problems". In: *GECCO 2020 - Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. Association for Computing Machinery, June 2020, pp. 533–541. ISBN: 9781450371285. DOI: `10.1145/3377930.3389817`.

[37] Panos Y. Papalambros and Douglass J. Wilde. *Principles of optimal design : modeling and computation*. Cambridge University Press, 2000, p. 390. ISBN: 0521622158.

[38] Bo Peng et al. "Cost-based optimization of steel frame member sizing and connection type using dimension increasing search". In: *Optimization and Engineering* 23.3 (Sept. 2022), pp. 1525–1558. ISSN: 15732924. DOI: `10.1007/s11081-021-09665-5`.

[39] Plotly Technologies Inc. *raphing libraries for making interactive, publication-quality graphs online*. URL: `https://plotly.com/graphing-libraries/`.

[40] Cláudio Resende et al. “Automatic Column Grouping of 3D Steel Frames via Multi-Objective Structural Optimization”. In: *Buildings* 14.1 (Jan. 2024). ISSN: 20755309. DOI: `10.3390/buildings14010191`.

[41] Robert McNeel & Associates. *Grasshopper 3D – Algorithmic Modeling for Rhino*. 2025. URL: `https://www.grasshopper3d.com/`.

[42] Kamal C. Sarma and Hojjat Adeli. “Life-cycle cost optimization of steel structures”. In: *International Journal for Numerical Methods in Engineering* 55.12 (Dec. 2002), pp. 1451–1462. ISSN: 00295981. DOI: `10.1002/nme.549`.

[43] Thornton Tomasetti CORE studio. *Colibri – Design Space Exporter for Grasshopper*. 2025. URL: `https://www.food4rhino.com/en/app/colibri`.

[44] Václav Hlaváˇc, Keith G. Jeffery, and s. *Trends in Theory and Practice of Informatics*. 2000.

[45] Bernard Vaudeville et al. *How Irregular Geometry and Industrial Process Come Together: A Case Study of the ”Fondation Louis Vuitton Pour la Creation”*. Tech. rep. 2013.

[46] Richard Walls and Alex Elvin. “An algorithm for grouping members in a structure”. In: *Engineering Structures* 32.6 (June 2010), pp. 1760–1768. ISSN: 01410296. DOI: `10.1016/j.engstruct.2010.02.027`.

[47] Shuhei Watanabe. “Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance”. In: (Apr. 2023). URL: `http://arxiv.org/abs/2304.11127`.

[48] Tomas R. van Woudenberg. “Buildable Design in Optimisation of Steel Skeletal Structures: A Comparison of Existing and New Methods for Finding the Best Solution with Low Diversity”. PhD thesis. TU Delft, 2020. DOI: `10.4121/uuid:4e32b29f-6647-4a36-9ea1-8931c88f8864`. URL: `http://doi.org/10.4121/uuid:4e32b29f-6647-4a36-9ea1-8931c88f8864`.

[49] Tomas R. van Woudenberg and Frans P. van der Meer. “A grouping method for optimization of steel skeletal structures by applying a combinatorial search algorithm based on a fully stressed design”. In: *Engineering Structures* 249 (Dec. 2021). ISSN: 18737323. DOI: `10.1016/j.engstruct.2021.113299`.

[50] Gustavo R. Zavala et al. *A survey of multi-objective metaheuristics applied to structural optimization*. 2014. DOI: `10.1007/s00158-013-0996-4`.

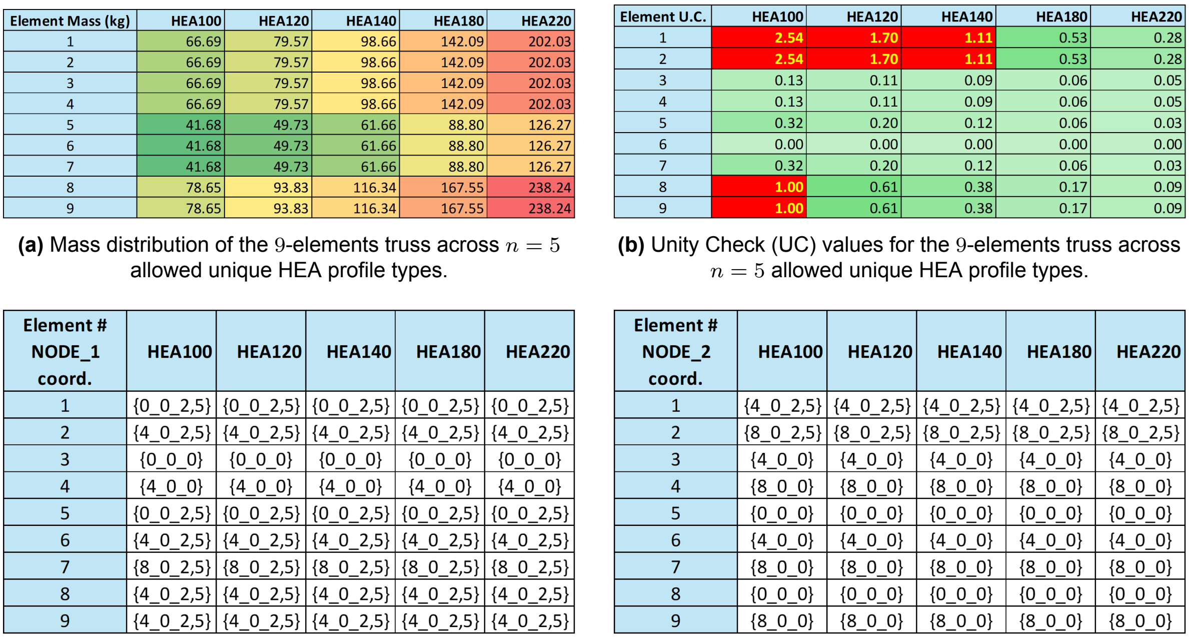
A

Supplementary data for truss configurations

This appendix presents comprehensive numerical and visual data supporting the optimization analyses of steel truss configurations in the thesis. It includes Excel tables detailing mass, Utilization Check (UC), and node coordinates (Section A.1). Additionally, Grasshopper-generated visualizations of deflection and axial force distribution for the benchmark and additional truss topologies are used to test the Python model (Section A.2). These supplementary problems feature varied element counts (k) and (n) allowed unique HEA profile configurations to evaluate the model's robustness. A reference figure from external literature illustrating optimization trends is also provided (Section A.3).

A.1. Numerical data for truss configurations

A.1.1. Benchmark model (9-elements truss, n =5 HEA profiles)



Element Mass (kg)	HEA100	HEA120	HEA140	HEA180	HEA220
1	66.69	79.57	98.66	142.09	202.03
2	66.69	79.57	98.66	142.09	202.03
3	66.69	79.57	98.66	142.09	202.03
4	66.69	79.57	98.66	142.09	202.03
5	41.68	49.73	61.66	88.80	126.27
6	41.68	49.73	61.66	88.80	126.27
7	41.68	49.73	61.66	88.80	126.27
8	78.65	93.83	116.34	167.55	238.24
9	78.65	93.83	116.34	167.55	238.24

(a) Mass distribution of the 9-elements truss across $n = 5$ allowed unique HEA profile types.

Element U.C.	HEA100	HEA120	HEA140	HEA180	HEA220
1	2.54	1.70	1.11	0.53	0.28
2	2.54	1.70	1.11	0.53	0.28
3	0.13	0.11	0.09	0.06	0.05
4	0.13	0.11	0.09	0.06	0.05
5	0.32	0.20	0.12	0.06	0.03
6	0.00	0.00	0.00	0.00	0.00
7	0.32	0.20	0.12	0.06	0.03
8	1.00	0.61	0.38	0.17	0.09
9	1.00	0.61	0.38	0.17	0.09

(b) Unity Check (UC) values for the 9-elements truss across $n = 5$ allowed unique HEA profile types.

Element # NODE_1 coord.	HEA100	HEA120	HEA140	HEA180	HEA220
1	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}
2	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}
3	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}
4	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}
5	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}
6	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}
7	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}
8	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}
9	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}

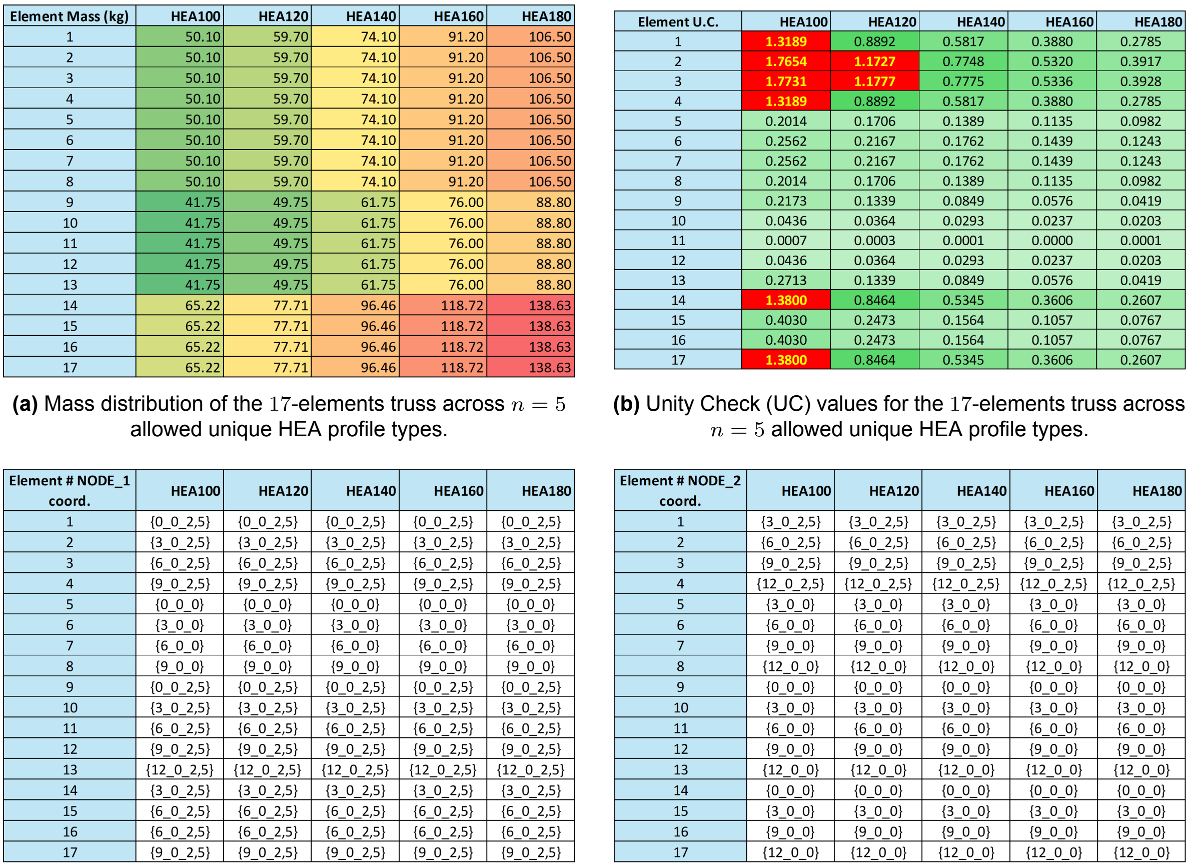
(c) Coordinates of the first node for each element of the 9-elements truss.

Element # NODE_2 coord.	HEA100	HEA120	HEA140	HEA180	HEA220
1	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}
2	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}
3	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}
4	{8_0_0}	{8_0_0}	{8_0_0}	{8_0_0}	{8_0_0}
5	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}
6	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}
7	{8_0_0}	{8_0_0}	{8_0_0}	{8_0_0}	{8_0_0}
8	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}
9	{8_0_0}	{8_0_0}	{8_0_0}	{8_0_0}	{8_0_0}

(d) Coordinates of the second node for each element of the 9-elements truss.

Figure A.1: Excel tables for a parametric truss with 9-elements and $n = 5$ allowed unique HEA profile types.

A.1.2. Typology 1 (17-elements truss, n =5 HEA profiles)



Element Mass (kg)	HEA100	HEA120	HEA140	HEA160	HEA180
1	50.10	59.70	74.10	91.20	106.50
2	50.10	59.70	74.10	91.20	106.50
3	50.10	59.70	74.10	91.20	106.50
4	50.10	59.70	74.10	91.20	106.50
5	50.10	59.70	74.10	91.20	106.50
6	50.10	59.70	74.10	91.20	106.50
7	50.10	59.70	74.10	91.20	106.50
8	50.10	59.70	74.10	91.20	106.50
9	41.75	49.75	61.75	76.00	88.80
10	41.75	49.75	61.75	76.00	88.80
11	41.75	49.75	61.75	76.00	88.80
12	41.75	49.75	61.75	76.00	88.80
13	41.75	49.75	61.75	76.00	88.80
14	65.22	77.71	96.46	118.72	138.63
15	65.22	77.71	96.46	118.72	138.63
16	65.22	77.71	96.46	118.72	138.63
17	65.22	77.71	96.46	118.72	138.63

(a) Mass distribution of the 17-elements truss across $n = 5$ allowed unique HEA profile types.

Element U.C.	HEA100	HEA120	HEA140	HEA160	HEA180
1	1.3189	0.8892	0.5817	0.3880	0.2785
2	1.7654	1.1727	0.7748	0.5320	0.3917
3	1.7731	1.1777	0.7775	0.5336	0.3928
4	1.3189	0.8892	0.5817	0.3880	0.2785
5	0.2014	0.1706	0.1389	0.1135	0.0982
6	0.2562	0.2167	0.1762	0.1439	0.1243
7	0.2562	0.2167	0.1762	0.1439	0.1243
8	0.2014	0.1706	0.1389	0.1135	0.0982
9	0.2173	0.1339	0.0849	0.0576	0.0419
10	0.0436	0.0364	0.0293	0.0237	0.0203
11	0.0007	0.0003	0.0001	0.0000	0.0001
12	0.0436	0.0364	0.0293	0.0237	0.0203
13	0.2713	0.1339	0.0849	0.0576	0.0419
14	1.3800	0.8464	0.5345	0.3606	0.2607
15	0.4030	0.2473	0.1564	0.1057	0.0767
16	0.4030	0.2473	0.1564	0.1057	0.0767
17	1.3800	0.8464	0.5345	0.3606	0.2607

(b) Unity Check (UC) values for the 17-elements truss across $n = 5$ allowed unique HEA profile types.

Element # NODE_1 coord.	HEA100	HEA120	HEA140	HEA160	HEA180
1	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}
2	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}
3	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}
4	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}
5	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}
6	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}
7	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}
8	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}
9	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}
10	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}
11	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}
12	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}
13	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}
14	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}
15	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}
16	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}
17	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}

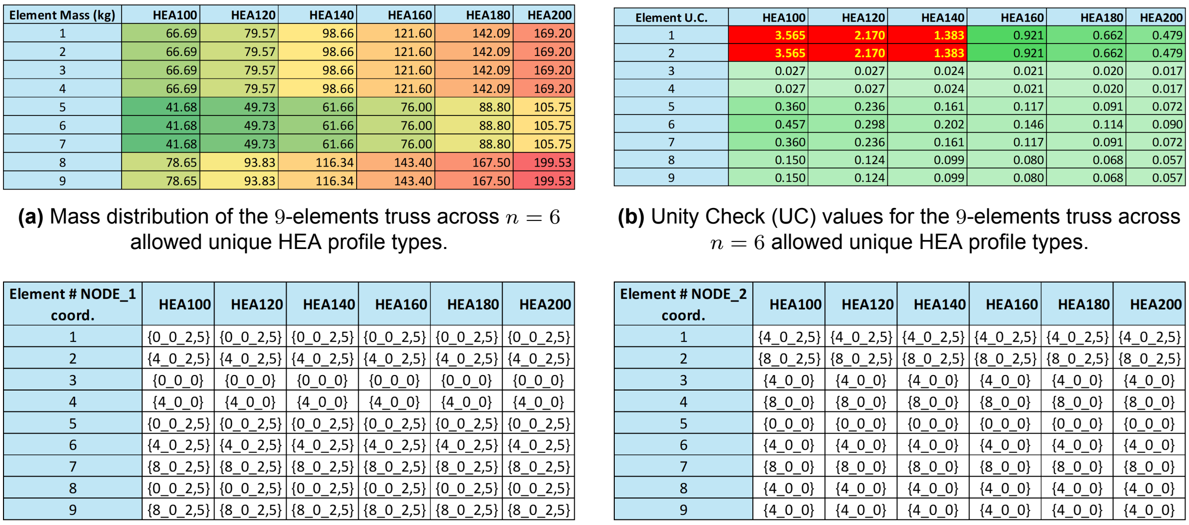
(c) Coordinates of the first node for each element of the 17-elements truss.

Element # NODE_2 coord.	HEA100	HEA120	HEA140	HEA160	HEA180
1	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}
2	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}
3	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}
4	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}
5	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}
6	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}
7	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}
8	{12_0_0}	{12_0_0}	{12_0_0}	{12_0_0}	{12_0_0}
9	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}
10	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}
11	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}
12	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}
13	{12_0_0}	{12_0_0}	{12_0_0}	{12_0_0}	{12_0_0}
14	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}
15	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}
16	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}
17	{12_0_0}	{12_0_0}	{12_0_0}	{12_0_0}	{12_0_0}

(d) Coordinates of the second node for each element of the 17-elements truss.

Figure A.2: Excel tables for a parametric truss with 17-elements and $n = 5$ allowed unique HEA profile types.

A.1.3. Typology 2 (9-elements truss, n =6 HEA profiles)



Element Mass (kg)	HEA100	HEA120	HEA140	HEA160	HEA180	HEA200
1	66.69	79.57	98.66	121.60	142.09	169.20
2	66.69	79.57	98.66	121.60	142.09	169.20
3	66.69	79.57	98.66	121.60	142.09	169.20
4	66.69	79.57	98.66	121.60	142.09	169.20
5	41.68	49.73	61.66	76.00	88.80	105.75
6	41.68	49.73	61.66	76.00	88.80	105.75
7	41.68	49.73	61.66	76.00	88.80	105.75
8	78.65	93.83	116.34	143.40	167.50	199.53
9	78.65	93.83	116.34	143.40	167.50	199.53

(a) Mass distribution of the 9-elements truss across $n = 6$ allowed unique HEA profile types.

Element U.C.	HEA100	HEA120	HEA140	HEA160	HEA180	HEA200
1	3.565	2.170	1.383	0.921	0.662	0.479
2	3.565	2.170	1.383	0.921	0.662	0.479
3	0.027	0.027	0.024	0.021	0.020	0.017
4	0.027	0.027	0.024	0.021	0.020	0.017
5	0.360	0.236	0.161	0.117	0.091	0.072
6	0.457	0.298	0.202	0.146	0.114	0.090
7	0.360	0.236	0.161	0.117	0.091	0.072
8	0.150	0.124	0.099	0.080	0.068	0.057
9	0.150	0.124	0.099	0.080	0.068	0.057

(b) Unity Check (UC) values for the 9-elements truss across $n = 6$ allowed unique HEA profile types.

Element # NODE_1 coord.	HEA100	HEA120	HEA140	HEA160	HEA180	HEA200
1	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}
2	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}
3	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}
4	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}
5	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}
6	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}
7	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}
8	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}
9	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}

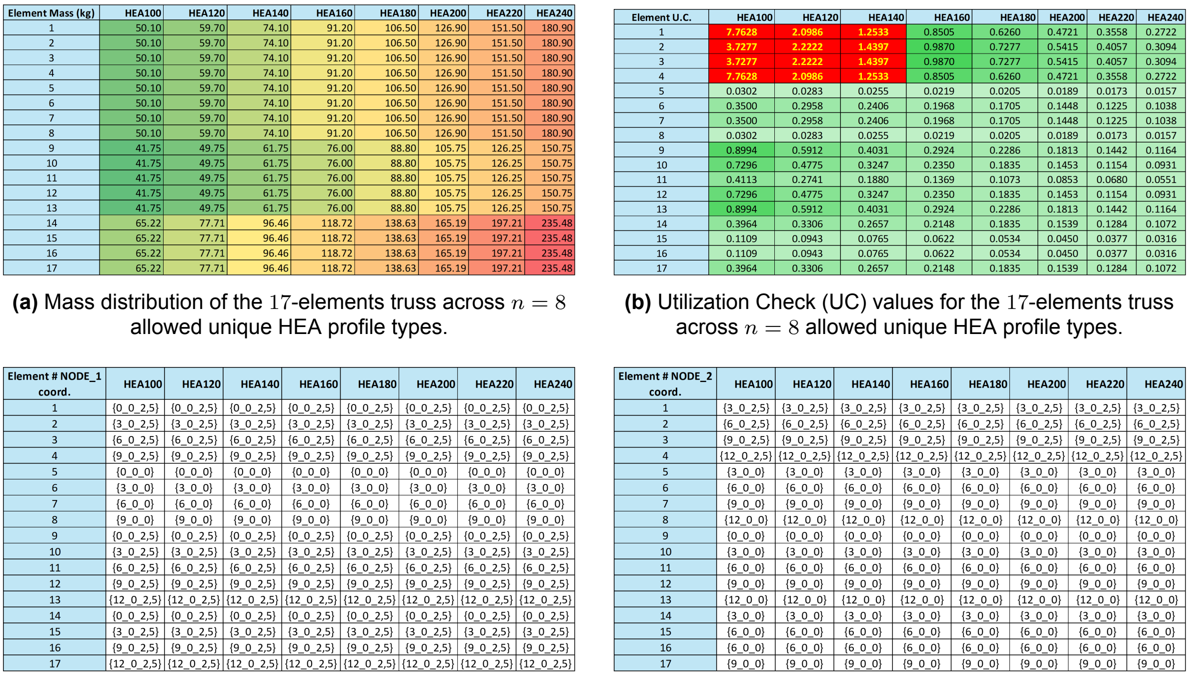
(c) Coordinates of the first node for each element of the 9-elements truss.

Element # NODE_2 coord.	HEA100	HEA120	HEA140	HEA160	HEA180	HEA200
1	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}	{4_0_2,5}
2	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}	{8_0_2,5}
3	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}
4	{8_0_0}	{8_0_0}	{8_0_0}	{8_0_0}	{8_0_0}	{8_0_0}
5	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}
6	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}
7	{8_0_0}	{8_0_0}	{8_0_0}	{8_0_0}	{8_0_0}	{8_0_0}
8	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}
9	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}	{4_0_0}

(d) Coordinates of the second node for each element of the 9-elements truss.

Figure A.3: Excel tables for a parametric truss with 9-elements and $n = 6$ allowed unique HEA profile types.

A.1.4. Typology 3 (17-elements truss, n =8 HEA profiles)



Element Mass (kg)	HEA100	HEA120	HEA140	HEA160	HEA180	HEA200	HEA220	HEA240
1	50.10	59.70	74.10	91.20	106.50	126.90	151.50	180.90
2	50.10	59.70	74.10	91.20	106.50	126.90	151.50	180.90
3	50.10	59.70	74.10	91.20	106.50	126.90	151.50	180.90
4	50.10	59.70	74.10	91.20	106.50	126.90	151.50	180.90
5	50.10	59.70	74.10	91.20	106.50	126.90	151.50	180.90
6	50.10	59.70	74.10	91.20	106.50	126.90	151.50	180.90
7	50.10	59.70	74.10	91.20	106.50	126.90	151.50	180.90
8	50.10	59.70	74.10	91.20	106.50	126.90	151.50	180.90
9	41.75	49.75	61.75	76.00	88.80	105.75	126.25	150.75
10	41.75	49.75	61.75	76.00	88.80	105.75	126.25	150.75
11	41.75	49.75	61.75	76.00	88.80	105.75	126.25	150.75
12	41.75	49.75	61.75	76.00	88.80	105.75	126.25	150.75
13	41.75	49.75	61.75	76.00	88.80	105.75	126.25	150.75
14	65.22	77.71	96.46	118.72	138.63	165.19	197.21	235.48
15	65.22	77.71	96.46	118.72	138.63	165.19	197.21	235.48
16	65.22	77.71	96.46	118.72	138.63	165.19	197.21	235.48
17	65.22	77.71	96.46	118.72	138.63	165.19	197.21	235.48

(a) Mass distribution of the 17-elements truss across $n = 8$ allowed unique HEA profile types.

Element U.C.	HEA100	HEA120	HEA140	HEA160	HEA180	HEA200	HEA220	HEA240
1	7.7628	2.0986	1.2533	0.8505	0.6260	0.4721	0.3558	0.2722
2	3.7277	2.2222	1.4397	0.9870	0.7277	0.5415	0.4057	0.3094
3	3.7277	2.2222	1.4397	0.9870	0.7277	0.5415	0.4057	0.3094
4	7.7628	2.0986	1.2533	0.8505	0.6260	0.4721	0.3558	0.2722
5	0.0302	0.0283	0.0255	0.0219	0.0205	0.0189	0.0173	0.0157
6	0.3500	0.2958	0.2406	0.1968	0.1705	0.1448	0.1225	0.1038
7	0.3500	0.2958	0.2406	0.1968	0.1705	0.1448	0.1225	0.1038
8	0.0302	0.0283	0.0255	0.0219	0.0205	0.0189	0.0173	0.0157
9	0.8994	0.5912	0.4031	0.2924	0.2286	0.1813	0.1442	0.1164
10	0.7296	0.4775	0.3247	0.2350	0.1835	0.1453	0.1154	0.0931
11	0.4113	0.2741	0.1880	0.1369	0.1073	0.0853	0.0680	0.0551
12	0.7296	0.4775	0.3247	0.2350	0.1835	0.1453	0.1154	0.0931
13	0.8994	0.5912	0.4031	0.2924	0.2286	0.1813	0.1442	0.1164
14	0.3964	0.3306	0.2657	0.2148	0.1835	0.1539	0.1284	0.1072
15	0.1109	0.0943	0.0765	0.0622	0.0534	0.0450	0.0377	0.0316
16	0.1109	0.0943	0.0765	0.0622	0.0534	0.0450	0.0377	0.0316
17	0.3964	0.3306	0.2657	0.2148	0.1835	0.1539	0.1284	0.1072

(b) Utilization Check (UC) values for the 17-elements truss across $n = 8$ allowed unique HEA profile types.

Element # NODE_1 coord.	HEA100	HEA120	HEA140	HEA160	HEA180	HEA200	HEA220	HEA240
1	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}
2	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}
3	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}
4	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}
5	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}
6	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}
7	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}
8	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}
9	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}
10	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}
11	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}
12	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}
13	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}
14	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}	{0_0_2,5}
15	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}
16	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}
17	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}

(c) Coordinates of the first node for each element of the 17-elements truss.

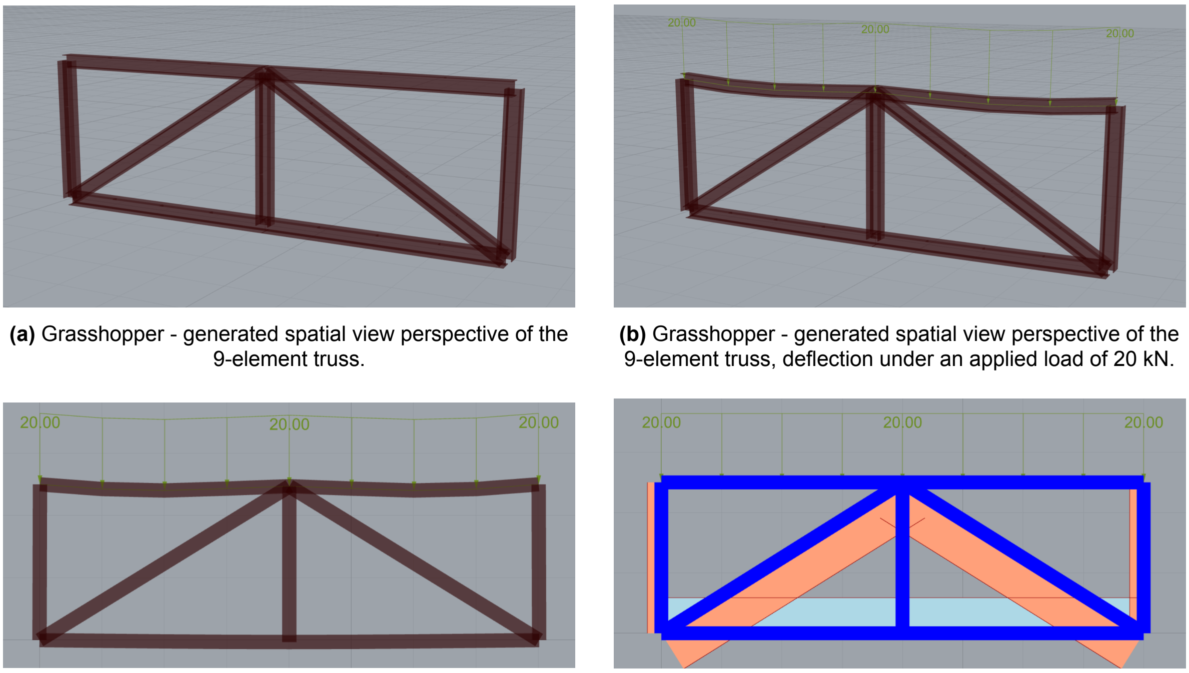
Element # NODE_2 coord.	HEA100	HEA120	HEA140	HEA160	HEA180	HEA200	HEA220	HEA240
1	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}	{3_0_2,5}
2	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}	{6_0_2,5}
3	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}	{9_0_2,5}
4	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}	{12_0_2,5}
5	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}
6	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}
7	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}
8	{12_0_0}	{12_0_0}	{12_0_0}	{12_0_0}	{12_0_0}	{12_0_0}	{12_0_0}	{12_0_0}
9	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}	{0_0_0}
10	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}
11	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}
12	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}
13	{12_0_0}	{12_0_0}	{12_0_0}	{12_0_0}	{12_0_0}	{12_0_0}	{12_0_0}	{12_0_0}
14	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}	{3_0_0}
15	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}
16	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}	{6_0_0}
17	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}	{9_0_0}

(d) Coordinates of the second node for each element of the 17-elements truss.

Figure A.4: Excel tables for a parametric truss with 17-elements and $n = 8$ allowed unique HEA profile types.

A.2. Grasshopper visualizations of truss configurations

A.2.1. Benchmark model (9-elements truss, n = 5 HEA profiles, 20 kN load)



(a) Grasshopper - generated spatial view perspective of the 9-element truss.

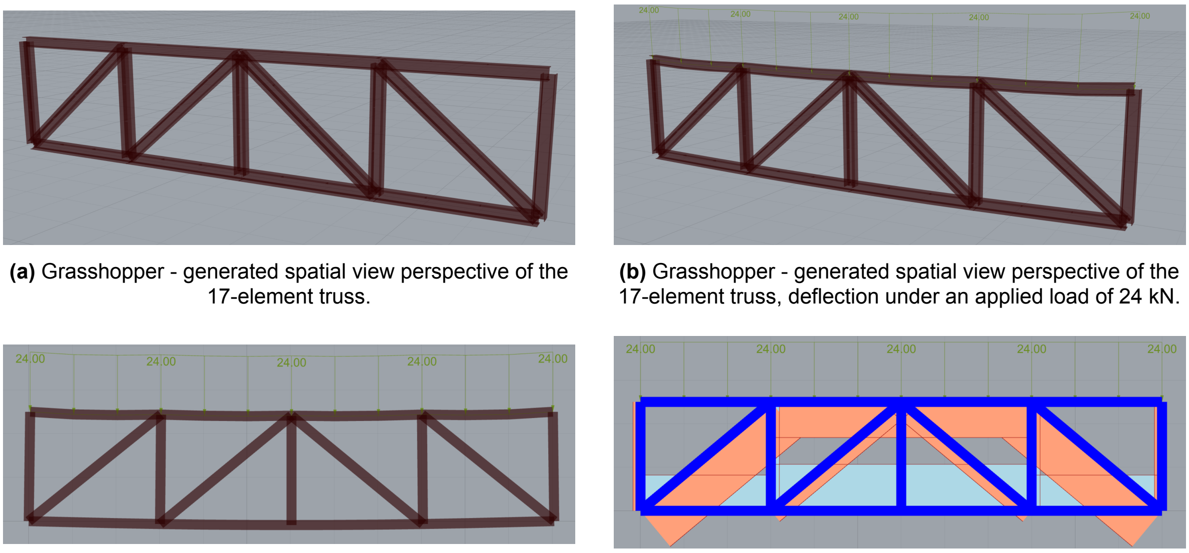
(b) Grasshopper - generated spatial view perspective of the 9-element truss, deflection under an applied load of 20 kN.

(c) Grasshopper - generated front view perspective of the 9-element truss under a 20kN load, illustrating structural deflection.

(d) Grasshopper - generated front view perspective of the 9-element truss under a 20kN load, illustrating axial force distribution.

Figure A.5: Benchmark problem Typology 1: Grasshopper - generated parametric truss with 9 elements, shown under a 20kN load with visualizations of deflection and axial force distribution from multiple perspectives.

A.2.2. Typology 1 (17-elements truss, n =5 HEA profiles, 24 kN load)



(a) Grasshopper - generated spatial view perspective of the 17-element truss.

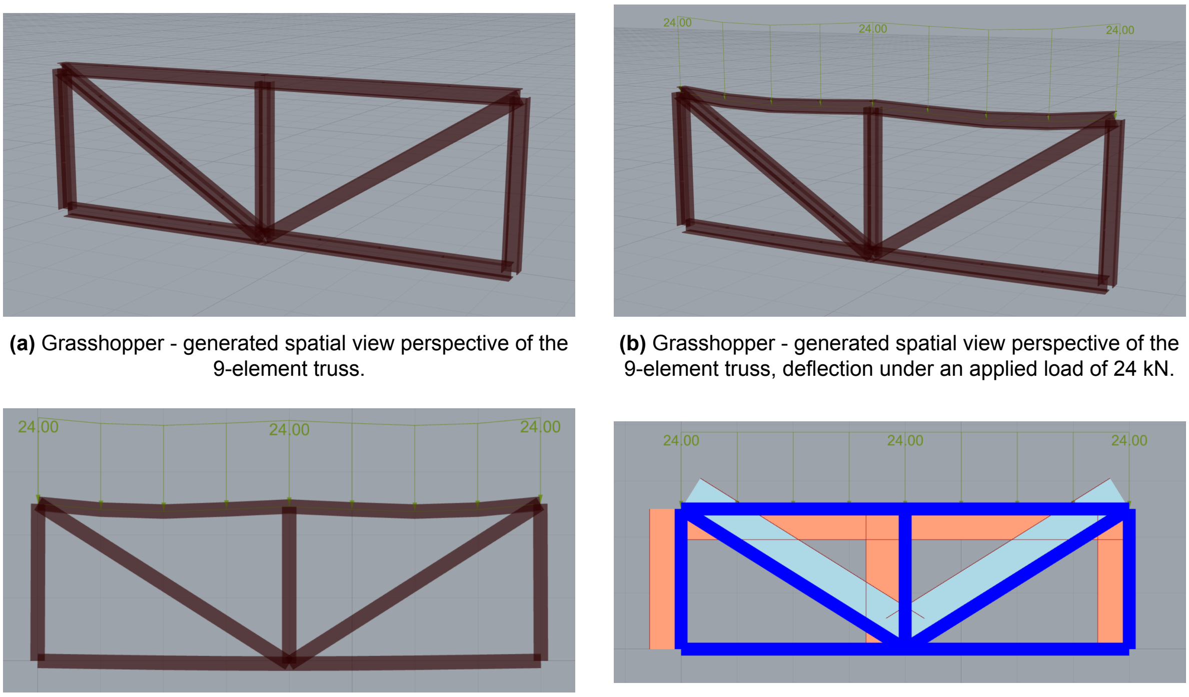
(b) Grasshopper - generated spatial view perspective of the 17-element truss, deflection under an applied load of 24 kN.

(c) Grasshopper - generated front view perspective of the 17-element truss under a 24kN load, illustrating structural deflection.

(d) Grasshopper - generated front view perspective of the 17-element truss under a 24kN load, illustrating axial force distribution.

Figure A.6: Typology 1 problem: Grasshopper - generated parametric truss with 17 elements, shown under a 24kN load with visualizations of deflection and axial force distribution from multiple perspectives.

A.2.3. Typology 2 (9-elements truss, n =6 HEA profiles, 24 kN load)



(a) Grasshopper - generated spatial view perspective of the 9-element truss.

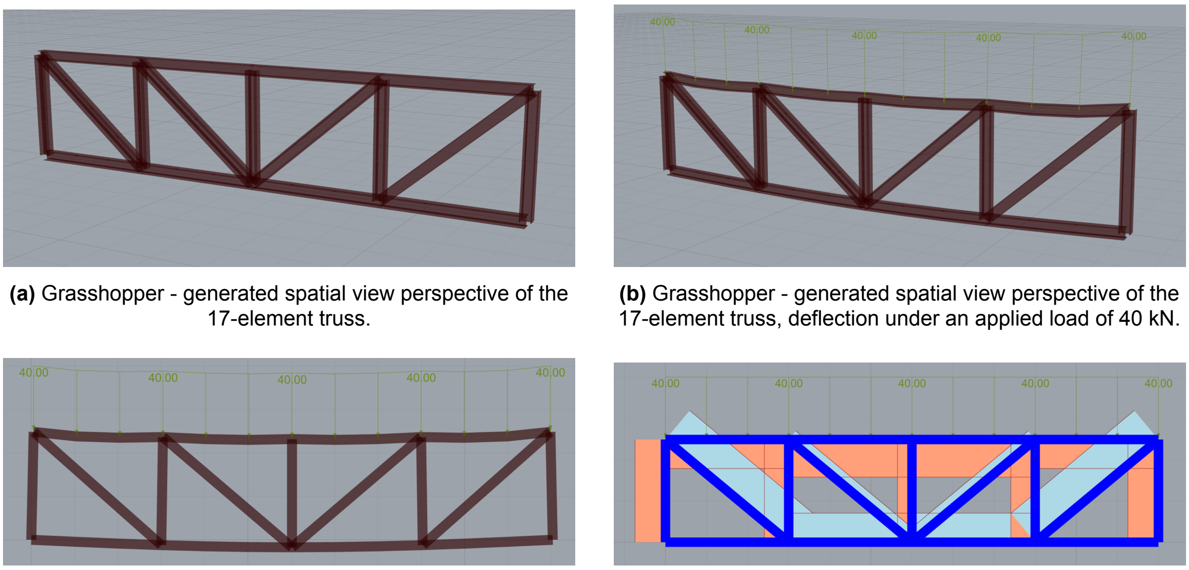
(b) Grasshopper - generated spatial view perspective of the 9-element truss, deflection under an applied load of 24 kN.

(c) Grasshopper - generated front view perspective of the 9-element truss under a 24kN load, illustrating structural deflection.

(d) Grasshopper - generated front view perspective of the 9-element truss under a 24kN load, illustrating axial force distribution.

Figure A.7: Typology 2 problem: Grasshopper - generated parametric truss with 9 elements, shown under a 24kN load with visualizations of deflection and axial force distribution from multiple perspectives.

A.2.4. Typology 3 (17-elements truss, n =8 HEA profiles, 40 kN load)



(a) Grasshopper - generated spatial view perspective of the 17-element truss.

(b) Grasshopper - generated spatial view perspective of the 17-element truss, deflection under an applied load of 40 kN.

(c) Grasshopper - generated front view perspective of the 17-element truss under a 40kN load, illustrating structural deflection.

(d) Grasshopper - generated front view perspective of the 17-element truss under a 40kN load, illustrating axial force distribution.

Figure A.8: Typology 3 problem: Grasshopper - generated parametric truss with 17 elements, shown under a 40kN load with visualizations of deflection and axial force distribution from multiple perspectives.

A.3. Example for reinforcement optimization

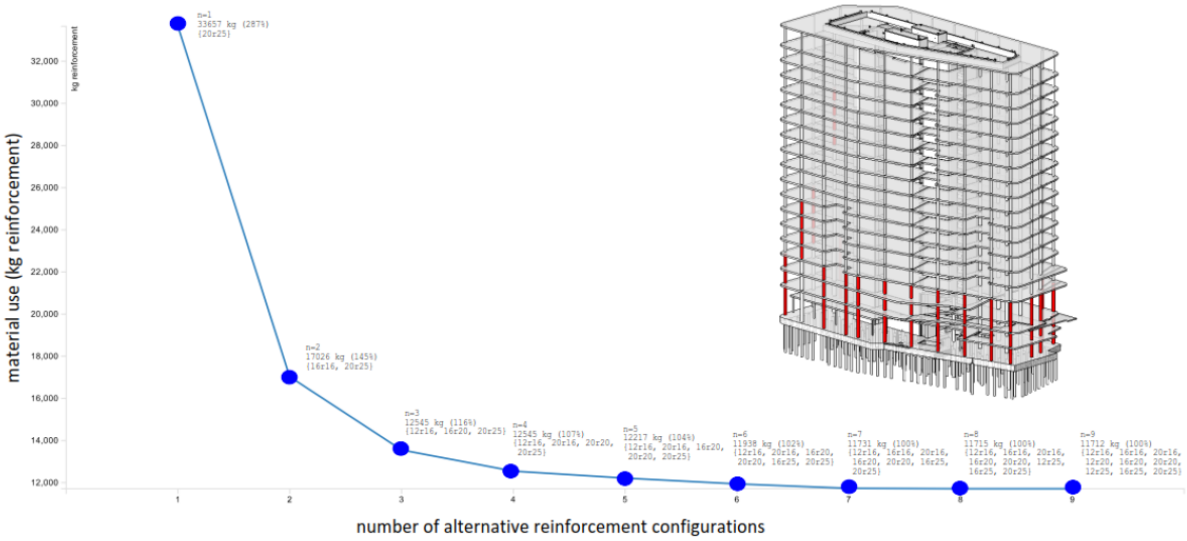


Figure A.9: Influence of applied number of alternative reinforcement configurations on matreial use [13].

B

Grasshopper model

This appendix presents a visual summary of the Grasshopper-based parametric model developed for this research. As described in Chapter 4 (Methodology), the structural behavior of steel trusses was modeled in the Grasshopper environment using the Karamba3D and Colibi plugins for structural analysis. This setup allowed for flexible control over geometry, load application, material assignment, and profile configuration.

The figures below illustrate the various components of the model, including the definition of truss geometry and boundary conditions (Figure B.1), the assignment of HEA steel profiles and material properties (Figure B.2), and the use of the Colibri plugin for exporting configuration data (Figure B.3). These visualizations support the description in the main text and provide transparency on how the analysis inputs were structured and automated within the Grasshopper environment.

While only a representative figure was included in the methodology chapter, the full set of Grasshopper model views is documented here to aid reproducibility and technical clarity.

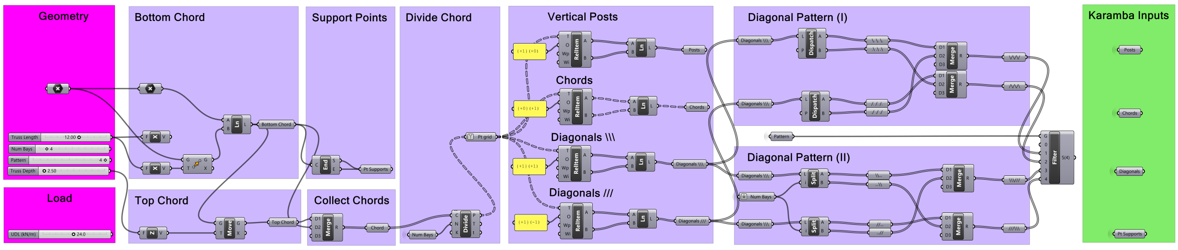
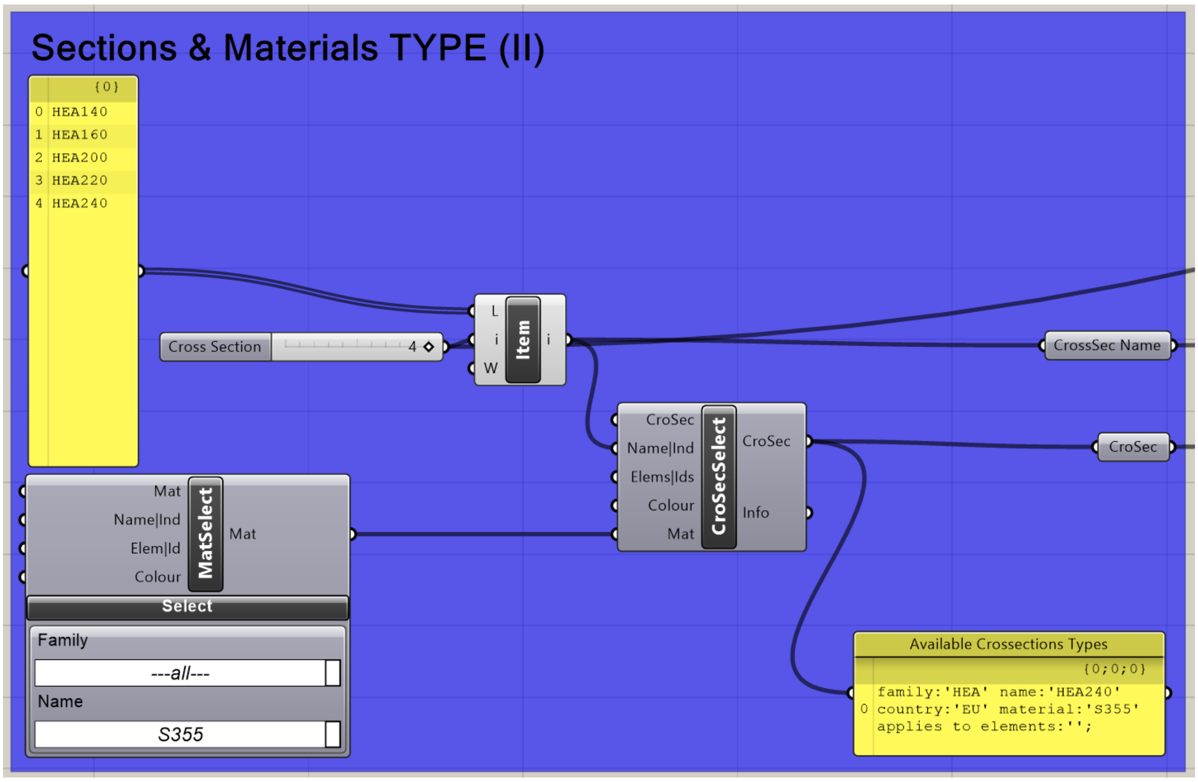


Figure B.1: Grasshopper setup defining geometry and load input for parametric truss analysis, input into Karamba for structural analysis.

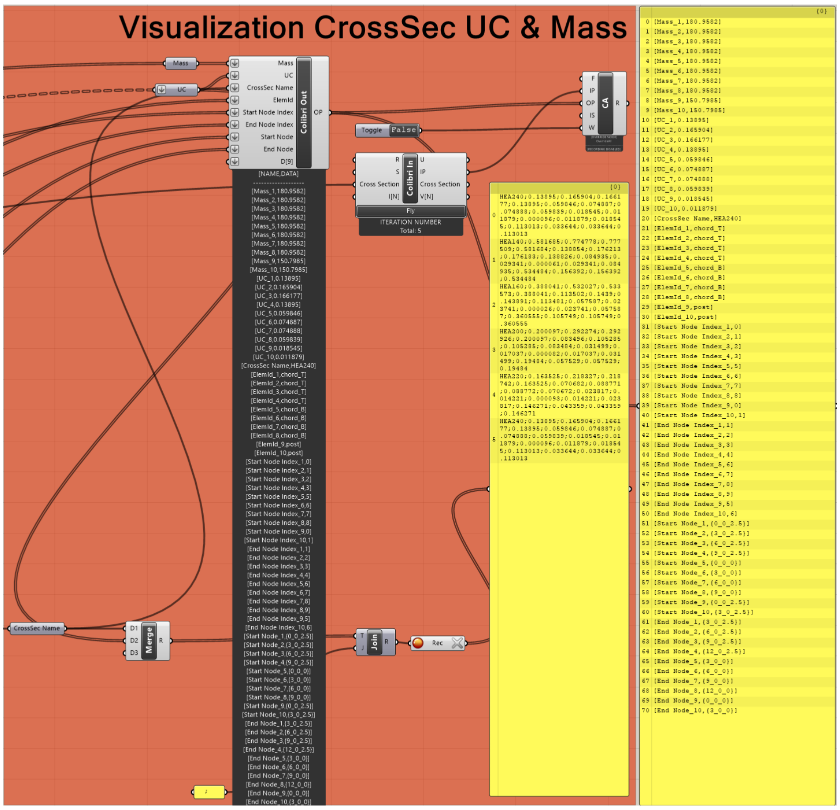


```
{0}
0 HEA140
1 HEA160
2 HEA200
3 HEA220
4 HEA240
```

Available Crossections Types

```
{0;0;0}
0 family:'HEA' name:'HEA240'
  country:'EU' material:'S355'
  applies to elements:'';
```

Figure B.2: Grasshopper setup defining HEA profiles set (HEA140, HEA160, HEA200, HEA220, HEA240) and steel material S355 input for parametric truss analysis across $n = 1$ to $n = 5$.



```
{0}
0 HEA240;0.13895;0.165904;0.166177;0.13895;0.059846;0.074887;0.074888;0.059839;0.018545;0.011879;0.000096;0.011879;0.018545;0.113013;0.033644;0.033644;0.113013
1 HEA140;0.581685;0.774778;0.777509;0.581684;0.138854;0.176213;0.176183;0.138826;0.084935;0.029341;0.000061;0.029341;0.084935;0.534484;0.156392;0.156392;0.534484
2 HEA160;0.388041;0.532027;0.533573;0.388041;0.113502;0.1439;0.143891;0.113481;0.057587;0.023741;0.000026;0.023741;0.057587;0.360555;0.105749;0.105749;0.360555
3 HEA200;0.200097;0.292274;0.292926;0.200097;0.083496;0.105285;0.105285;0.083484;0.031499;0.017037;0.000082;0.017037;0.031499;0.19484;0.057529;0.057529;0.19484
4 HEA220;0.163525;0.218327;0.218742;0.163525;0.070682;0.088771;0.088772;0.070672;0.023817;0.014221;0.000093;0.014221;0.023817;0.146271;0.043359;0.043359;0.146271
5 HEA240;0.13895;0.165904;0.166177;0.13895;0.059846;0.074887;0.074888;0.059839;0.018545;0.011879;0.000096;0.011879;0.018545;0.113013;0.033644;0.033644;0.113013
```

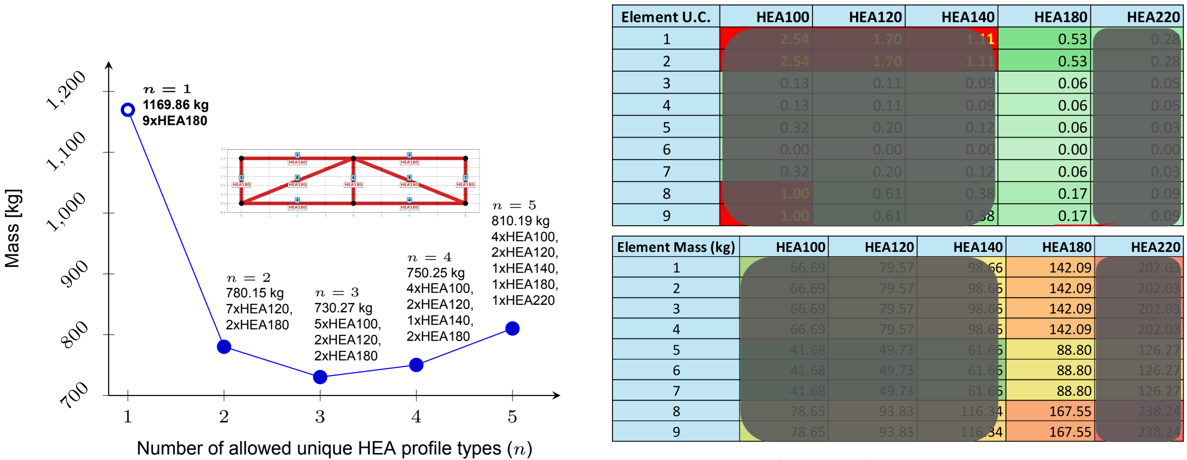
```
{0}
0 [Mass_1,180.9582]
1 [Mass_2,180.9582]
2 [Mass_3,180.9582]
3 [Mass_4,180.9582]
4 [Mass_5,180.9582]
5 [Mass_6,180.9582]
6 [Mass_7,180.9582]
7 [Mass_8,180.9582]
8 [Mass_9,150.7985]
9 [Mass_10,150.7985]
10 [UC_1,0.13895]
11 [UC_2,0.165904]
12 [UC_3,0.166177]
13 [UC_4,0.13895]
14 [UC_5,0.059846]
15 [UC_6,0.074887]
16 [UC_7,0.074888]
17 [UC_8,0.059839]
18 [UC_9,0.018545]
19 [UC_10,0.011879]
20 [CrossSec Name,HEA240]
21 [ElemId_1,chord_T]
22 [ElemId_2,chord_T]
23 [ElemId_3,chord_T]
24 [ElemId_4,chord_T]
25 [ElemId_5,chord_B]
26 [ElemId_6,chord_B]
27 [ElemId_7,chord_B]
28 [ElemId_8,chord_B]
29 [ElemId_9,post]
30 [ElemId_10,post]
31 [Start Node Index_1,0]
32 [Start Node Index_2,1]
33 [Start Node Index_3,2]
34 [Start Node Index_4,3]
35 [Start Node Index_5,5]
36 [Start Node Index_6,6]
37 [Start Node Index_7,7]
38 [Start Node Index_8,8]
39 [Start Node Index_9,0]
40 [Start Node Index_10,1]
41 [End Node Index_1,1]
42 [End Node Index_2,2]
43 [End Node Index_3,3]
44 [End Node Index_4,4]
45 [End Node Index_5,6]
46 [End Node Index_6,7]
47 [End Node Index_7,8]
48 [End Node Index_8,9]
49 [End Node Index_9,5]
50 [End Node Index_10,6]
51 [Start Node_1,{0_0_2.5}]
52 [Start Node_2,{3_0_2.5}]
53 [Start Node_3,{6_0_2.5}]
54 [Start Node_4,{9_0_2.5}]
55 [Start Node_5,{0_0_0}]
56 [Start Node_6,{3_0_0}]
57 [Start Node_7,{6_0_0}]
58 [Start Node_8,{9_0_0}]
59 [Start Node_9,{0_0_2.5}]
60 [Start Node_10,{3_0_2.5}]
61 [End Node_1,{3_0_2.5}]
62 [End Node_2,{6_0_2.5}]
63 [End Node_3,{9_0_2.5}]
64 [End Node_4,{12_0_2.5}]
65 [End Node_5,{3_0_0}]
66 [End Node_6,{6_0_0}]
67 [End Node_7,{9_0_0}]
68 [End Node_8,{12_0_0}]
69 [End Node_9,{0_0_0}]
70 [End Node_10,{3_0_0}]
```

Figure B.3: Grasshopper setup using Colibri for iterating and conducting multiple structural analyses.

C

Grouping configurations for truss optimization

This appendix provides a detailed representation of the element groupings and corresponding truss configurations for the rest values of (n), derived using the same procedure as outlined for $(n = 3)$ in Chapter 4. The methodology involves analyzing (UC) values to form optimal groups of HEA profiles, minimizing overdesign while ensuring structural safety with (UC ≤ 1).



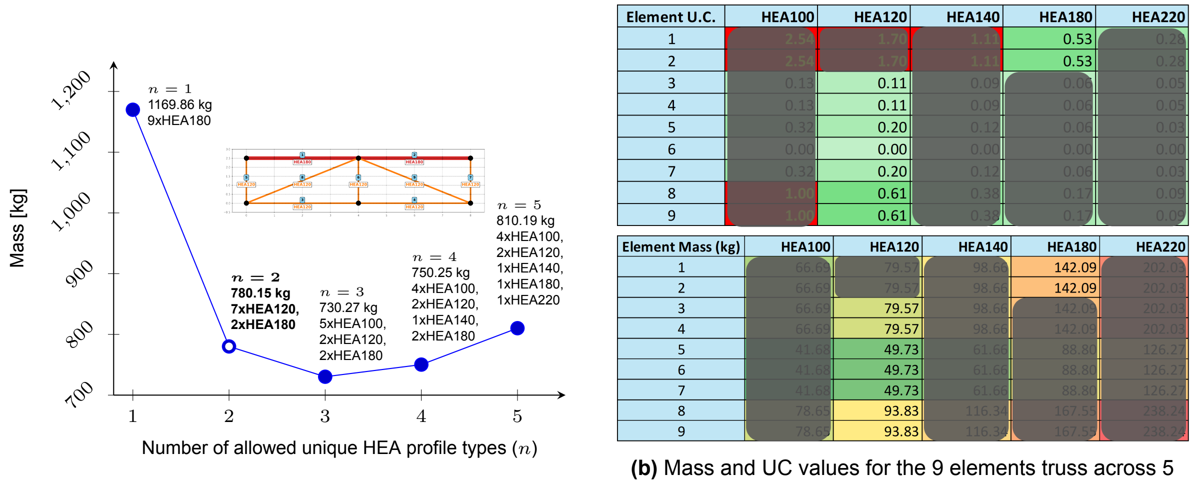
Element U.C.	HEA100	HEA120	HEA140	HEA180	HEA220
1	2.54	1.70	1.11	0.53	0.28
2	2.54	1.70	1.11	0.53	0.28
3	0.13	0.11	0.09	0.06	0.05
4	0.13	0.11	0.09	0.06	0.05
5	0.32	0.20	0.12	0.06	0.03
6	0.00	0.00	0.00	0.00	0.00
7	0.32	0.20	0.12	0.06	0.03
8	1.00	0.61	0.38	0.17	0.09
9	1.00	0.61	0.38	0.17	0.09

Element Mass (kg)	HEA100	HEA120	HEA140	HEA180	HEA220
1	66.69	79.57	98.66	142.09	202.03
2	66.69	79.57	98.66	142.09	202.03
3	66.69	79.57	98.66	142.09	202.03
4	66.69	79.57	98.66	142.09	202.03
5	41.68	49.73	61.66	88.80	126.27
6	41.68	49.73	61.66	88.80	126.27
7	41.68	49.73	61.66	88.80	126.27
8	78.65	93.83	116.34	167.55	238.24
9	78.65	93.83	116.34	167.55	238.24

(a) TPE-Based Optimization of Mass vs n

(b) Mass and UC values for the 9 elements truss across 5 allowed unique HEA profiles $(n = 1)$ *(highlighted)* grouping

Figure C.1: Comparison of TPE-based mass optimization and constrained mass with UC values for a 9-element truss. $(n = 1)$



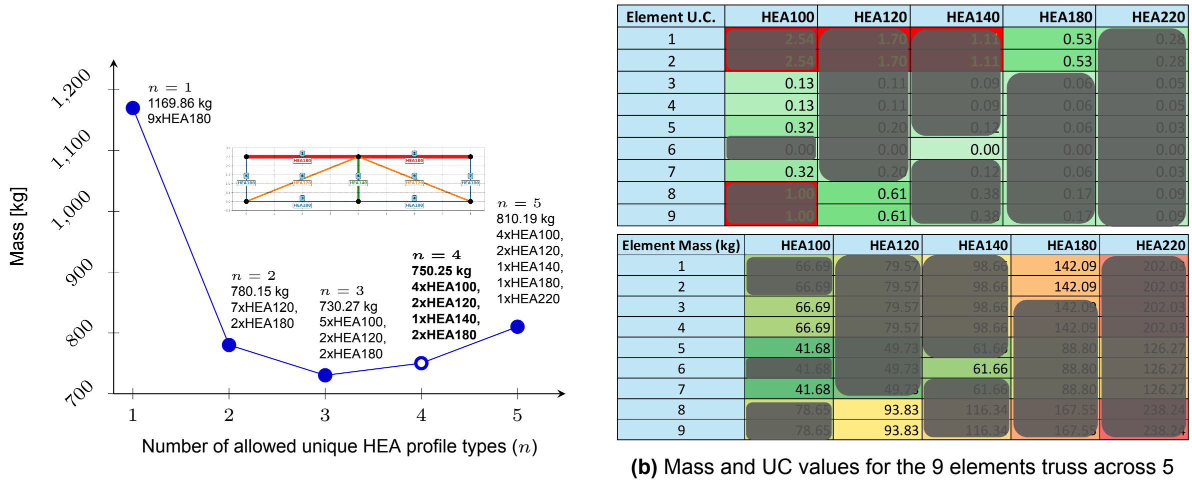
Element U.C.	HEA100	HEA120	HEA140	HEA180	HEA220
1	2.54	1.70	1.11	0.53	0.28
2	2.54	1.70	1.11	0.53	0.28
3	0.13	0.11	0.09	0.06	0.05
4	0.13	0.11	0.09	0.06	0.05
5	0.32	0.20	0.12	0.06	0.03
6	0.00	0.00	0.00	0.00	0.00
7	0.32	0.20	0.12	0.06	0.03
8	1.00	0.61	0.38	0.17	0.09
9	1.00	0.61	0.38	0.17	0.09

Element Mass (kg)	HEA100	HEA120	HEA140	HEA180	HEA220
1	66.69	79.57	98.66	142.09	202.03
2	66.69	79.57	98.66	142.09	202.03
3	66.69	79.57	98.66	142.09	202.03
4	66.69	79.57	98.66	142.09	202.03
5	41.68	49.73	61.66	88.80	126.27
6	41.68	49.73	61.66	88.80	126.27
7	41.68	49.73	61.66	88.80	126.27
8	78.65	93.83	116.34	167.55	238.24
9	78.65	93.83	116.34	167.55	238.24

(a) TPE-Based Optimization of Mass vs n

(b) Mass and UC values for the 9 elements truss across 5 allowed unique HEA profiles ($n = 2$) *(highlighted)* grouping

Figure C.2: Comparison of TPE-based mass optimization and constrained mass with UC values for a 9-element truss. ($n = 2$)



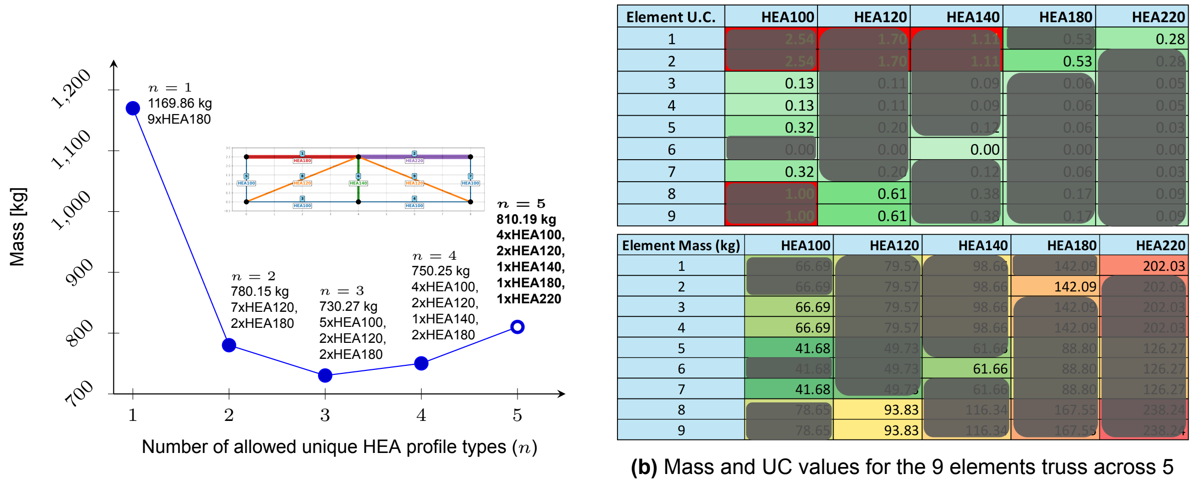
Element U.C.	HEA100	HEA120	HEA140	HEA180	HEA220
1	2.54	1.70	1.11	0.53	0.28
2	2.54	1.70	1.11	0.53	0.28
3	0.13	0.11	0.09	0.06	0.05
4	0.13	0.11	0.09	0.06	0.05
5	0.32	0.20	0.12	0.06	0.03
6	0.00	0.00	0.00	0.00	0.00
7	0.32	0.20	0.12	0.06	0.03
8	1.00	0.61	0.38	0.17	0.09
9	1.00	0.61	0.38	0.17	0.09

Element Mass (kg)	HEA100	HEA120	HEA140	HEA180	HEA220
1	66.69	79.57	98.66	142.09	202.03
2	66.69	79.57	98.66	142.09	202.03
3	66.69	79.57	98.66	142.09	202.03
4	66.69	79.57	98.66	142.09	202.03
5	41.68	49.73	61.66	88.80	126.27
6	41.68	49.73	61.66	88.80	126.27
7	41.68	49.73	61.66	88.80	126.27
8	78.65	93.83	116.34	167.55	238.24
9	78.65	93.83	116.34	167.55	238.24

(a) TPE-Based Optimization of Mass vs n

(b) Mass and UC values for the 9 elements truss across 5 allowed unique HEA profiles ($n = 4$) *(highlighted)* grouping

Figure C.3: Comparison of TPE-based mass optimization and constrained mass with UC values for a 9-element truss. ($n = 4$)



Element U.C.	HEA100	HEA120	HEA140	HEA180	HEA220
1	2.54	1.70	1.11	0.53	0.28
2	2.54	1.70	1.11	0.53	0.28
3	0.13	0.11	0.09	0.06	0.05
4	0.13	0.11	0.09	0.06	0.05
5	0.32	0.20	0.12	0.06	0.03
6	0.00	0.00	0.00	0.00	0.00
7	0.32	0.20	0.12	0.06	0.03
8	1.00	0.61	0.38	0.17	0.09
9	1.00	0.61	0.38	0.17	0.09

Element Mass (kg)	HEA100	HEA120	HEA140	HEA180	HEA220
1	66.69	79.57	98.66	142.09	202.03
2	66.69	79.57	98.66	142.09	202.03
3	66.69	79.57	98.66	142.09	202.03
4	66.69	79.57	98.66	142.09	202.03
5	41.68	49.73	61.66	88.80	126.27
6	41.68	49.73	61.66	88.80	126.27
7	41.68	49.73	61.66	88.80	126.27
8	78.65	93.83	116.34	167.55	238.24
9	78.65	93.83	116.34	167.55	238.24

(a) TPE-Based Optimization of Mass vs n

(b) Mass and UC values for the 9 elements truss across 5 allowed unique HEA profiles ($n = 5$) *(highlighted)* grouping

Figure C.4: Comparison of TPE-based mass optimization and constrained mass with UC values for a 9-element truss. ($n = 5$)

D

Python model validation results

D.1. Python model validation outputs for Mass

This section presents the validation of the Tree-structured Parzen Estimator (TPE) sampler against Exhaustive Search (EXS) for parametric truss optimization, focusing on the convergence of (n from 1 to 5) allowed HEA profiles with varying (N_{trials}). The distributions of TPE and EXS are compared across Mass objective to evaluate TPE performance. Figures D.1, D.2 and D.3 further illustrate this validation by showcasing TPE convergence to EXS across Mass, Connection Degree, Symmetry, and Beam Continuity, with Figure D.3 incorporating objective weight distributions.

D.2. Python model validation outputs for Connection Degree

This section presents the validation of the Tree-structured Parzen Estimator (TPE) sampler against Exhaustive Search (EXS) for parametric truss optimization, focusing on the convergence of (n from 1 to 5) allowed HEA profiles with varying (N_{trials}). The distributions of TPE and EXS are compared across Connection Degree objective to evaluate TPE's performance. Figures D.4, D.5 and D.6 further illustrate this validation by showcasing TPE convergence to EXS across Mass, Connection Degree, Symmetry, and Beam Continuity, with Figure D.6 incorporating objective weight distributions.

D.3. Python model validation outputs for Symmetry

This section presents the validation of the Tree-structured Parzen Estimator (TPE) sampler against Exhaustive Search (EXS) for parametric truss optimization, focusing on the convergence of (n from 1 to 5) allowed HEA profiles with varying (N_{trials}). The distributions of TPE and EXS are compared across Symmetry objective to evaluate TPE's performance. Figures D.7, D.8 and D.9 further illustrate this validation by showcasing TPE convergence to EXS across Mass, Connection Degree, Symmetry, and Beam Continuity, with Figure D.9 incorporating objective weight distributions.

D.4. Python model validation outputs for Beam Continuity

This section presents the validation of the Tree-structured Parzen Estimator (TPE) sampler against Exhaustive Search (EXS) for parametric truss optimization, focusing on the convergence of (n from 1 to 5) allowed HEA profiles with varying (N_{trials}). The distributions of TPE and EXS are compared across Beam Continuity objective to evaluate TPE's performance. Figures D.10, D.11 and D.12 further illustrate this validation by showcasing TPE convergence to EXS across Mass, Connection Degree, Symmetry, and Beam Continuity, with Figure D.12 incorporating objective weight distributions.

TPE and EXS distributions - Mass validation

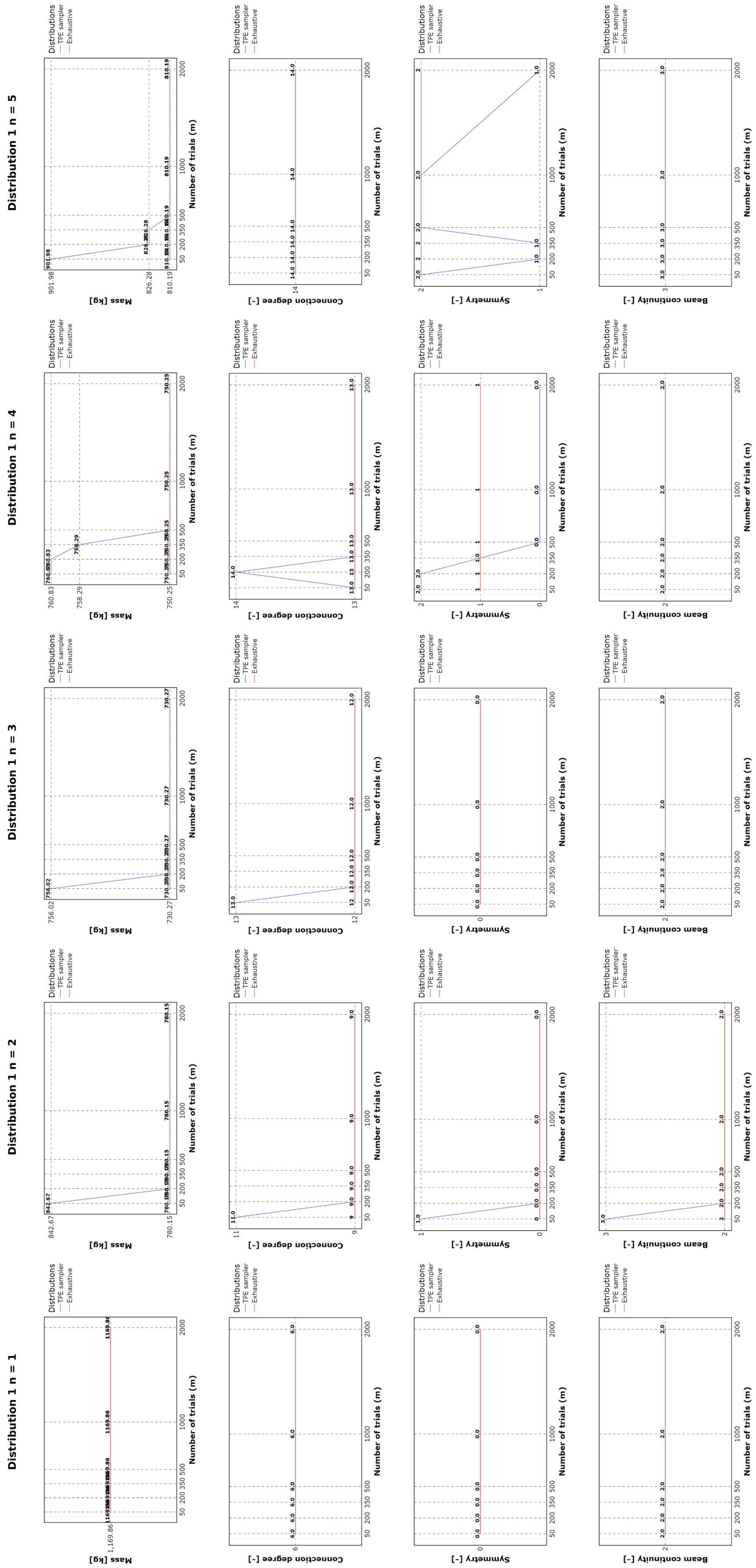


Figure D.1: Python - generated distributions of TPE sampler and Exhaustive Search (EXS) for Mass, Connection Degree, Symmetry, and Beam Continuity, validating TPE through Mass convergence to EXS across $n = 1$ to $n = 5$ with varying N_{trials}.

TPE and EXS distributions - highlighted Mass convergences

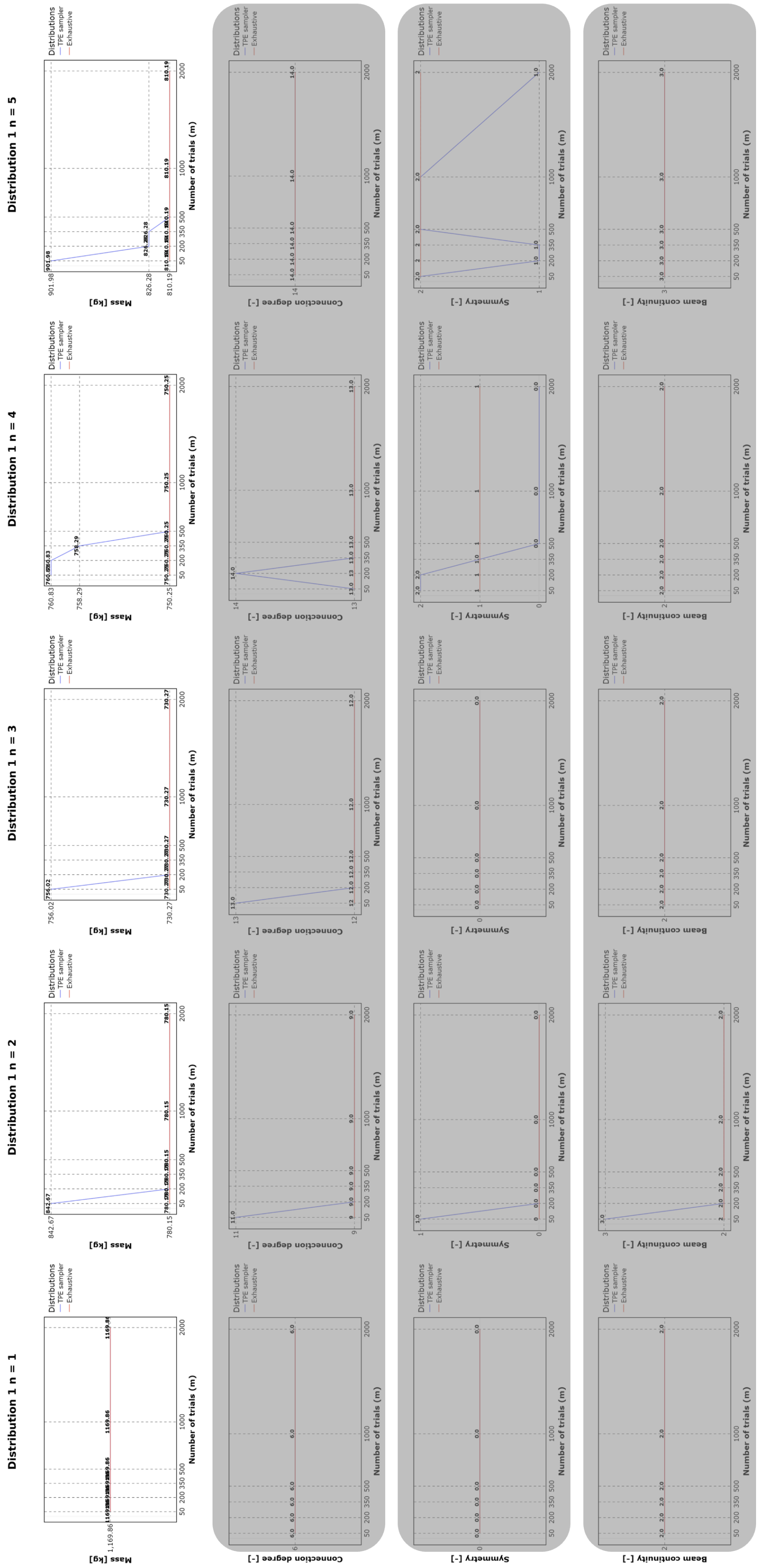


Figure D.2: Python - generated distributions of TPE sampler and Exhaustive Search (EXS) for Mass, Connection Degree, Symmetry, and Beam Continuity, validating TPE through Mass *(highlighted)* convergence to EXS across $n = 1$ to $n = 5$ with varying N_{trials}.

TPE and EXS distributions - Mass validation with objective weights

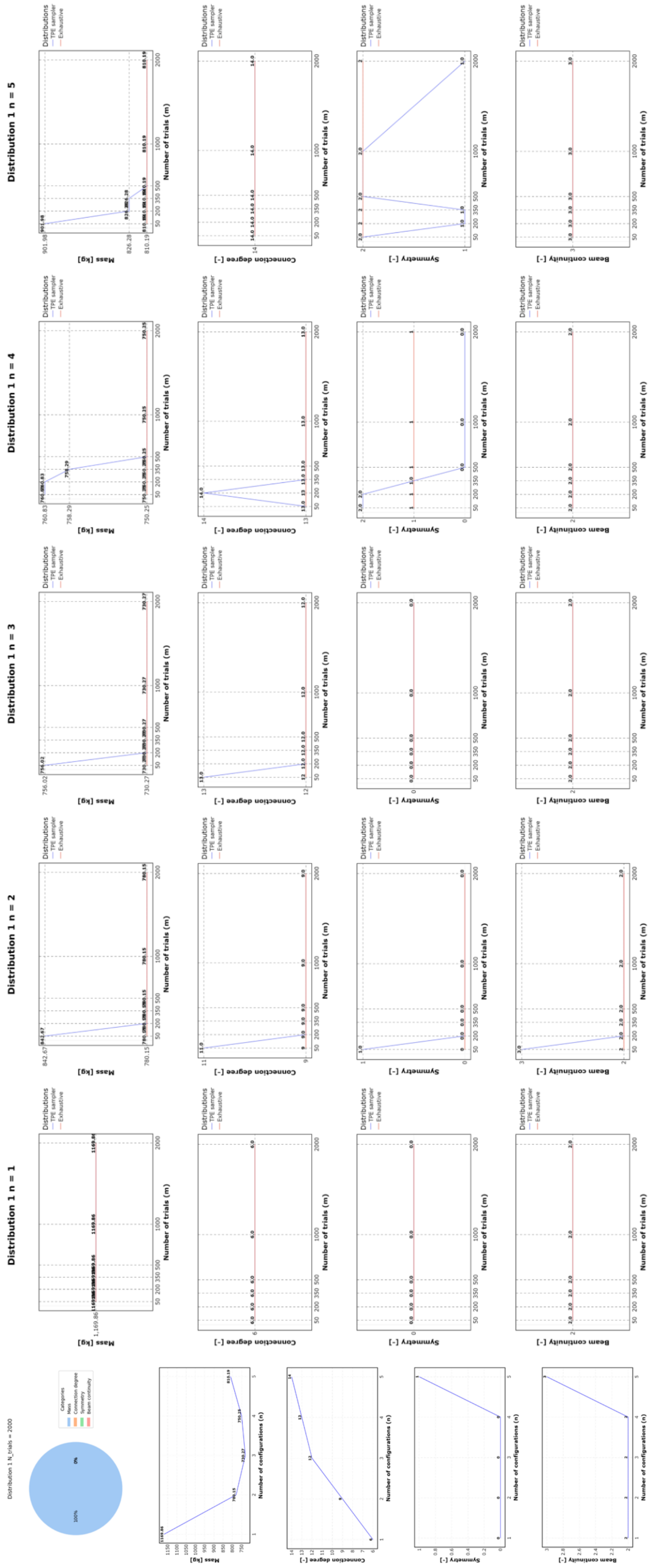


Figure D.3: Python - generated distributions of TPE sampler and Exhaustive Search (EXS) for Mass, Connection Degree, Symmetry, and Beam Continuity, validating TPE through Mass convergence to EXS across $n = 1$ to $n = 5$ with varying N_{trials}, with objective weight pie chart and objective function plots vs n.

TPE and EXS distributions - Connection Degree validation

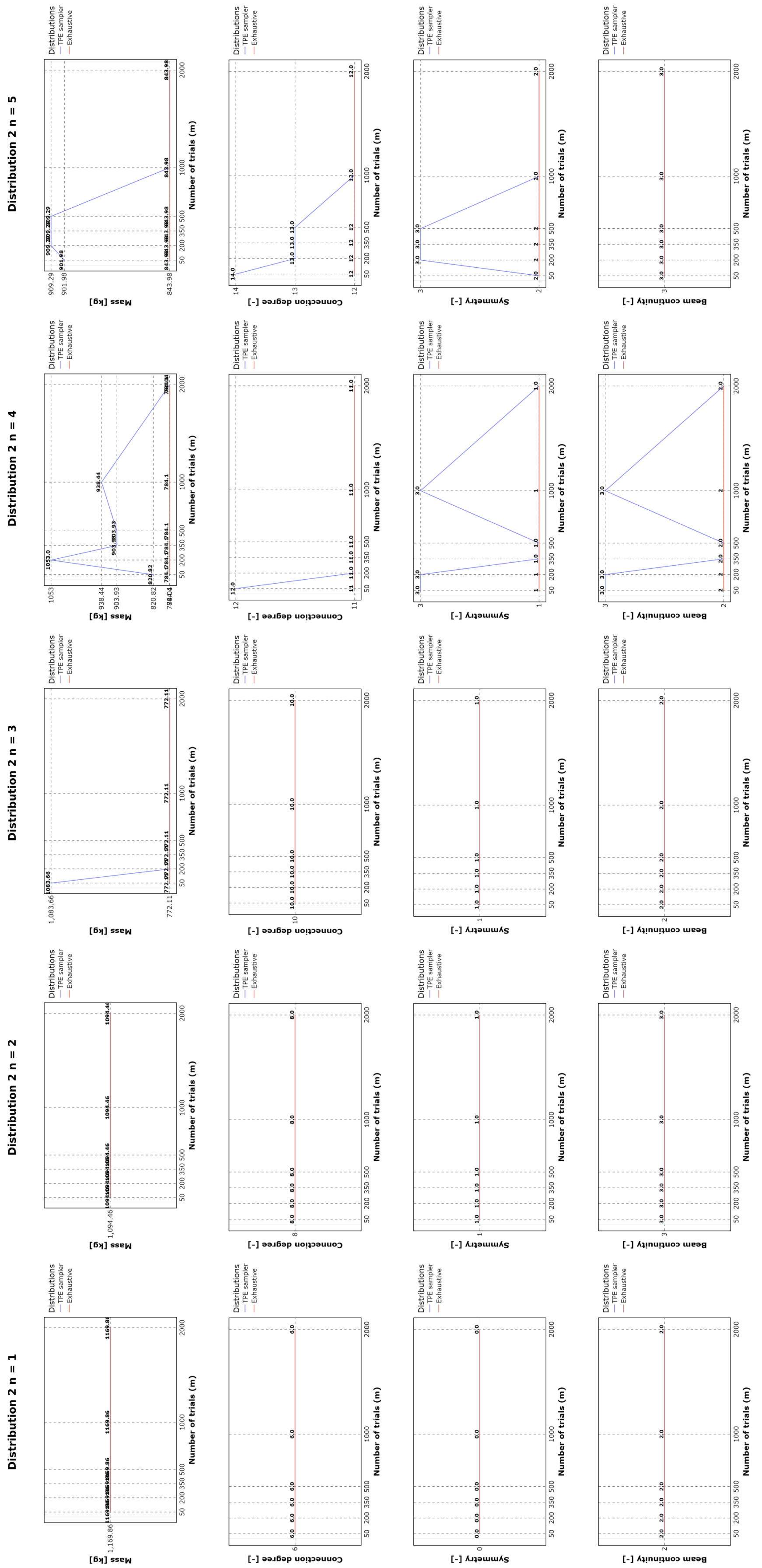


Figure D.4: Python - generated distributions of TPE sampler and Exhaustive Search (EXS) for Mass, Connection Degree, Symmetry, and Beam Continuity, validating TPE through Connection Degree convergence to EXS across $n = 1$ to $n = 5$ with varying N_{trials}.

TPE and EXS distributions - highlighted Connection Degree convergences

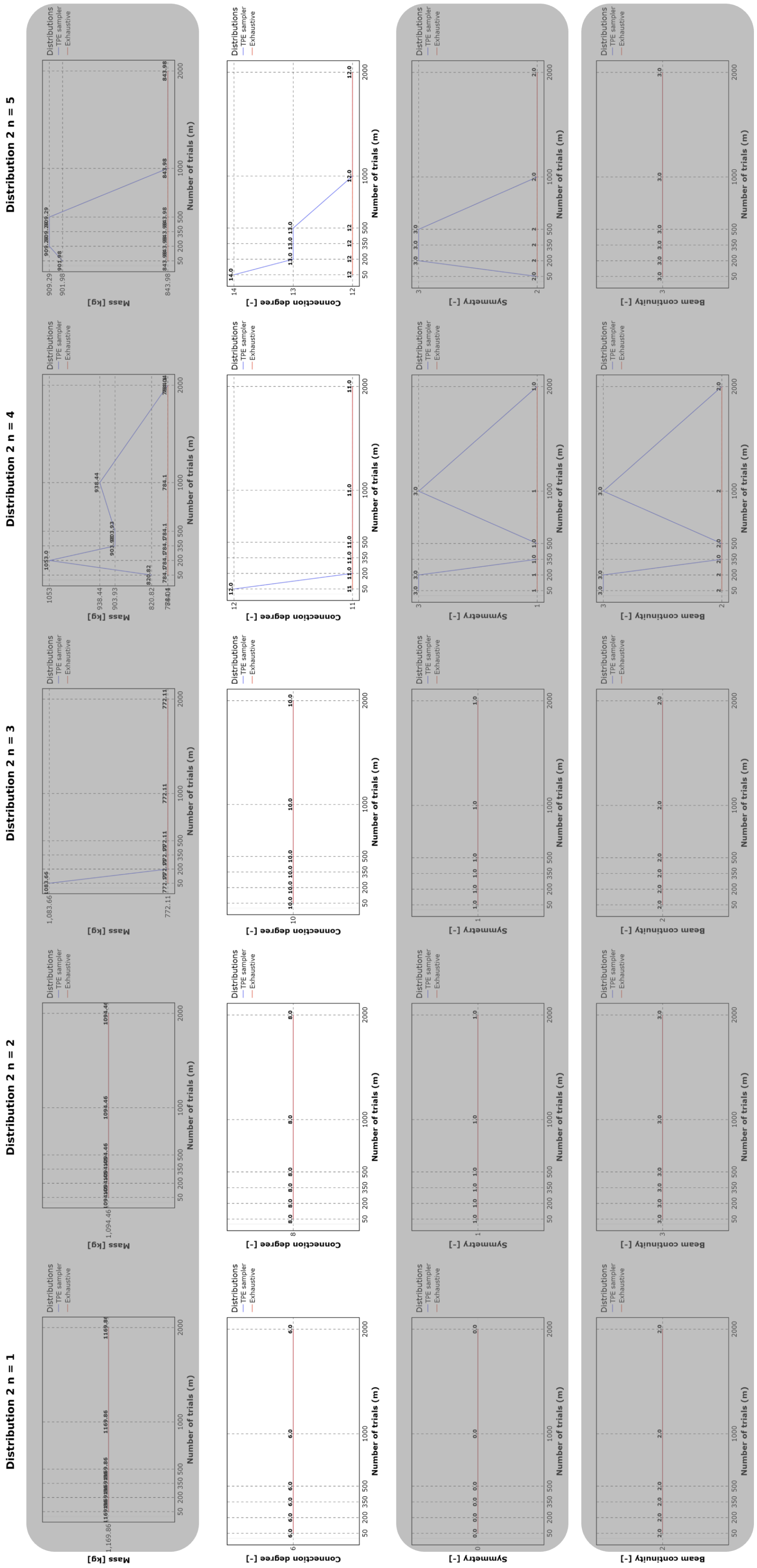


Figure D.5: Python - generated distributions of TPE sampler and Exhaustive Search (EXS) for Mass, Connection Degree, Symmetry, and Beam Continuity, validating TPE through Connection Degree *(highlighted)* convergence to EXS across $n = 1$ to $n = 5$ with varying N_{trials}.

TPE and EXS distributions - Connection Degree validation with objective weights

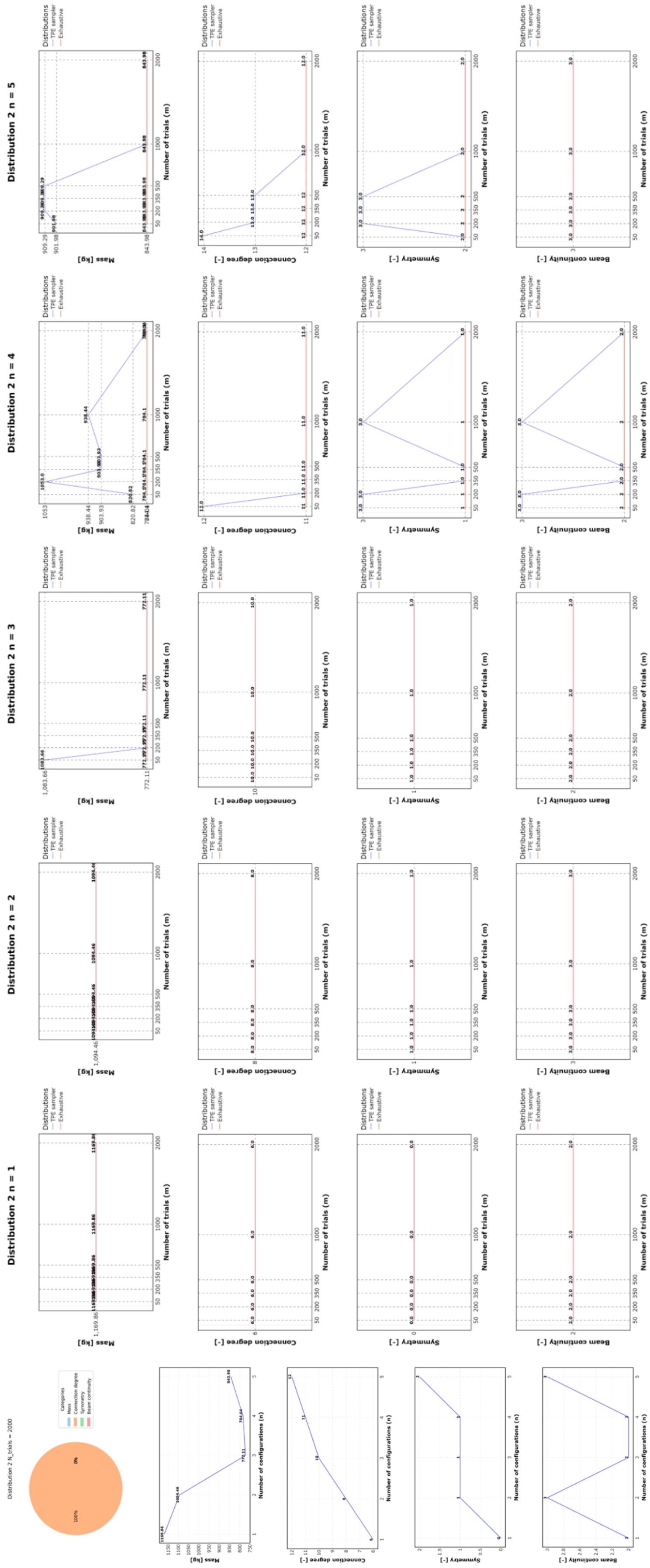


Figure D.6: Python - generated distributions of TPE sampler and Exhaustive Search (EXS) for Mass, Connection Degree, Symmetry, and Beam Continuity, validating TPE through Connection Degree convergence to EXS across $n = 1$ to $n = 5$ with varying N_{trials}, with objective weight pie chart and objective function plots vs n.

TPE and EXS distributions - Symmetry validation

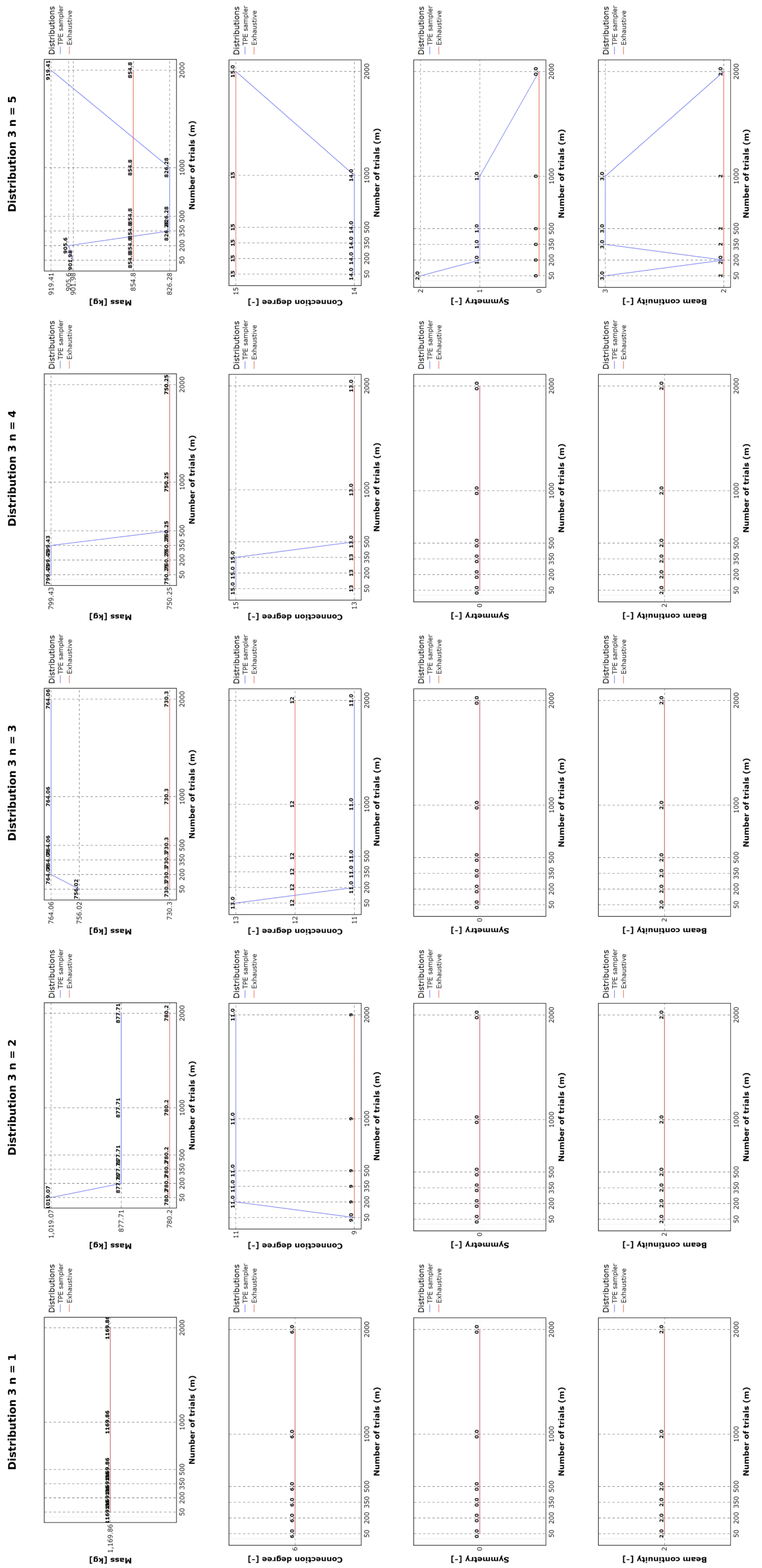


Figure D.7: Python - generated distributions of TPE sampler and Exhaustive Search (EXS) for Mass, Connection Degree, Symmetry, and Beam Continuity, validating TPE through Symmetry convergence to EXS across $n = 1$ to $n = 5$ with varying N_{trials}.

TPE and EXS distributions - highlighted Symmetry convergences

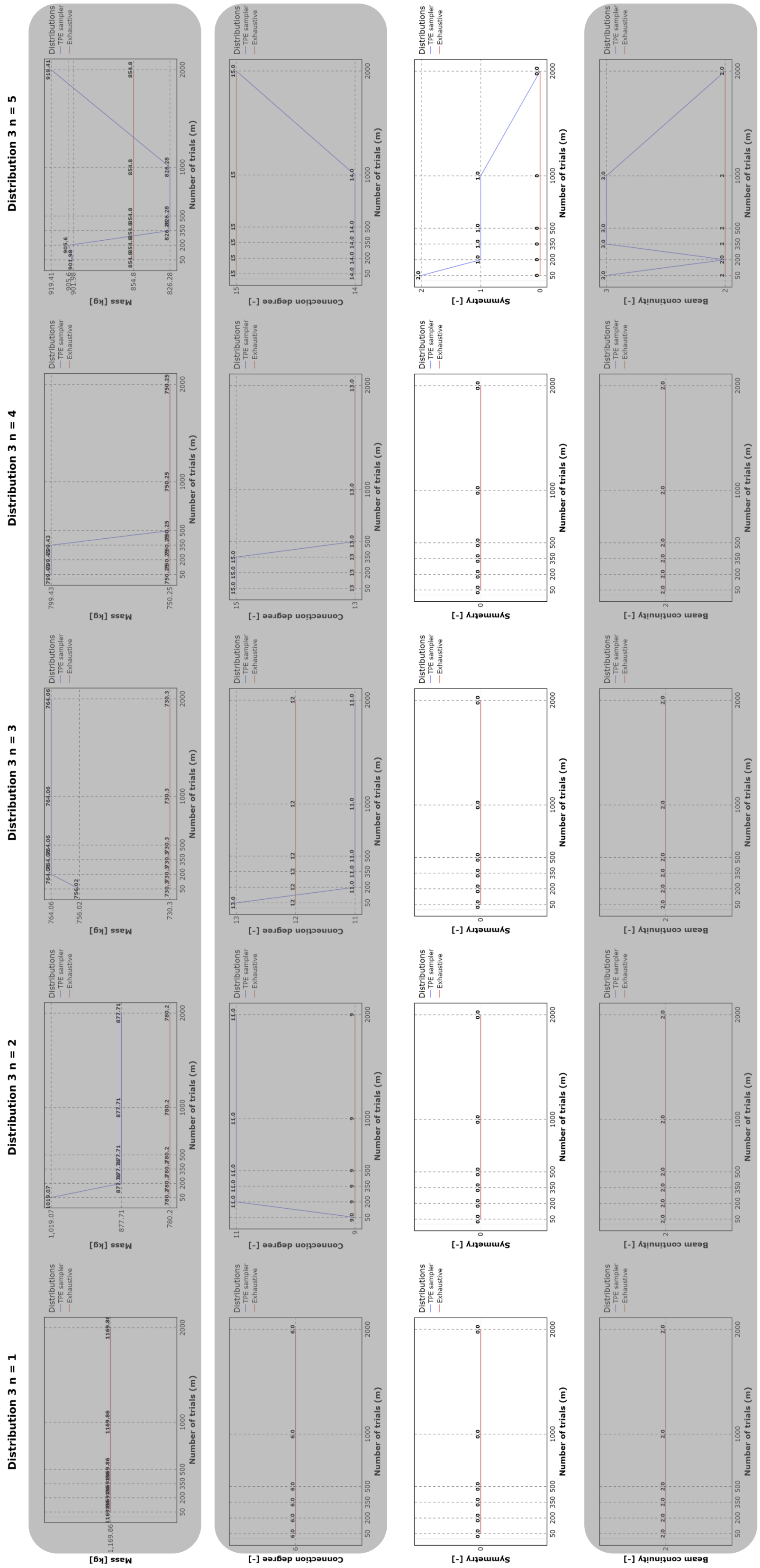


Figure D.8: Python - generated distributions of TPE sampler and Exhaustive Search (EXS) for Mass, Connection Degree, Symmetry, and Beam Continuity, validating TPE through Symmetry *(highlighted)* convergence to EXS across $n = 1$ to $n = 5$ with varying N_{trials}.

TPE and EXS distributions - Symmetry validation with objective weights

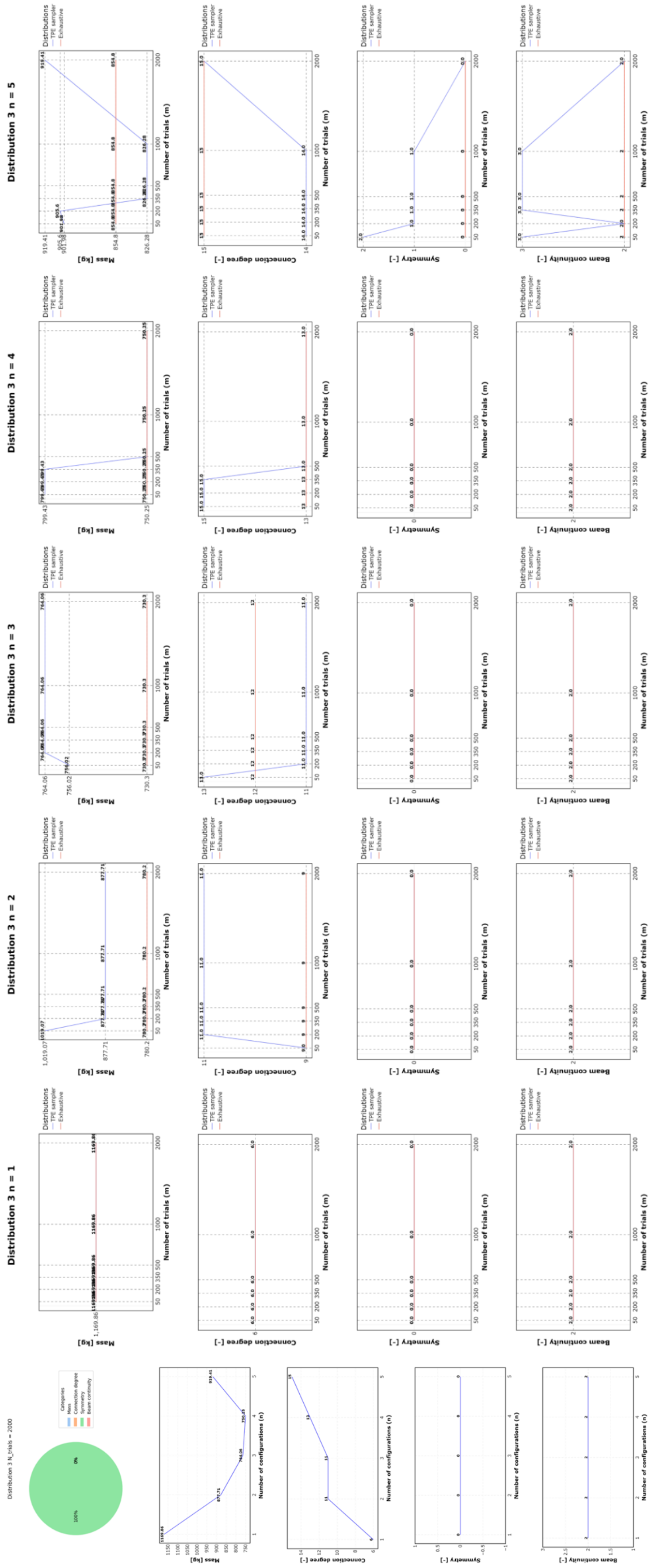


Figure D.9: Python - generated distributions of TPE sampler and Exhaustive Search (EXS) for Mass, Connection Degree, Symmetry, and Beam Continuity, validating TPE through Symmetry convergence to EXS across $n = 1$ to $n = 5$ with varying N_{trials}, with objective weight pie chart and objective function plots vs n.

TPE and EXS distributions - Beam Continuity validation

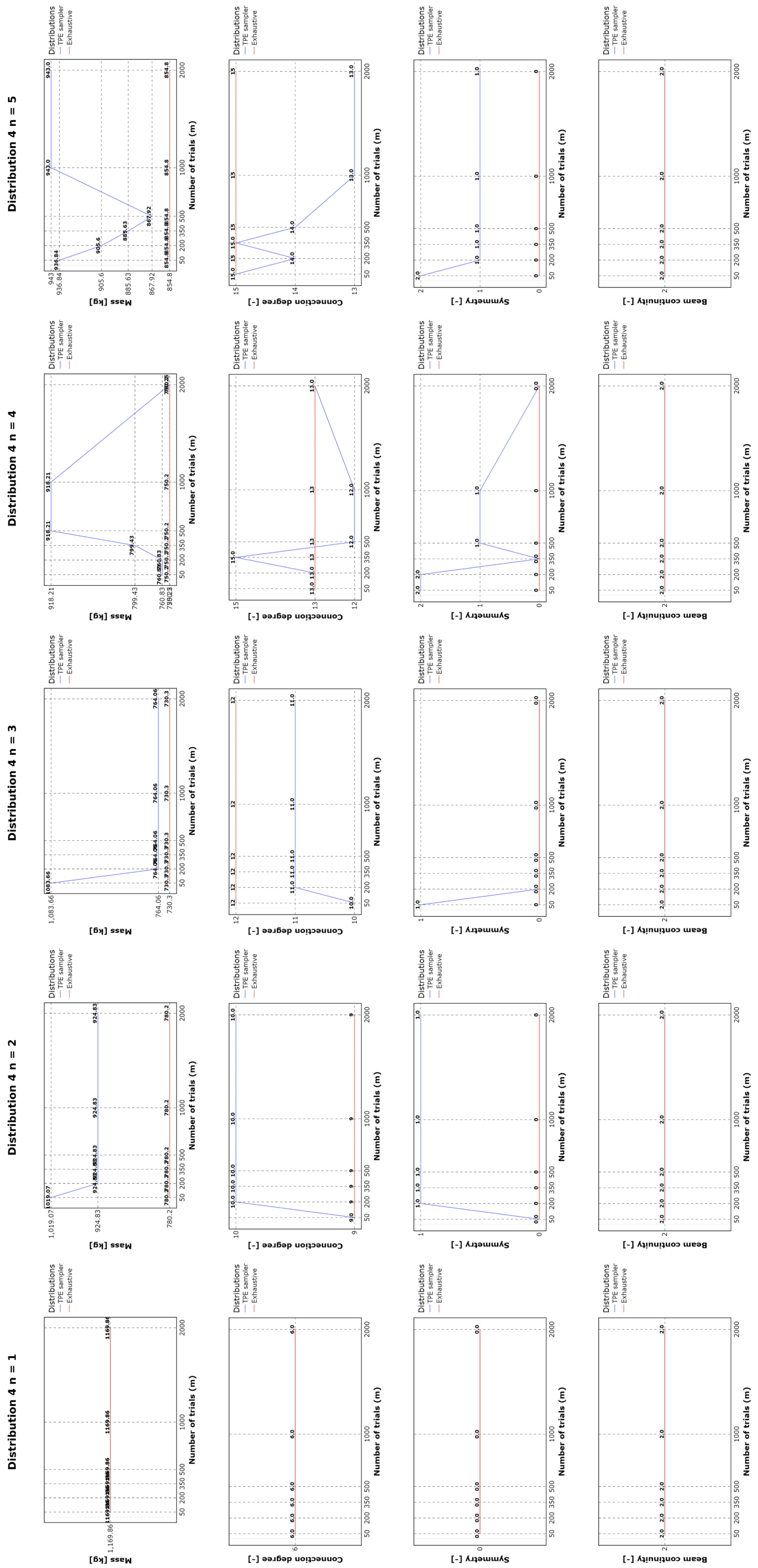


Figure D.10: Python - generated distributions of TPE sampler and Exhaustive Search (EXS) for Mass, Connection Degree, Symmetry, and Beam Continuity, validating TPE through Beam Continuity convergence to EXS across $n = 1$ to $n = 5$ with varying N_{trials}.

TPE and EXS distributions - highlighted Beam Continuity convergences

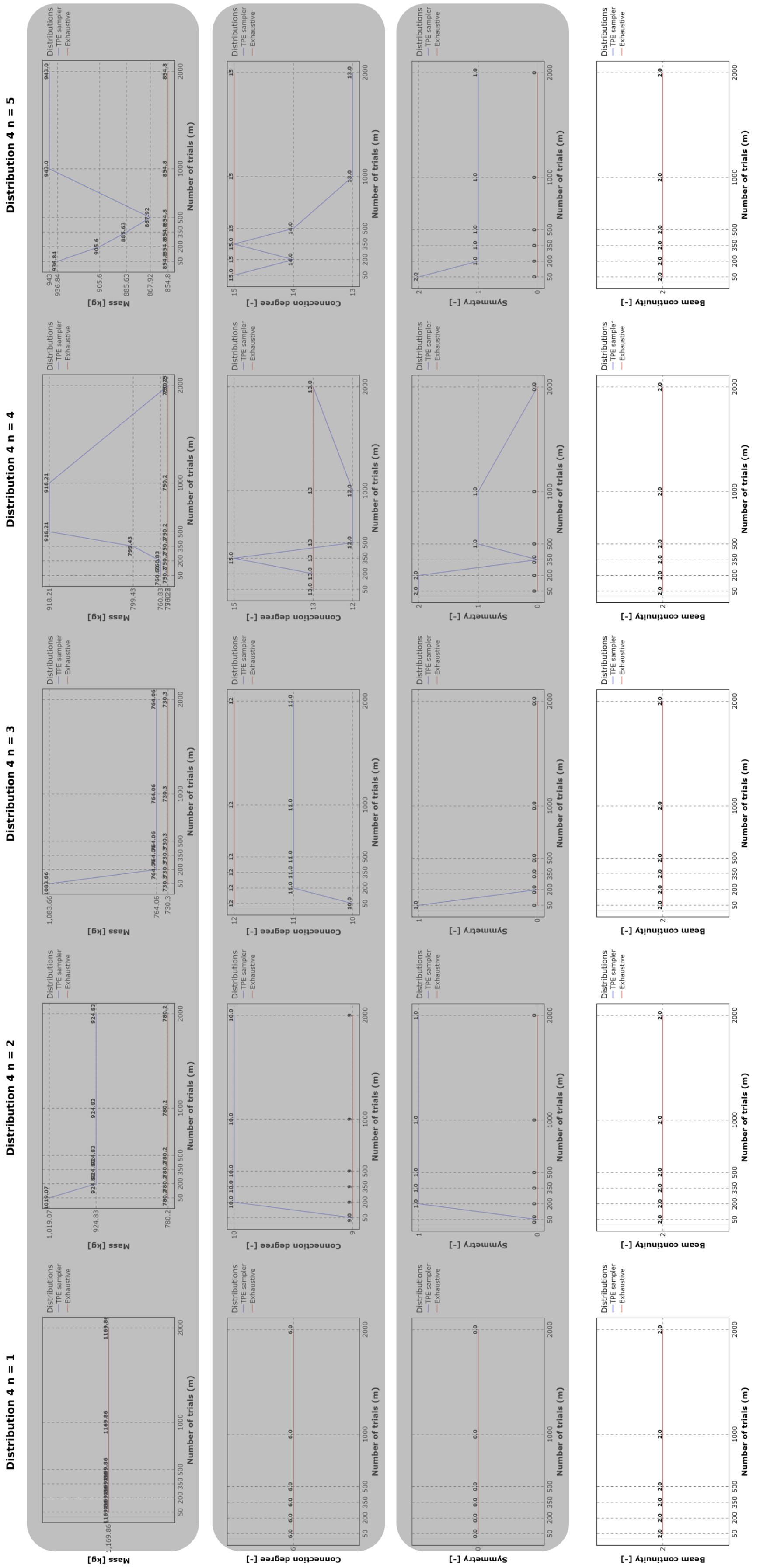


Figure D.11: Python - generated distributions of TPE sampler and Exhaustive Search (EXS) for Mass, Connection Degree, Symmetry, and Beam Continuity, validating TPE through Beam Continuity *(highlighted)* convergence to EXS across $n = 1$ to $n = 5$ with varying N_{trials}.

TPE and EXS distributions - Beam Continuity validation with objective weights

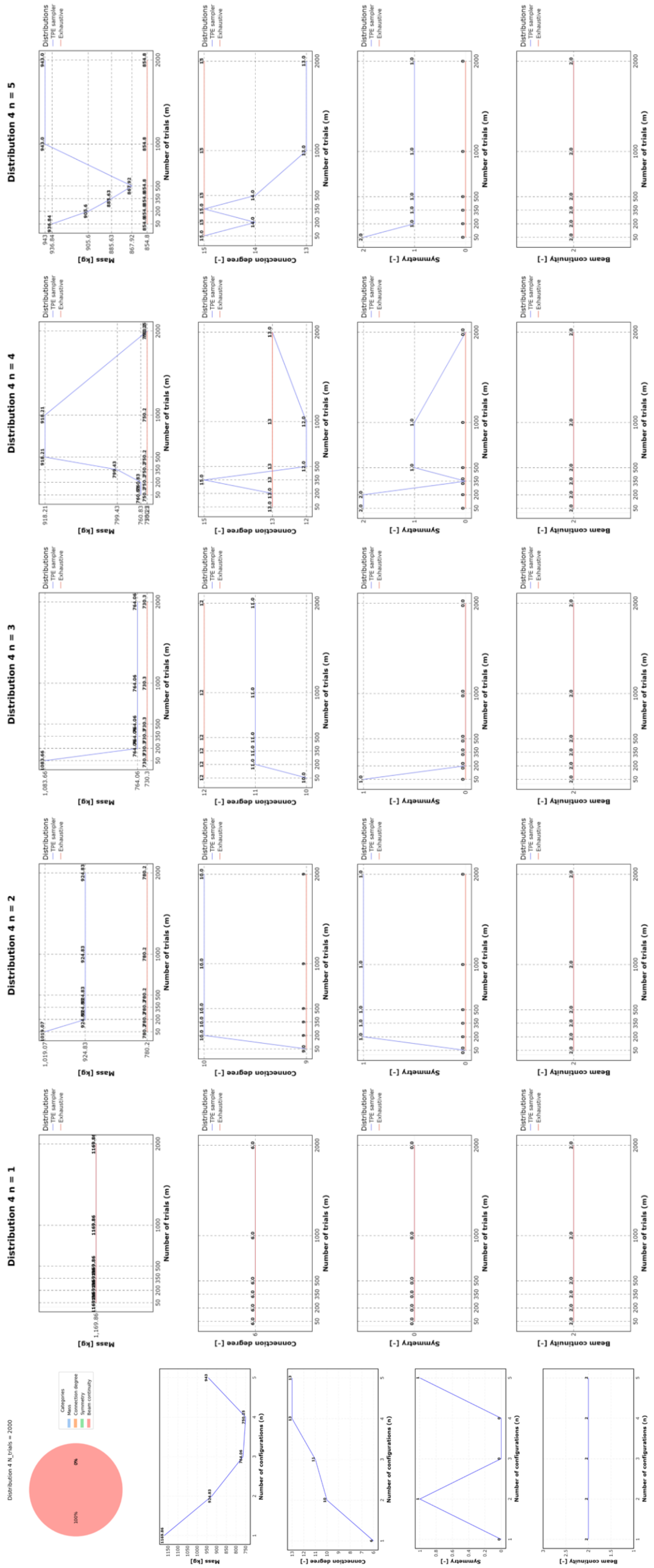


Figure D.12: Python - generated distributions of TPE sampler and Exhaustive Search (EXS) for Mass, Connection Degree, Symmetry, and Beam Continuity, validating TPE through Beam Continuity convergence to EXS across $n = 1$ to $n = 5$ with varying N_{trials}, with objective weight pie chart and objective function plots vs n.

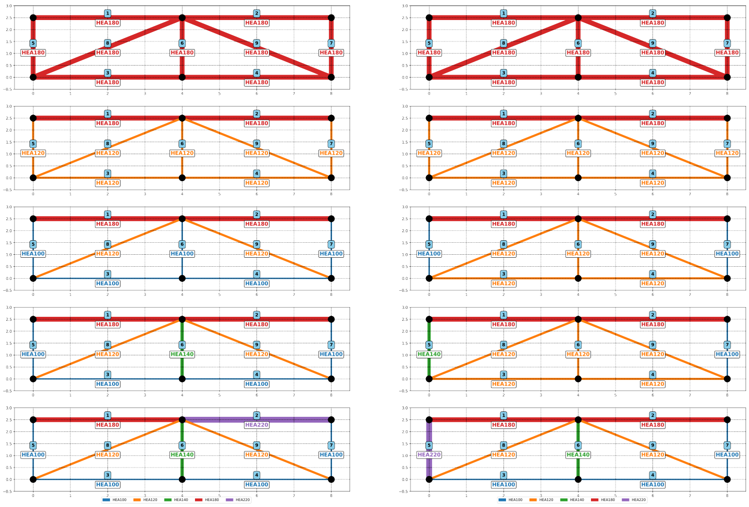
E

TPE Performance in Single vs Multi-Objective Scenarios

E.1. TPE($\mathcal{W}_1$) Comaprison with single obejective optimization TPE

Table E.1: Comparison of Tree-structured Parzen Estimator of Mass TPE(*M*) vs TPE($\mathcal{W}_1$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w**(TPE(*M*))**	$\mathcal{W}_1$	**TPE(*M*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_1$)**	Δ
Mass	1	0.25	1169.86	1169.86	=	780.15	780.15	=	730.27	764.06	↑	750.25	784.04	↑	810.19	834.83	↑
Connection Degree	0	0.25	6	6	=	9	9	=	12	11	↓	13	11	↓	14	14	=
Symmetry	0	0.25	0	0	=	0	0	=	0	0	=	0	1	↑	1	1	=
Beam Continuity	0	0.25	2	2	=	2	2	=	2	2	=	2	2	=	3	2	↓



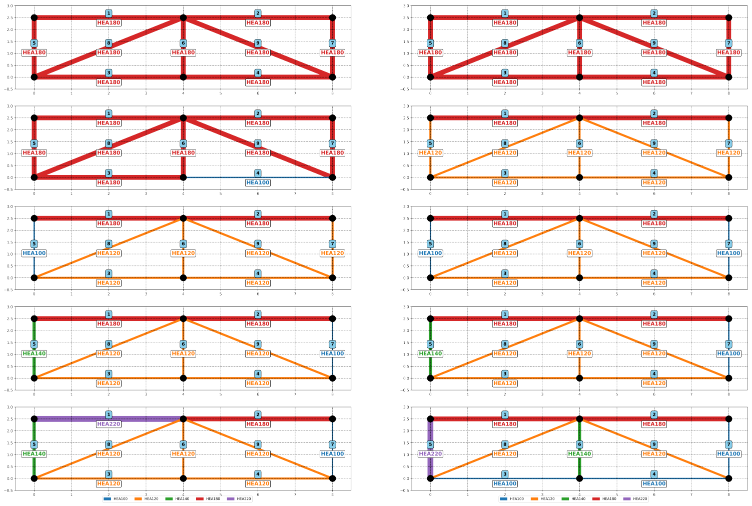
(a) TPE(*M*) truss configuration

(b) TPE($\mathcal{W}_1$) truss configuration

Figure E.1: Comparison between TPE(*M*) vs TPE($\mathcal{W}_1$) truss configurations.

Table E.2: Comparison of Tree-structured Parzen Estimator of Connection Degree TPE(*CN*) vs TPE($\mathcal{W}_1$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w**(TPE(*CN*))**	$\mathcal{W}_1$	**TPE(*CN*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_1$)**	Δ
Mass	0	0.25	1169.86	1169.86	=	1094.46	780.15	↓	772.11	764.06	↓	784.04	784.04	=	843.98	834.83	↓
Connection Degree	1	0.25	6	6	=	8	9	↑	10	11	↑	11	11	=	12	14	↑
Symmetry	0	0.25	0	0	=	1	0	↓	1	0	↓	2	1	↓	2	1	↓
Beam Continuity	0	0.25	2	2	=	3	2	↓	2	2	=	2	2	=	3	2	↓



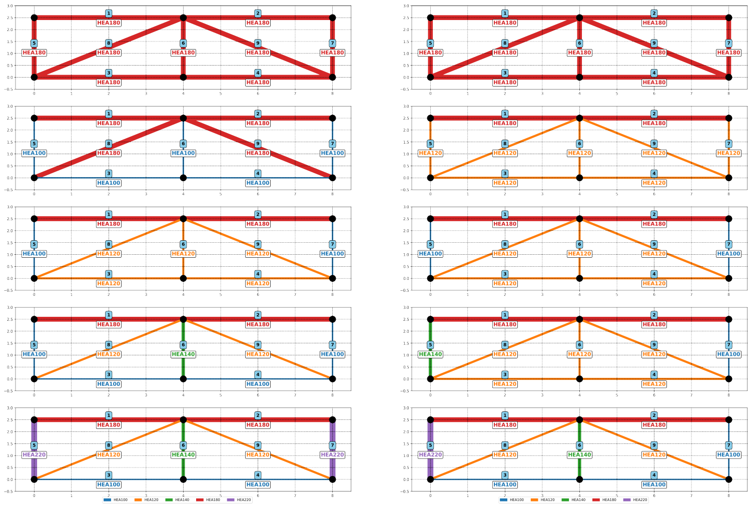
(a) TPE(*CN*) truss configuration

(b) TPE($\mathcal{W}_1$) truss configuration

Figure E.2: Comparison between TPE(*CN*) vs TPE($\mathcal{W}_1$) truss configurations.

Table E.3: Comparison of Tree-structured Parzen Estimator of Symmetry TPE(*SYM*) vs TPE($\mathcal{W}_1$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w**(TPE(*SYM*))**	$\mathcal{W}_1$	**TPE(*SYM*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_1$)**	Δ
Mass	0	0.25	1169.86	1169.86	=	877.71	780.15	↓	764.06	764.06	=	750.25	784.04	↑	919.41	834.83	↓
Connection Degree	0	0.25	6	6	=	11	9	↓	11	11	=	13	11	↓	15	14	↓
Symmetry	1	0.25	0	0	=	0	0	=	0	0	=	0	1	↑	0	1	↑
Beam Continuity	0	0.25	2	2	=	2	2	=	2	2	=	2	2	=	2	2	=



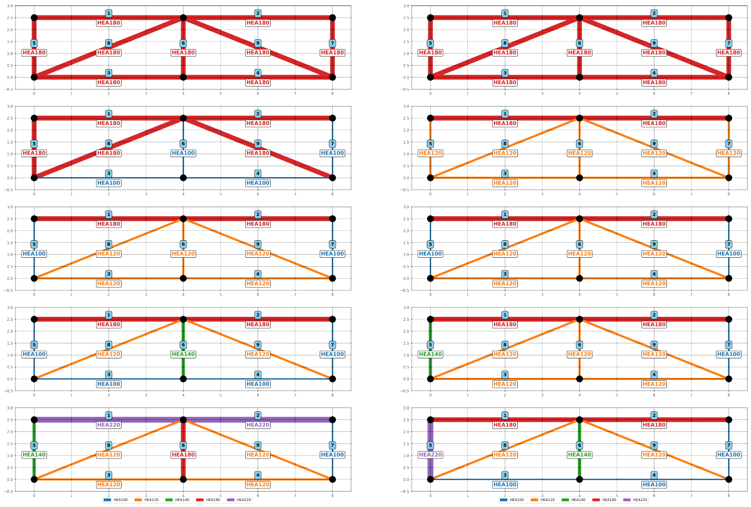
(a) TPE(*SYM*) truss configuration

(b) TPE($\mathcal{W}_1$) truss configuration

Figure E.3: Comparison between TPE(*SYM*) vs TPE($\mathcal{W}_1$) truss configurations.

Table E.4: Comparison of Tree-structured Parzen Estimator of Beam Continuity TPE(*BCN*) vs TPE($\mathcal{W}_1$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w**(TPE(*BCN*))**	$\mathcal{W}_1$	**TPE(*BCN*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_1$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_1$)**	Δ
Mass	0	0.25	1169.86	1169.86	=	924.83	780.15	↓	764.06	764.06	=	750.25	784.04	↑	943	834.83	↓
Connection Degree	0	0.25	6	6	=	10	9	↓	11	11	=	13	11	↓	13	14	↑
Symmetry	0	0.25	0	0	=	1	0	↓	0	0	=	0	1	↑	1	1	=
Beam Continuity	1	0.25	2	2	=	2	2	=	2	2	=	2	2	=	2	2	=



(a) TPE(*BCN*) truss configuration

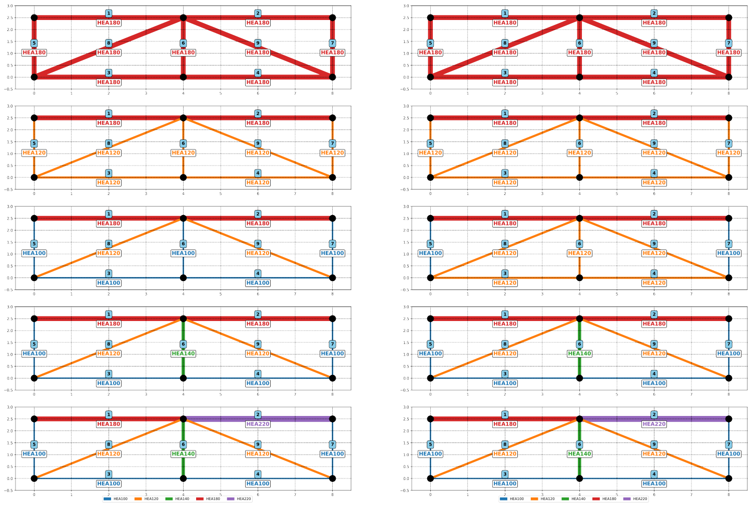
(b) TPE($\mathcal{W}_1$) truss configuration

Figure E.4: Comparison between TPE(*SYM*) vs TPE($\mathcal{W}_1$) truss configurations.

E.2. TPE($\mathcal{W}_2$) Comaprison with single obejective optimization TPE

Table E.5: Comparison of Tree-structured Parzen Estimator of Mass TPE(*M*) vs TPE($\mathcal{W}_2$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w(**TPE(*M*)**)	$\mathcal{W}_2$	**TPE(*M*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_2$)**	Δ
Mass	1	0.4	1169.86	1169.86	=	780.15	780.15	=	730.27	764.06	↑	750.25	750.25	=	810.19	810.19	=
Connection Degree	0	0.1	6	6	=	9	8	↓	12	11	↓	13	13	=	14	14	=
Symmetry	0	0.4	0	0	=	0	0	=	0	0	=	1	0	↓	1	1	=
Beam Continuity	0	0.1	2	2	=	2	2	=	2	2	=	2	2	=	3	3	=



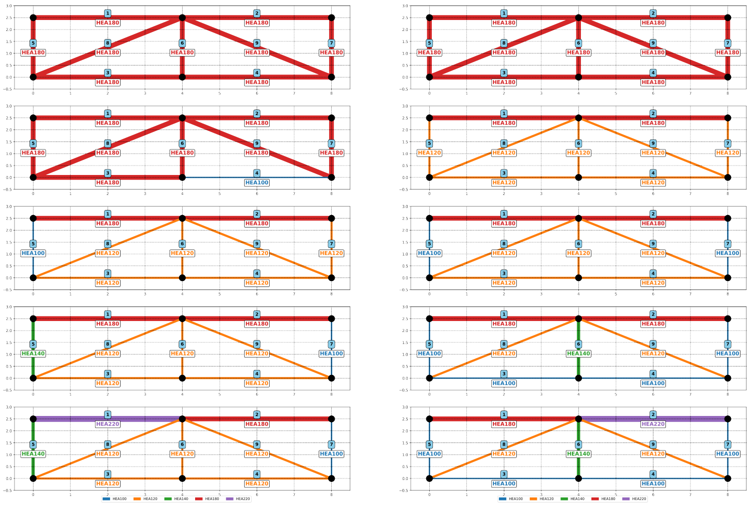
(a) TPE(*M*) truss configuration

(b) TPE($\mathcal{W}_2$) truss configuration

Figure E.5: Comparison between TPE(*M*) vs TPE($\mathcal{W}_2$) truss configurations.

Table E.6: Comparison of Tree-structured Parzen Estimator of Connection Degree TPE(*CN*) vs TPE($\mathcal{W}_2$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w(**TPE(*CN*)**)	$\mathcal{W}_2$	**TPE(*CN*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_2$)**	Δ
Mass	0	0.4	1169.86	1169.86	=	1094.46	780.15	↓	772.11	764.06	↓	784.04	750.25	↓	843.98	810.19	↓
Connection Degree	1	0.1	6	6	=	8	8	=	10	11	↑	11	13	↑	12	14	↑
Symmetry	0	0.4	0	0	=	1	0	↓	1	0	↓	1	0	↓	2	1	↓
Beam Continuity	0	0.1	2	2	=	3	2	↓	2	2	=	2	2	=	3	3	=



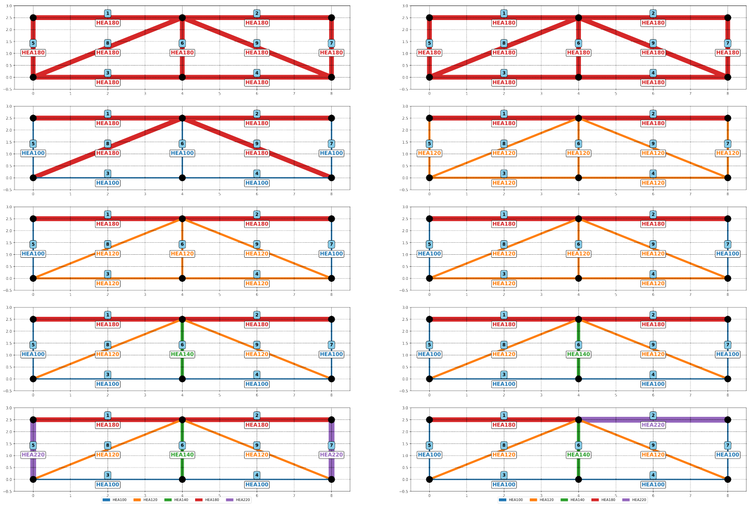
(a) TPE(*CN*) truss configuration

(b) TPE($\mathcal{W}_2$) truss configuration

Figure E.6: Comparison between TPE(*CN*) vs TPE($\mathcal{W}_2$) truss configurations.

Table E.7: Comparison of Tree-structured Parzen Estimator of Symmetry TPE(*SYM*) vs TPE($\mathcal{W}_2$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w(**TPE(*SYM*)**)	$\mathcal{W}_2$	**TPE(*SYM*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_2$)**	Δ
Mass	0	0.4	1169.86	1169.86	=	877.71	780.15	↓	764.06	764.06	=	750.25	750.25	=	919.41	810.19	=
Connection Degree	0	0.1	6	6	=	11	8	↓	11	11	=	13	13	=	15	14	↓
Symmetry	1	0.4	0	0	=	0	0	=	0	0	=	0	0	=	0	1	↑
Beam Continuity	0	0.1	2	2	=	2	2	=	2	2	=	2	2	=	2	3	↑



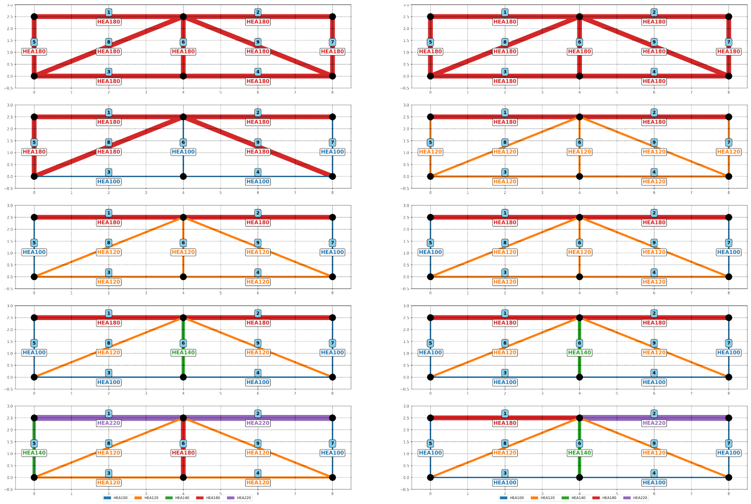
(a) TPE(*SYM*) truss configuration

(b) TPE($\mathcal{W}_2$) truss configuration

Figure E.7: Comparison between TPE(*SYM*) vs TPE($\mathcal{W}_2$) truss configurations.

Table E.8: Comparison of Tree-structured Parzen Estimator of Beam Continuity TPE(*BCN*) vs TPE($\mathcal{W}_2$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w(**TPE(*BCN*)**)	$\mathcal{W}_2$	**TPE(*BCN*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_2$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_2$)**	Δ
Mass	0	0.4	1169.86	1169.86	=	924.83	780.15	↓	764.06	764.06	=	750.25	750.25	=	943	810.19	↓
Connection Degree	0	0.1	6	6	=	10	8	↓	11	11	=	13	13	=	13	14	↑
Symmetry	0	0.4	0	0	=	1	0	↓	0	0	=	0	0	=	1	1	=
Beam Continuity	1	0.1	2	2	=	2	2	=	2	2	=	2	2	=	2	3	↑



(a) TPE(*BCN*) truss configuration

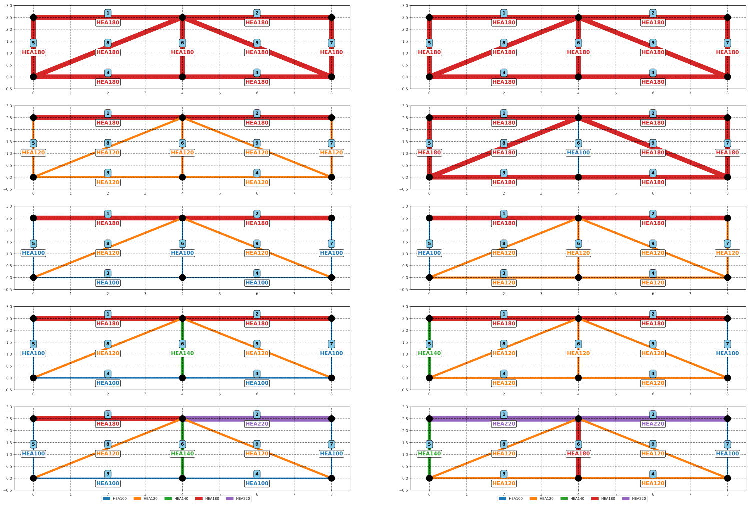
(b) TPE($\mathcal{W}_2$) truss configuration

Figure E.8: Comparison between TPE(*BCN*) vs TPE($\mathcal{W}_2$) truss configurations.

E.3. TPE($\mathcal{W}_3$) Comaprison with single obejective optimization TPE

Table E.9: Comparison of Tree-structured Parzen Estimator of Mass TPE(*M*) vs TPE($\mathcal{W}_3$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w**(TPE(*M*))**	$\mathcal{W}_3$	**TPE(*M*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_3$)**	Δ
Mass	1	0.1	1169.86	1169.86	=	780.15	1122.74	↑	730.27	772.11	↑	750.25	784.04	↑	810.19	943	↑
Connection Degree	0	0.4	6	6	=	9	8	↓	12	10	↓	13	11	↓	14	13	↓
Symmetry	0	0.1	0	0	=	0	0	=	0	1	↑	0	1	↑	1	1	=
Beam Continuity	0	0.4	2	2	=	2	2	=	2	2	=	2	2	=	3	2	↓



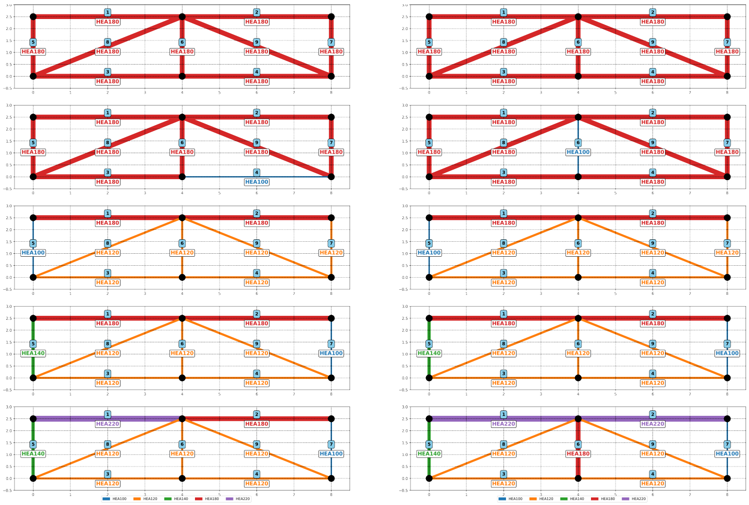
(a) TPE(*M*) truss configuration

(b) TPE($\mathcal{W}_3$) truss configuration

Figure E.9: Comparison between TPE(*M*) vs TPE($\mathcal{W}_3$) truss configurations.

Table E.10: Comparison of Tree-structured Parzen Estimator of Connection Degree TPE(*CN*) vs TPE($\mathcal{W}_3$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w(**TPE(*CN*)**)	$\mathcal{W}_3$	**TPE(*CN*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_3$)**	Δ
Mass	0	0.1	1169.86	1169.86	=	1094.46	1122.74	↑	772.11	772.11	=	784.04	784.04	=	843.98	943	↑
Connection Degree	1	0.4	6	6	=	8	8	=	10	10	=	11	11	=	12	13	↑
Symmetry	0	0.1	0	0	=	1	0	↓	1	1	=	1	1	=	2	1	↓
Beam Continuity	0	0.4	2	2	=	3	2	↓	2	2	=	2	2	=	3	2	↓



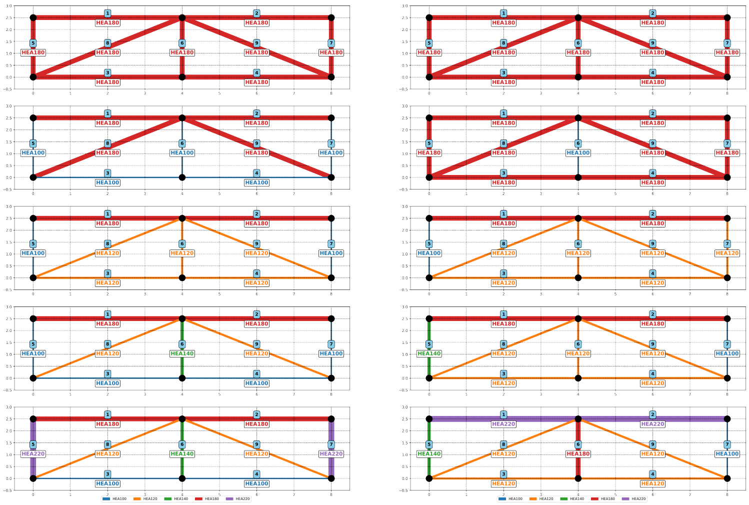
(a) TPE(*CN*) truss configuration

(b) TPE($\mathcal{W}_3$) truss configuration

Figure E.10: Comparison between TPE(*CN*) vs TPE($\mathcal{W}_3$) truss configurations.

Table E.11: Comparison of Tree-structured Parzen Estimator of Symmetry TPE(*SYM*) vs TPE($\mathcal{W}_3$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w(**TPE(*SYM*)**)	$\mathcal{W}_3$	**TPE(*SYM*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_3$)**	Δ
Mass	0	0.1	1169.86	1169.86	=	877.71	1122.74	↑	764.06	772.11	↑	750.25	784.04	↑	919.41	943	↑
Connection Degree	0	0.4	6	6	=	11	8	↓	11	10	↓	13	11	↓	15	13	↓
Symmetry	1	0.1	0	0	=	0	0	=	0	1	↑	0	1	↑	0	1	↑
Beam Continuity	0	0.4	2	2	=	2	2	=	2	2	=	2	2	=	2	2	=



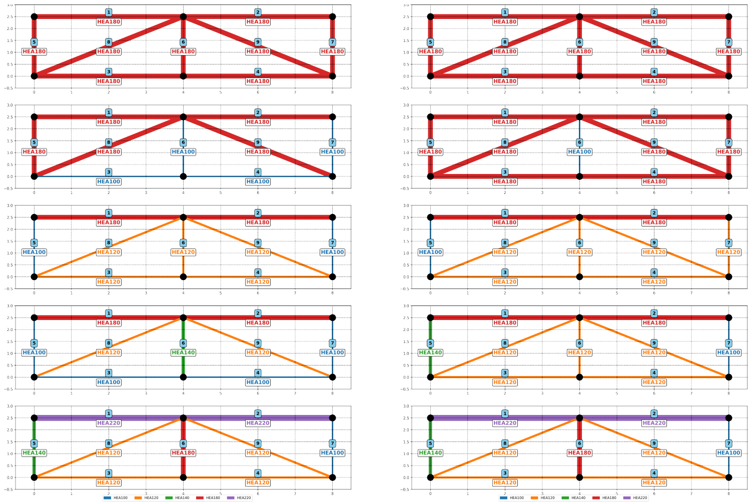
(a) TPE(*SYM*) truss configuration

(b) TPE($\mathcal{W}_3$) truss configuration

Figure E.11: Comparison between TPE(*SYM*) vs TPE($\mathcal{W}_3$) truss configurations.

Table E.12: Comparison of Tree-structured Parzen Estimator of Beam Continuity TPE(*BCN*) vs TPE($\mathcal{W}_3$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w(**TPE(*BCN*)**)	$\mathcal{W}_3$	**TPE(*BCN*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_3$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_3$)**	Δ
Mass	0	0.1	1169.86	1169.86	=	924.83	1122.74	↑	764.06	772.11	↑	750.25	784.04	↑	943	943	=
Connection Degree	0	0.4	6	6	=	10	8	↓	11	10	↓	13	11	↓	13	13	=
Symmetry	0	0.1	0	0	=	1	0	↓	0	1	↑	0	1	↑	1	1	=
Beam Continuity	1	0.4	2	2	=	2	2	=	2	2	=	2	2	=	2	2	=



(a) TPE(*BCN*) truss configuration

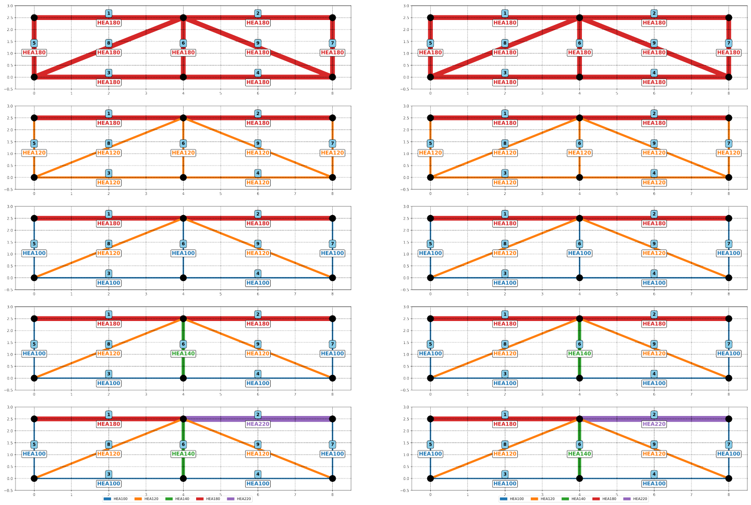
(b) TPE($\mathcal{W}_3$) truss configuration

Figure E.12: Comparison between TPE(*BCN*) vs TPE($\mathcal{W}_3$) truss configurations.

E.4. TPE($\mathcal{W}_4$) Comaprison with single obejective optimization TPE

Table E.13: Comparison of Tree-structured Parzen Estimator of Mass TPE(*M*) vs TPE($\mathcal{W}_4$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w**(TPE(*M*))**	$\mathcal{W}_4$	**TPE(*M*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*M*)**	**TPE($\mathcal{W}_4$)**	Δ
Mass	1	0.4	1169.86	1169.86	=	780.15	780.15	=	730.27	730.27	=	750.25	750.25	=	810.19	810.19	=
Connection Degree	0	0.05	6	6	=	9	9	=	12	12	=	13	13	=	14	14	=
Symmetry	0	0.5	0	0	=	0	0	=	0	0	=	0	0	=	1	1	=
Beam Continuity	0	0.05	2	2	=	2	2	=	2	2	=	2	2	=	3	3	=



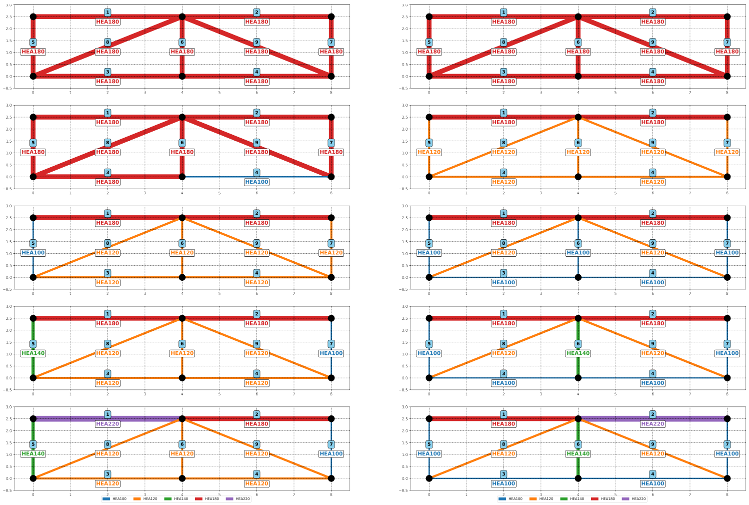
(a) TPE(*M*) truss configuration

(b) TPE($\mathcal{W}_4$) truss configuration

Figure E.13: Comparison between TPE(*M*) vs TPE($\mathcal{W}_4$) truss configurations.

Table E.14: Comparison of Tree-structured Parzen Estimator of Connection Degree TPE(*CN*) vs TPE($\mathcal{W}_4$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w(**TPE(*CN*)**)	$\mathcal{W}_4$	**TPE(*CN*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*CN*)**	**TPE($\mathcal{W}_4$)**	Δ
Mass	0	0.4	1169.86	1169.86	=	1094.46	780.15	↓	772.11	730.27	↓	784.04	750.25	↓	843.98	810.19	↓
Connection Degree	1	0.05	6	6	=	8	9	↑	10	12	↑	11	13	↑	12	14	↑
Symmetry	0	0.5	0	0	=	1	0	↓	1	0	↓	1	0	↓	2	1	↓
Beam Continuity	0	0.05	2	2	=	3	2	↓	2	2	=	2	2	=	3	3	=

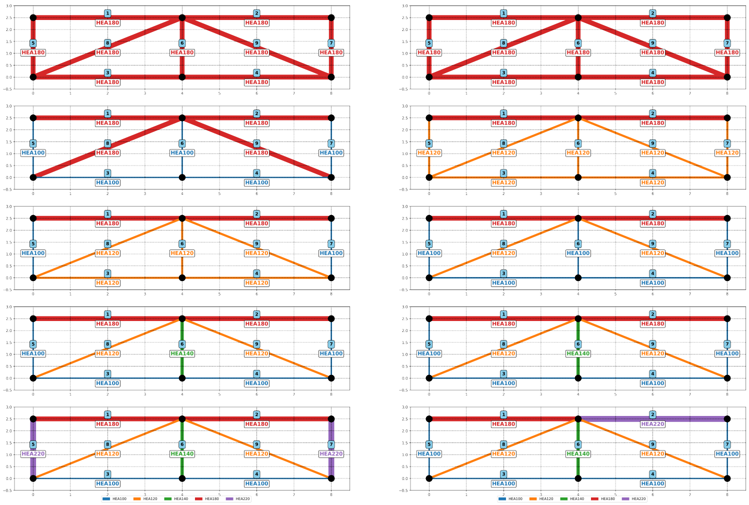


(a) TPE(*CN*) truss configuration **(b)** TPE($\mathcal{W}_4$) truss configuration

Figure E.14: Comparison between TPE(*CN*) vs TPE($\mathcal{W}_4$) truss configurations.

Table E.15: Comparison of Tree-structured Parzen Estimator of Symmetry TPE(*SYM*) vs TPE($\mathcal{W}_4$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w(**TPE(*SYM*)**)	$\mathcal{W}_4$	**TPE(*SYM*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*SYM*)**	**TPE($\mathcal{W}_4$)**	Δ
Mass	0	0.4	1169.86	1169.86	=	877.71	780.15	↓	764.06	730.27	↓	750.25	750.25	=	919.41	810.19	↓
Connection Degree	0	0.05	6	6	=	11	9	↓	11	12	↑	13	13	=	15	14	↓
Symmetry	1	0.5	0	0	=	0	0	=	0	0	=	0	0	=	0	1	↑
Beam Continuity	0	0.05	2	2	=	2	2	=	2	2	=	2	2	=	2	3	↑

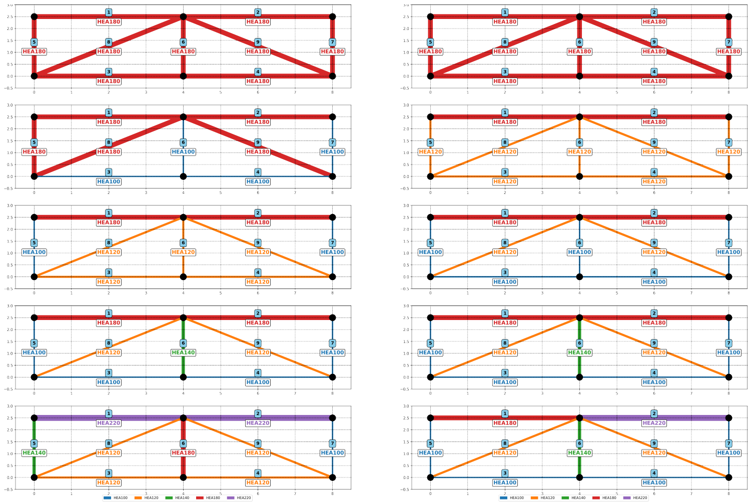


(a) TPE(*SYM*) truss configuration **(b)** TPE($\mathcal{W}_4$) truss configuration

Figure E.15: Comparison between TPE(*SYM*) vs TPE($\mathcal{W}_4$) truss configurations.

Table E.16: Comparison of Tree-structured Parzen Estimator of Beam Continuity TPE(*BCN*) vs TPE($\mathcal{W}_4$) for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

			$n=1$			$n=2$			$n=3$			$n=4$			$n=5$		
Objective	w(**TPE(*BCN*)**)	$\mathcal{W}_4$	**TPE(*BCN*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_4$)**	Δ	**TPE(*BCN*)**	**TPE($\mathcal{W}_4$)**	Δ
Mass	0	0.4	1169.86	1169.86	=	924.83	780.15	↓	764.06	730.27	↓	750.25	750.25	=	943	810.19	↓
Connection Degree	0	0.05	6	6	=	10	9	↓	11	12	↑	13	13	=	13	14	↑
Symmetry	0	0.5	0	0	=	1	0	↓	0	0	=	0	0	=	1	1	=
Beam Continuity	1	0.05	2	2	=	2	2	=	2	2	=	2	2	=	2	3	↑



(a) TPE(*BCN*) truss configuration

(b) TPE($\mathcal{W}_4$) truss configuration

Figure E.16: Comparison between TPE(*BCN*) vs TPE($\mathcal{W}_4$) truss configurations.

F Source Code

The full python code repository is available for download at **this link**.

This appendix provides the source code for the key components of the multi-objective truss optimization framework presented in Chapter 4.

```python
def total_objective(trial, choices_complexity, n_required, nodes_1, nodes_2, nodes_1_coord, nodes_2_coord, mass):
    """Computes four structural metrics (mass, connection degree, symmetry, beam continuity)
    for a truss configuration suggested by an Optuna trial. Applies a penalty if the number
    of distinct truss types is less than required.

    Args:
        trial (optuna.trial.Trial): Current optimization trial suggesting truss types.
        choices_complexity (list): List of valid truss type choices for each parameter.
        n_required (int): Required number of distinct truss types.
        nodes_1 (pd.DataFrame): DataFrame with first node identifiers for edges.
        nodes_2 (pd.DataFrame): DataFrame with second node identifiers for edges.
        nodes_1_coord (pd.DataFrame): Coordinates for first nodes.
        nodes_2_coord (pd.DataFrame): Coordinates for second nodes.
        mass (pd.DataFrame): Mass data for truss types.

    Returns:
        tuple: (graph_mass, connection_metric, symmetry_metric, beam_continuity_metric)
    """
    param_names = mass.iloc[:, 0].values.astype(str).flatten().tolist()
    truss_types_var = []
    for param_name, choice in zip(param_names, choices_complexity):
        filtered_choice = [choice_el for choice_el in choice if choice_el != 999]
        if len(filtered_choice) > 0:
            suggested = trial.suggest_categorical(f"s_{param_name}", filtered_choice)
            truss_types_var.append(suggested)

    truss_types_var_df = pd.DataFrame(truss_types_var, columns=["truss_type"])
    G, data_nodes = create_graph(nodes_1, nodes_2, nodes_1_coord, nodes_2_coord, mass, truss_types_var_df)

    connection_metric = get_connection_degree_metric(G)
    symmetry_metric = get_symmetric_metric(G)
    beam_continuity_metric = get_beam_continuity_metric(G)
    graph_mass = sum([data["mass"] for _, _, data in G.edges(data=True)])

    unique_truss_types = set(truss_types_var)
    n_distinct = len(unique_truss_types)
    if n_distinct < n_required:
        PENALTY_MULTIPLIER = 1e9
        penalty = (n_required - n_distinct) * PENALTY_MULTIPLIER
        graph_mass += penalty
        connection_metric += penalty
        symmetry_metric += penalty
        beam_continuity_metric += penalty

    return graph_mass, connection_metric, symmetry_metric, beam_continuity_metric
```

Listing F.1: Computing Truss Objective Metrics

```python
def run_total_tpe_optimizer(n, data_complexity, n_trials=100, nodes_1=None, nodes_2=None,
                            nodes_1_coord=None, nodes_2_coord=None, mass=None):
    """Runs the Tree-structured Parzen Estimator (TPE) optimization for a given number of
    unique truss profiles (n), evaluating multiple grouping configurations.

    Args:
        n (int): Number of unique truss profiles.
        data_complexity (pd.DataFrame): DataFrame with truss type complexity data.
        n_trials (int): Number of optimization trials.
        nodes_1 (pd.DataFrame): DataFrame with first node identifiers for edges.
        nodes_2 (pd.DataFrame): DataFrame with second node identifiers for edges.
        nodes_1_coord (pd.DataFrame): Coordinates for first nodes.
        nodes_2_coord (pd.DataFrame): Coordinates for second nodes.
        mass (pd.DataFrame): Mass data for truss types.

    Returns:
        tuple: (best_trial_values, best_trial_params) containing best metric values and parameters.
    """
    choices_list_complexity, _ = select_column_combinations(data_complexity.iloc[:, 1:], n)
    best_trial_values = []
    best_trial_params = []

    for i in range(len(choices_list_complexity)):
        sampler = TPESampler(
            consider_prior=False, consider_magic_clip=False, multivariate=True,
            group=True, consider_endpoints=False, constant_liar=False,
            n_startup_trials=100, n_ei_candidates=100, seed=42
        )
        study = optuna.create_study(sampler=sampler, directions=["minimize"] * 4)

        def objective_with_choice(trial):
            return total_objective(trial, choices_list_complexity[i], n,
                                   nodes_1, nodes_2, nodes_1_coord, nodes_2_coord, mass)

        study.optimize(objective_with_choice, n_trials=n_trials)
        best_trial_values += [trial.values for trial in study.best_trials]
        best_trial_params += [trial.params for trial in study.best_trials]

    return best_trial_values, best_trial_params
```

Listing F.2: Running TPE Optimization

```python
def compute_post_optimization_score(optimized_data_column, weights):
    """Computes a weighted score for optimized truss configurations based on stakeholder-defined
    weights, ranking solutions by their overall performance.

    Args:
        optimized_data_column (pd.DataFrame): DataFrame with optimization results (mass,
                                              connect_deg, symmetry, beam_cont).
        weights (list): Weights for each objective (mass, connect_deg, symmetry, beam_cont).

    Returns:
        pd.DataFrame: DataFrame with the best configuration based on weighted score.
    """
    scaled_values = optimized_data_column[["mass", "connect_deg", "symmetry", "beam_cont"]].apply(minmax_scale)
    score = (scaled_values * weights).sum(axis=1)
    optimized_data_column["weighted_score"] = score
    return optimized_data_column.loc[[score.idxmin()]]
```

Listing F.3: Computing Weighted Post-Optimization Score

```python
def create_graph(data_nodes_1, data_nodes_2, data_nodes_1_coord, data_nodes_2_coord, mass, truss_types_data=None):
    """Constructs a NetworkX graph representing a steel truss, incorporating node positions
    and edge attributes (element ID, mass, truss type).

    Args:
        data_nodes_1 (pd.DataFrame): DataFrame with edge and first node identifiers.
        data_nodes_2 (pd.DataFrame): DataFrame with second node identifiers.
        data_nodes_1_coord (pd.DataFrame): Coordinates for first nodes.
        data_nodes_2_coord (pd.DataFrame): Coordinates for second nodes.
        mass (pd.DataFrame): Mass data for truss types.
        truss_types_data (pd.DataFrame, optional): Truss type information for edges.

    Returns:
        tuple: (G, data_nodes) where G is the NetworkX graph and data_nodes is the processed DataFrame.
    """
    G = nx.Graph()
    data_nodes = pd.concat([
        data_nodes_1.iloc[:, :2], data_nodes_1_coord.iloc[:, 1],
        data_nodes_2.iloc[:, 1], data_nodes_2_coord.iloc[:, 1]
    ], axis=1)
    data_nodes = pd.concat([data_nodes, truss_types_data], axis=1)
    data_nodes.columns = ["edge", "node_1", "coord_node_1", "node_2", "coord_node_2", "truss_type"]

    unique_nodes = pd.concat([data_nodes["node_1"], data_nodes["node_2"]]).drop_duplicates().values
    unique_coordinates = pd.concat([data_nodes["coord_node_1"], data_nodes["coord_node_2"]]).drop_duplicates().values
    unique_coordinates = [
        coordinates_tuple.replace(',', '.').replace('{', '').replace('}', '').split("_")
        for coordinates_tuple in unique_coordinates
    ]
    unique_coordinates = [(float(coord[0]), float(coord[2])) for coord in unique_coordinates]

    for node, coordinates in zip(unique_nodes, unique_coordinates):
        G.add_node(node, pos=coordinates)

    for _, row in data_nodes.iterrows():
        if pd.notna(row["truss_type"]):
            G.add_edge(
                row["node_1"], row["node_2"],
                element_id=row["edge"],
                mass=mass.loc[int(row["edge"]) - 1, row["truss_type"]],
                truss_type=row["truss_type"]
            )
    return G, data_nodes
```

Listing F.4: Constructing Truss Graph

```python
def get_connection_degree_metric(G):
    """Computes the connection degree metric by summing the number of unique truss types
    connected to each node in the truss graph.

    Args:
        G (networkx.Graph): Truss graph with edges containing 'truss_type' attributes.

    Returns:
        int: Total connection degree metric.
    """
    connection_degree_dict = defaultdict(int)
    for node in G.nodes():
        temp = G.edges(node, data=True)
        connection_degree_dict[node] = len({edge[2]["truss_type"] for edge in temp if "truss_type" in edge[2]})
    return sum(connection_degree_dict.values())
```

Listing F.5: Computing Connection Degree Metric

```python
def get_symmetric_metric(G):
    """Computes the symmetry metric by counting edges with different truss types between
    symmetric node pairs, reflecting structural symmetry.

    Args:
        G (networkx.Graph): Truss graph with node positions and edge truss types.

    Returns:
        int: Symmetry metric value.
    """
    symmetric_nodes = find_symmetric_nodes(G)
    symmetric_elements = find_symmetric_elements(G, symmetric_nodes)
    symmetric_metric = sum([1 if symmetric_element[0] != symmetric_element[1] else 0
                            for symmetric_element in symmetric_elements])
    return symmetric_metric
```

Listing F.6: Computing Symmetry Metric

```python
def get_beam_continuity_metric(G):
    """Computes the beam continuity metric by analyzing truss type consistency in
    horizontal beams at the top and bottom of the truss.

    Args:
        G (networkx.Graph): Truss graph with node positions and edge truss types.

    Returns:
        int: Beam continuity metric value.
    """
    top_left_node = min(G.nodes(data=True), key=lambda x: (x[1]['pos'][0], -x[1]['pos'][1]))
    bottom_left_node = min(G.nodes(data=True), key=lambda x: (x[1]['pos'][0], x[1]['pos'][1])
        )
    top_beam = iterate_over_beam(G, top_left_node)
    bottom_beam = iterate_over_beam(G, bottom_left_node)

    beam_continuity_metric = 2
    for i in range(1, len(top_beam)):
        if top_beam[i-1] != top_beam[i]:
            beam_continuity_metric += 1
        if bottom_beam[i-1] != bottom_beam[i]:
            beam_continuity_metric += 1
    return beam_continuity_metric
```

Listing F.7: Computing Beam Continuity Metric

G

Truss typology 1

This appendix presents the results of TPE$_{t1}$ optimization for a truss Typology 1, evaluated across (n from 1 to 5), using four different weight distributions: TPE$_{t1}$($\mathcal{W}_1$), TPE$_{t1}$($\mathcal{W}_2$), TPE$_{t1}$(*M*) and TPE$_{t1}$(*SYM*) as represented in Table G.1. The analysis investigates the impact of each of the selected weight distribution on the four target objectives mass, connection degree, symmetry, and beam continuity under a fixed ($N_{\text{trials}} = 2000$).

Table G.1: Comparison of different weights combination for TPE$_{t1}$ distributions in parametric truss optimization

	w_M	w_{CN}	w_{SYM}	w_{BCN}
TPE$_{t1}$($\mathcal{W}_1$)	0.25	0.25	0.25	0.25
TPE$_{t1}$($\mathcal{W}_2$)	0.4	0.1	0.4	0.1
TPE$_{t1}$(*M*)	1	0	0	0
TPE$_{t1}$(*SYM*)	0	0	1	0

Table G.2: TPE$_{t1}$($\mathcal{W}_1$) results for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

		$n=1$	$n=2$	$n=3$	$n=4$	$n=5$
Objective	$\mathcal{W}_1$	**TPE$_{t1}$($\mathcal{W}_1$)**	**TPE$_{t1}$($\mathcal{W}_1$)**	**TPE$_{t1}$($\mathcal{W}_1$)**	**TPE$_{t1}$($\mathcal{W}_1$)**	**TPE$_{t1}$($\mathcal{W}_1$)**
Mass	0.25	1287.38	1275.38	1084.89	1054.46	1126.58
Connection Degree	0.25	10	12	17	20	20
Symmetry	0.25	0	0	0	1	2
Beam Continuity	0.25	2	2	2	2	2

Table G.3: TPE$_{t1}$($\mathcal{W}_2$) results for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

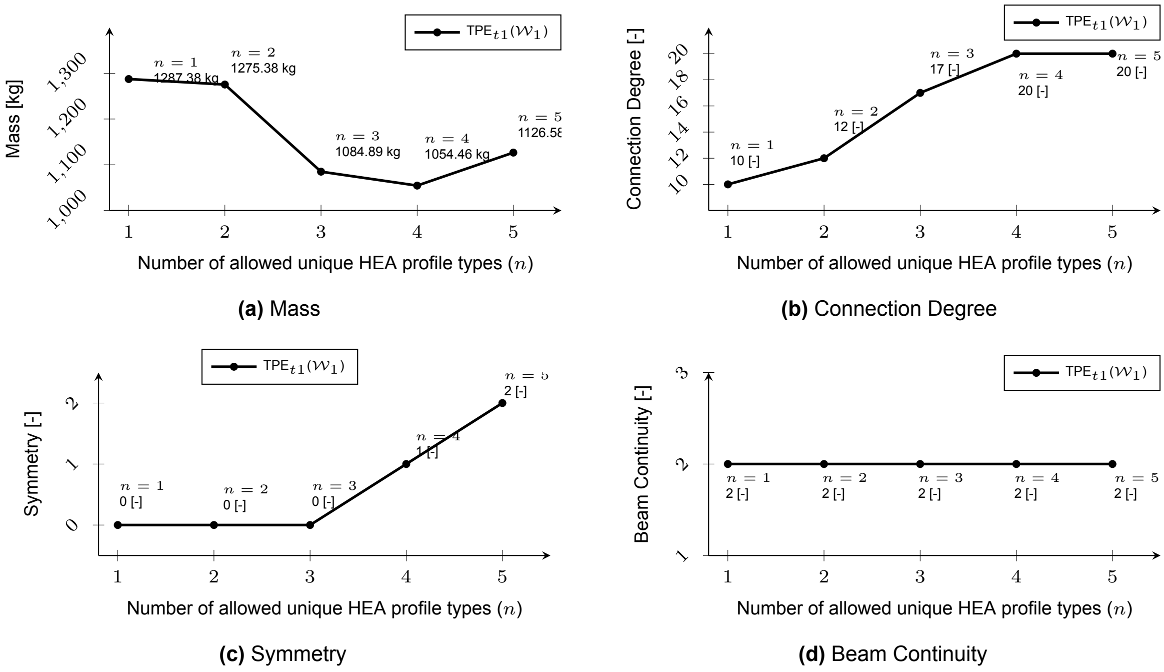
		$n=1$	$n=2$	$n=3$	$n=4$	$n=5$
Objective	$\mathcal{W}_2$	**TPE$_{t1}$($\mathcal{W}_2$)**	**TPE$_{t1}$($\mathcal{W}_2$)**	**TPE$_{t1}$($\mathcal{W}_2$)**	**TPE$_{t1}$($\mathcal{W}_2$)**	**TPE$_{t1}$($\mathcal{W}_2$)**
Mass	0.4	1287.38	1068.89	991.41	1065.66	1098.46
Connection Degree	0.1	10	15	19	19	23
Symmetry	0.4	0	0	0	0	1
Beam Continuity	0.1	2	2	2	4	4

Table G.4: TPE$_{t1}$(*M*) results for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

		$n=1$	$n=2$	$n=3$	$n=4$	$n=5$
Objective	$\mathcal{W}_M$	**TPE$_{t1}$(*M*)**	**TPE$_{t1}$(*M*)**	**TPE$_{t1}$(*M*)**	**TPE$_{t1}$(*M*)**	**TPE$_{t1}$(*M*)**
Mass	1	1287.38	1028.89	962.61	1010.11	1057.75
Connection Degree	0	10	17	19	20	21
Symmetry	0	0	0	0	2	4
Beam Continuity	0	2	2	4	4	5

Table G.5: TPE$_{t1}$(*SYM*) results for parametric truss optimization across $n = 1$ to 5, $N_{\text{trials}} = 2000$.

		$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$
Objective	$\mathcal{W}_{SYM}$	**TPE$_{t1}$(*SYM*)**	**TPE$_{t1}$(*SYM*)**	**TPE$_{t1}$(*SYM*)**	**TPE$_{t1}$(*SYM*)**	**TPE$_{t1}$(*SYM*)**
Mass	0	1287.38	1156.89	1018.61	1073.61	1172.71
Connection Degree	0	10	13	17	23	27
Symmetry	1	0	0	0	0	0
Beam Continuity	0	2	4	4	4	4



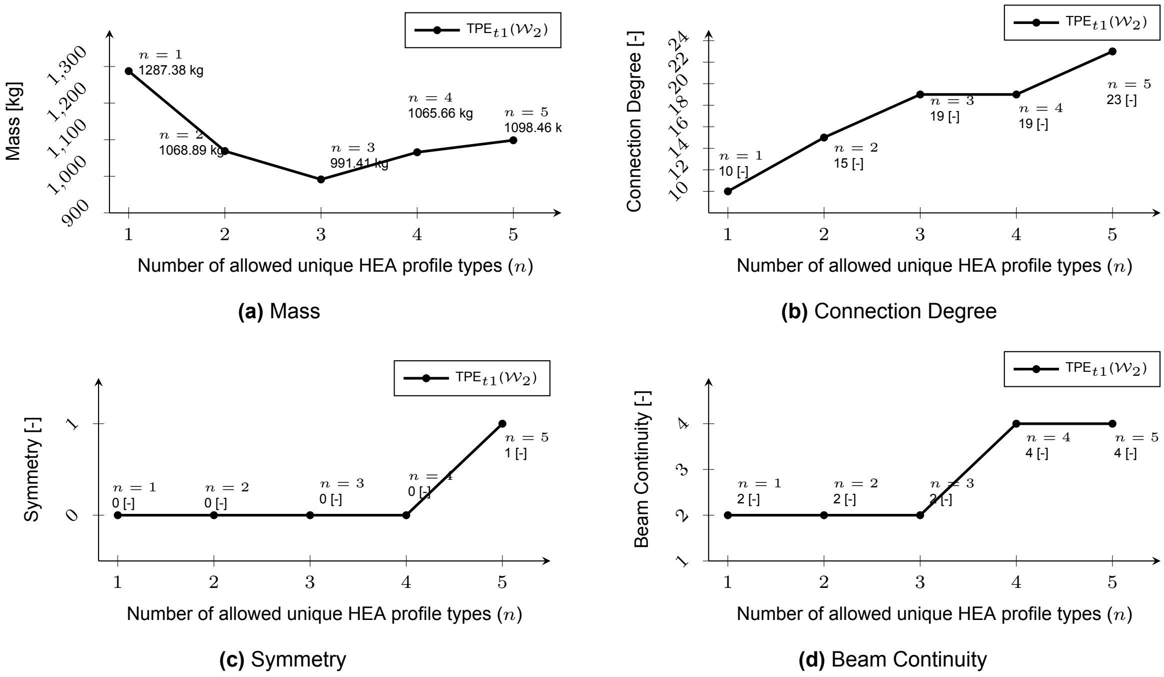
(a) Mass

(b) Connection Degree

(c) Symmetry

(d) Beam Continuity

Figure G.1: Results of TPE$_{t1}(\mathcal{W}_1)$ Optimization across Objectives for $N_{\text{trials}} = 2000$.



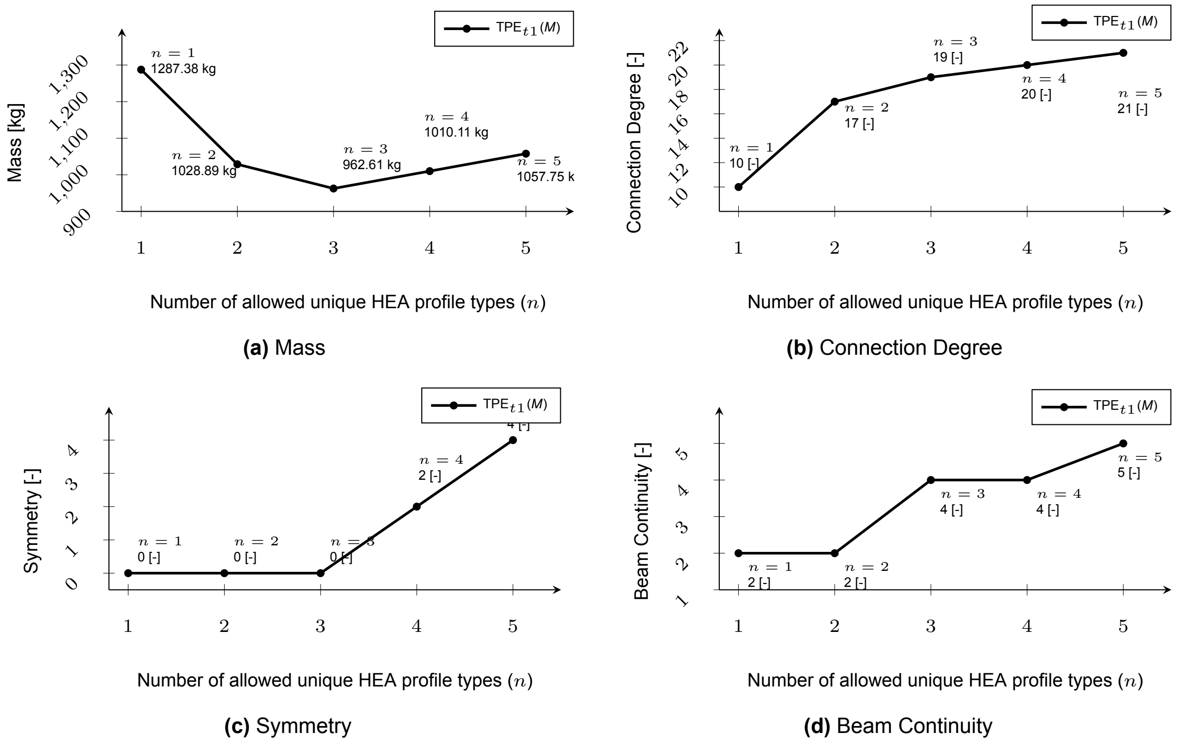
(a) Mass

(b) Connection Degree

(c) Symmetry

(d) Beam Continuity

Figure G.2: Results of TPE$_{t1}(\mathcal{W}_2)$ Optimization across Objectives for $N_{\text{trials}} = 2000$.



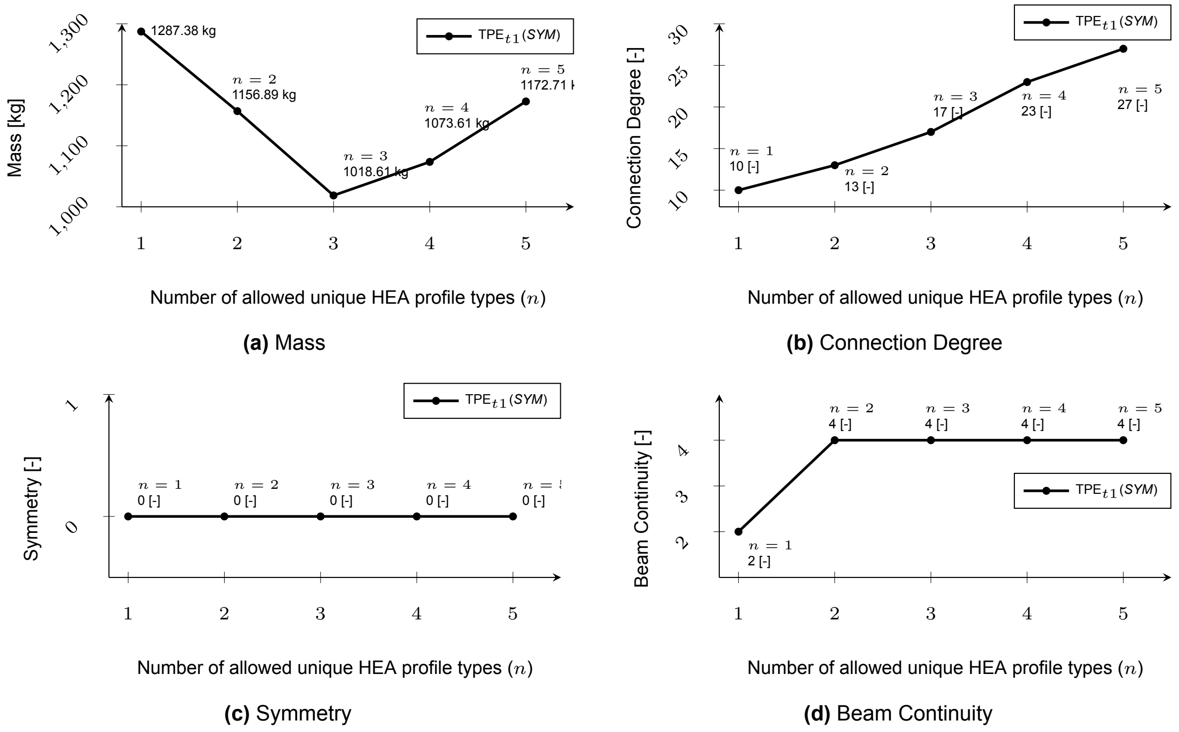
(a) Mass

(b) Connection Degree

(c) Symmetry

(d) Beam Continuity

Figure G.3: Results of TPE$_{t1}$(*M*) Optimization across Objectives for $N_{\text{trials}} = 2000$.



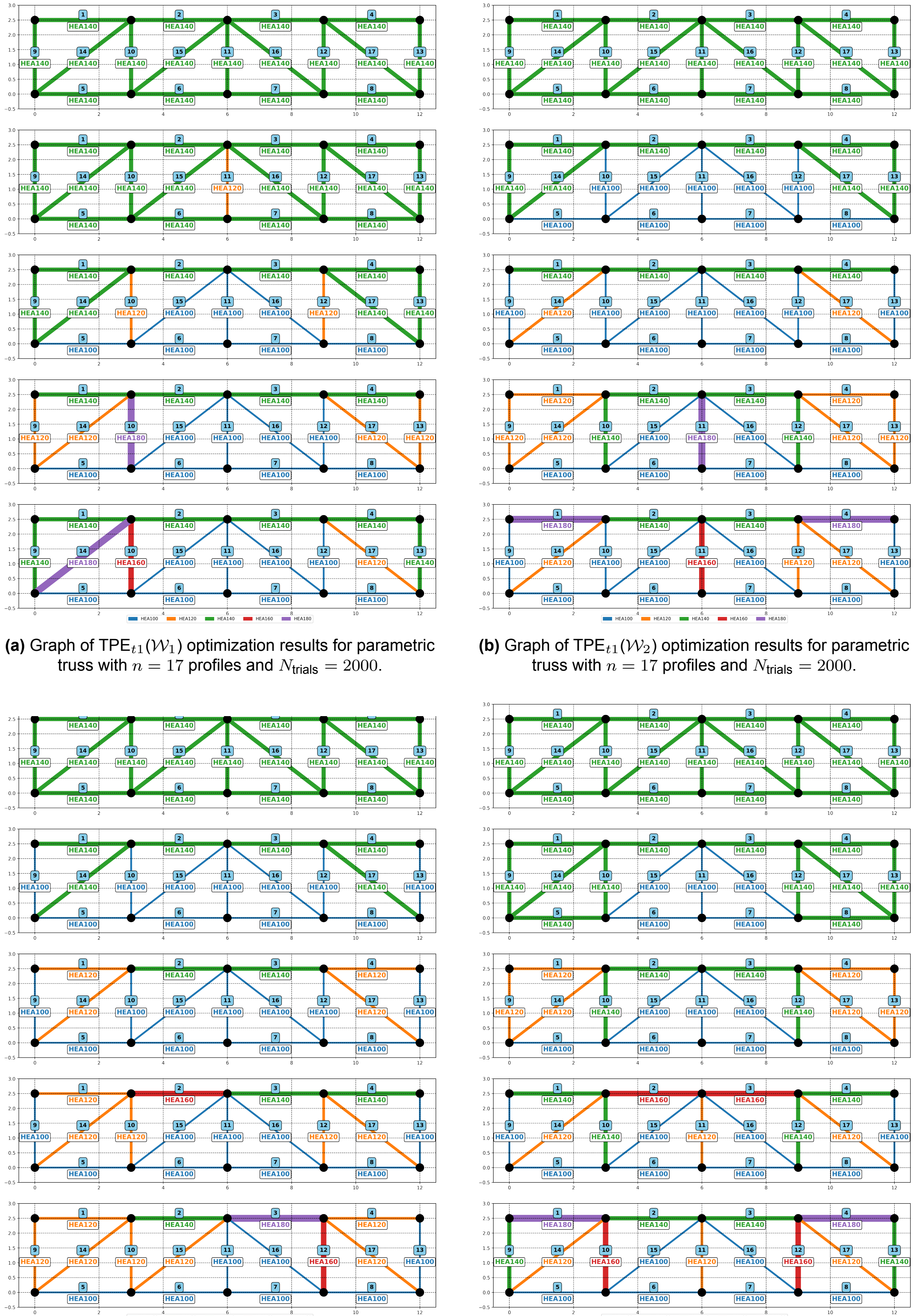
(a) Mass

(b) Connection Degree

(c) Symmetry

(d) Beam Continuity

Figure G.4: Results of TPE$_{t1}$(*SYM*) Optimization across Objectives for $N_{\text{trials}} = 2000$.



(a) Graph of TPE$_{t1}(\mathcal{W}_1)$ optimization results for parametric truss with $n = 17$ profiles and $N_{\text{trials}} = 2000$.

(b) Graph of TPE$_{t1}(\mathcal{W}_2)$ optimization results for parametric truss with $n = 17$ profiles and $N_{\text{trials}} = 2000$.

(c) Graph of TPE$_{t1}$(*M*) optimization results for parametric truss with $n = 17$ profiles and $N_{\text{trials}} = 2000$.

(d) Graph of TPE$_{t1}$(*SYM*) optimization results for parametric truss with $n = 17$ profiles and $N_{\text{trials}} = 2000$.

Figure G.5: Graph of TPE$_{t1}(\mathcal{W}_1)(\mathcal{W}_2)$(*M*)(*SYM*) optimization results for parametric truss with $n = 17$ profiles and $N_{\text{trials}} = 2000$.

H

Truss typology 2

This appendix presents the results of TPE$_{t2}$ optimization for a truss Typology 2, evaluated across (n from 1 to 6), using four different weight distributions: TPE$_{t2}(\mathcal{W}_1)$, TPE$_{t2}(\mathcal{W}_2)$, TPE$_{t2}$(*M*) and TPE$_{t2}$(*SYM*) as represented in Table H.1. The analysis investigates the impact of each of the selected weight distribution on the four target objectives mass, connection degree, symmetry, and beam continuity under a fixed ($N_{\text{trials}} = 2000$).

Table H.1: Comparison of different weights combination for TPE$_{t2}$ distributions in parametric truss optimization

	w_M	w_{CN}	w_{SYM}	w_{BCN}
TPE$_{t2}$($\mathcal{W}_1$)	0.25	0.25	0.25	0.25
TPE$_{t2}$($\mathcal{W}_2$)	0.4	0.1	0.4	0.1
TPE$_{t2}$(*M*)	1	0	0	0
TPE$_{t2}$(*SYM*)	0	0	1	0

Table H.2: TPE$_{t2}(\mathcal{W}_1)$ results for parametric truss optimization across $n = 1$ to 6, $N_{\text{trials}} = 2000$.

		$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
Objective	$\mathcal{W}_1$	**TPE$_{t2}$($\mathcal{W}_1$)**	**TPE$_{t2}$($\mathcal{W}_1$)**	**TPE$_{t2}$($\mathcal{W}_1$)**	**TPE$_{t2}$($\mathcal{W}_1$)**	**TPE$_{t2}$($\mathcal{W}_1$)**	**TPE$_{t2}$($\mathcal{W}_1$)**
Mass	0.25	1001.19	966.88	666.98	850.77	751.79	833.45
Connection Degree	0.25	6	8	10	11	13	15
Symmetry	0.25	0	0	0	1	2	2
Beam Continuity	0.25	2	2	2	2	2	2

Table H.3: TPE$_{t2}(\mathcal{W}_2)$ results for parametric truss optimization across $n = 1$ to 6, $N_{\text{trials}} = 2000$.

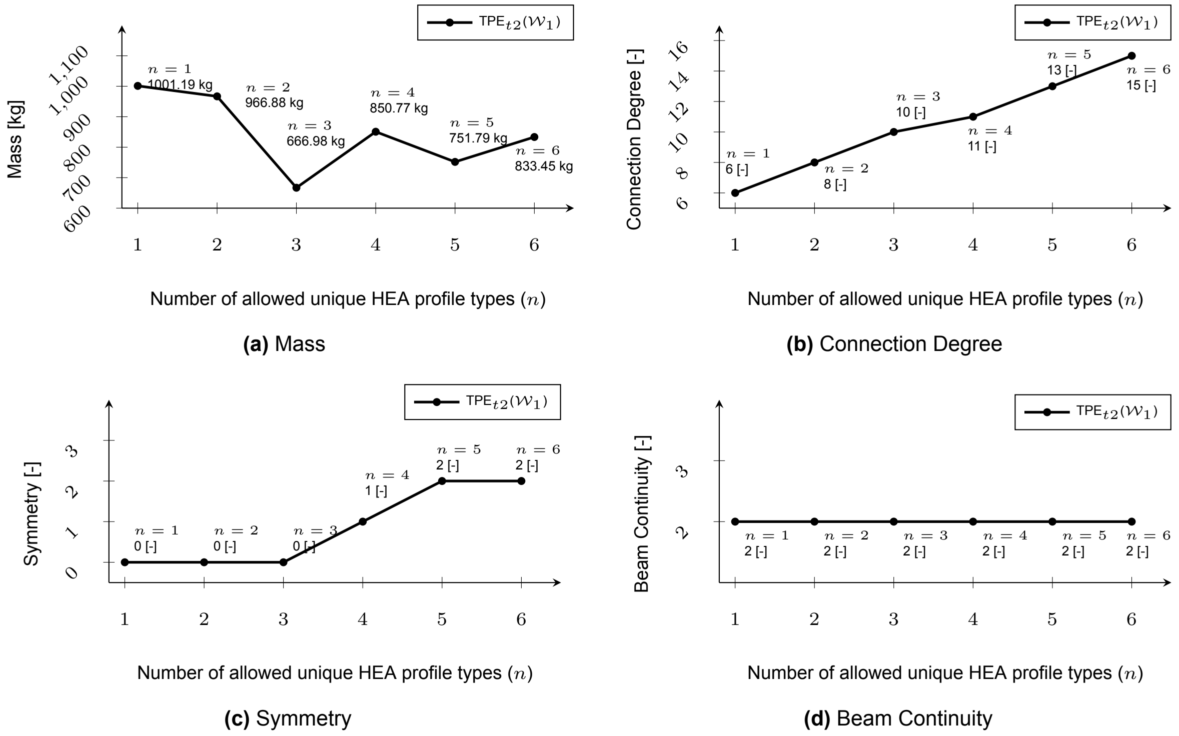
		$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
Objective	$\mathcal{W}_2$	**TPE$_{t2}$($\mathcal{W}_2$)**	**TPE$_{t2}$($\mathcal{W}_2$)**	**TPE$_{t2}$($\mathcal{W}_2$)**	**TPE$_{t2}$($\mathcal{W}_2$)**	**TPE$_{t2}$($\mathcal{W}_2$)**	**TPE$_{t2}$($\mathcal{W}_2$)**
Mass	0.4	1001.19	658.93	666.98	704.66	734.08	845.99
Connection Degree	0.1	6	9	10	13	14	16
Symmetry	0.4	0	0	0	0	1	1
Beam Continuity	0.1	2	2	2	2	2	3

Table H.4: TPE$_{t2}$(*M*) results for parametric truss optimization across $n = 1$ to 6, $N_{\text{trials}} = 2000$.

		$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
Objective	$\mathcal{W}_M$	**TPE$_{t2}$(*M*)**	**TPE$_{t2}$(*M*)**	**TPE$_{t2}$(*M*)**	**TPE$_{t2}$(*M*)**	**TPE$_{t2}$(*M*)**	**TPE$_{t2}$(*M*)**
Mass	1	1001.19	658.93	666.98	686.96	707.44	771.51
Connection Degree	0	6	9	11	12	13	15
Symmetry	0	0	0	1	1	2	2
Beam Continuity	0	2	2	2	2	3	3

Table H.5: TPE$_{t2}$(*SYM*) results for parametric truss optimization across $n = 1$ to 6, $N_{\text{trials}} = 2000$.

		$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$
Objective	$\mathcal{W}_{SYM}$	**TPE$_{t2}$(*SYM*)**	**TPE$_{t2}$(*SYM*)**	**TPE$_{t2}$(*SYM*)**	**TPE$_{t2}$(*SYM*)**	**TPE$_{t2}$(*SYM*)**	**TPE$_{t2}$(*SYM*)**
Mass	0	1001.19	966.88	666.98	695	803.02	933.54
Connection Degree	0	6	8	10	14	15	16
Symmetry	1	0	0	0	0	0	1
Beam Continuity	0	2	2	2	2	2	2



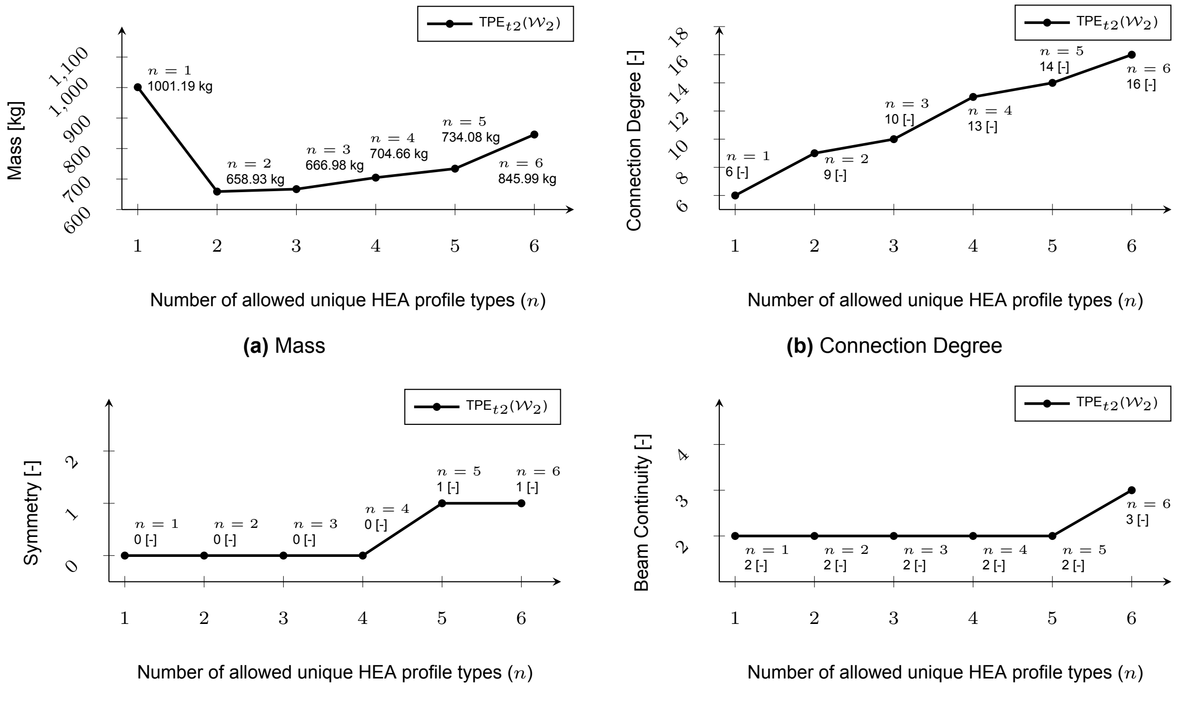
(a) Mass

(b) Connection Degree

(c) Symmetry

(d) Beam Continuity

Figure H.1: Results of TPE$_{t2}$($\mathcal{W}_1$) Optimization across Objectives for $N_{\text{trials}} = 2000$.



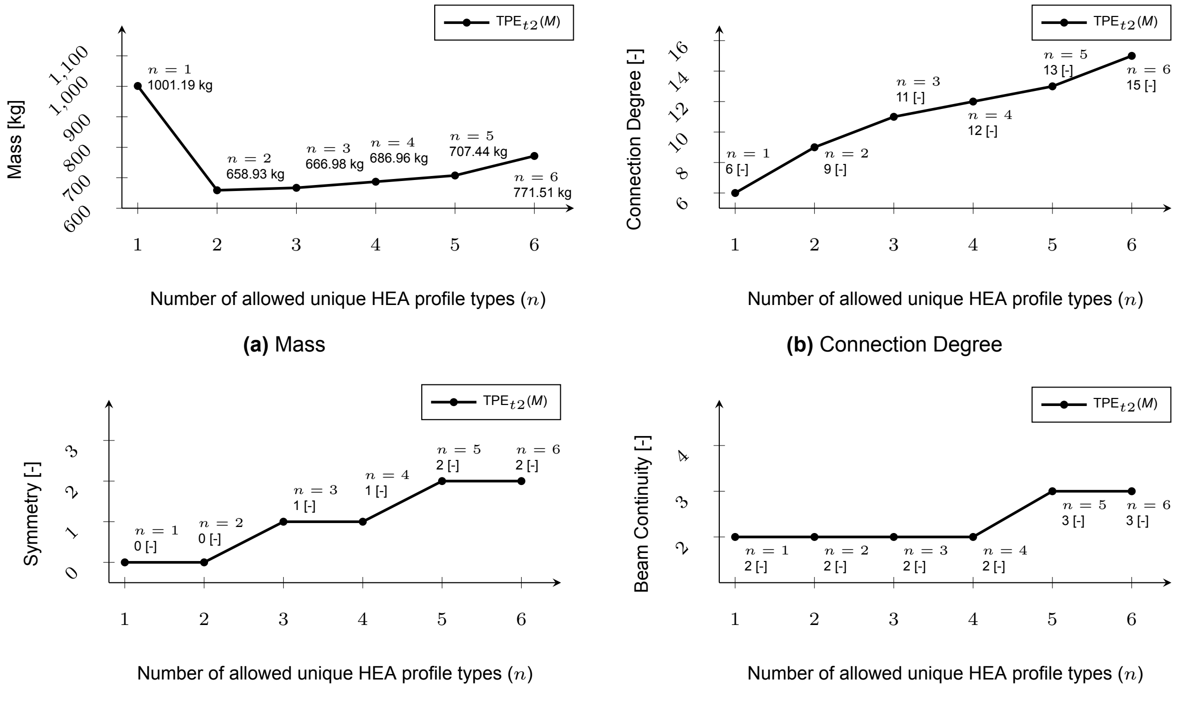
(a) Mass

(b) Connection Degree

(c) Symmetry

(d) Beam Continuity

Figure H.2: Results of TPE$_{t2}(\mathcal{W}_2)$ Optimization across Objectives for $N_{\text{trials}} = 2000$.



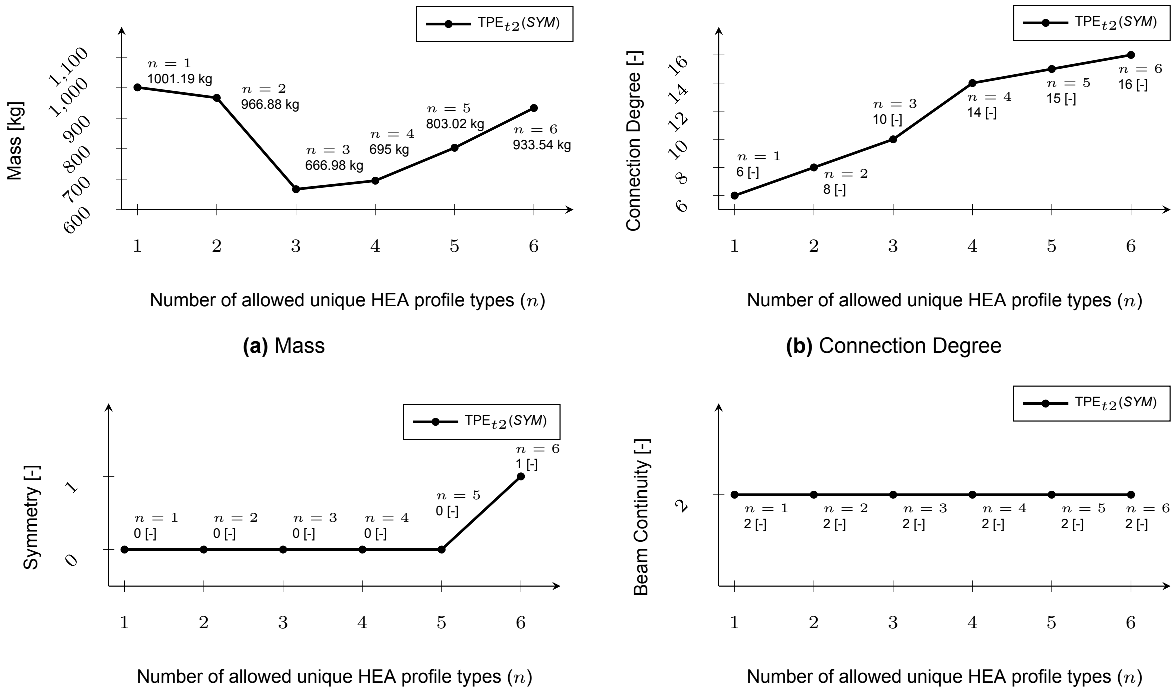
(a) Mass

(b) Connection Degree

(c) Symmetry

(d) Beam Continuity

Figure H.3: Results of TPE$_{t2}(M)$ Optimization across Objectives for $N_{\text{trials}} = 2000$.



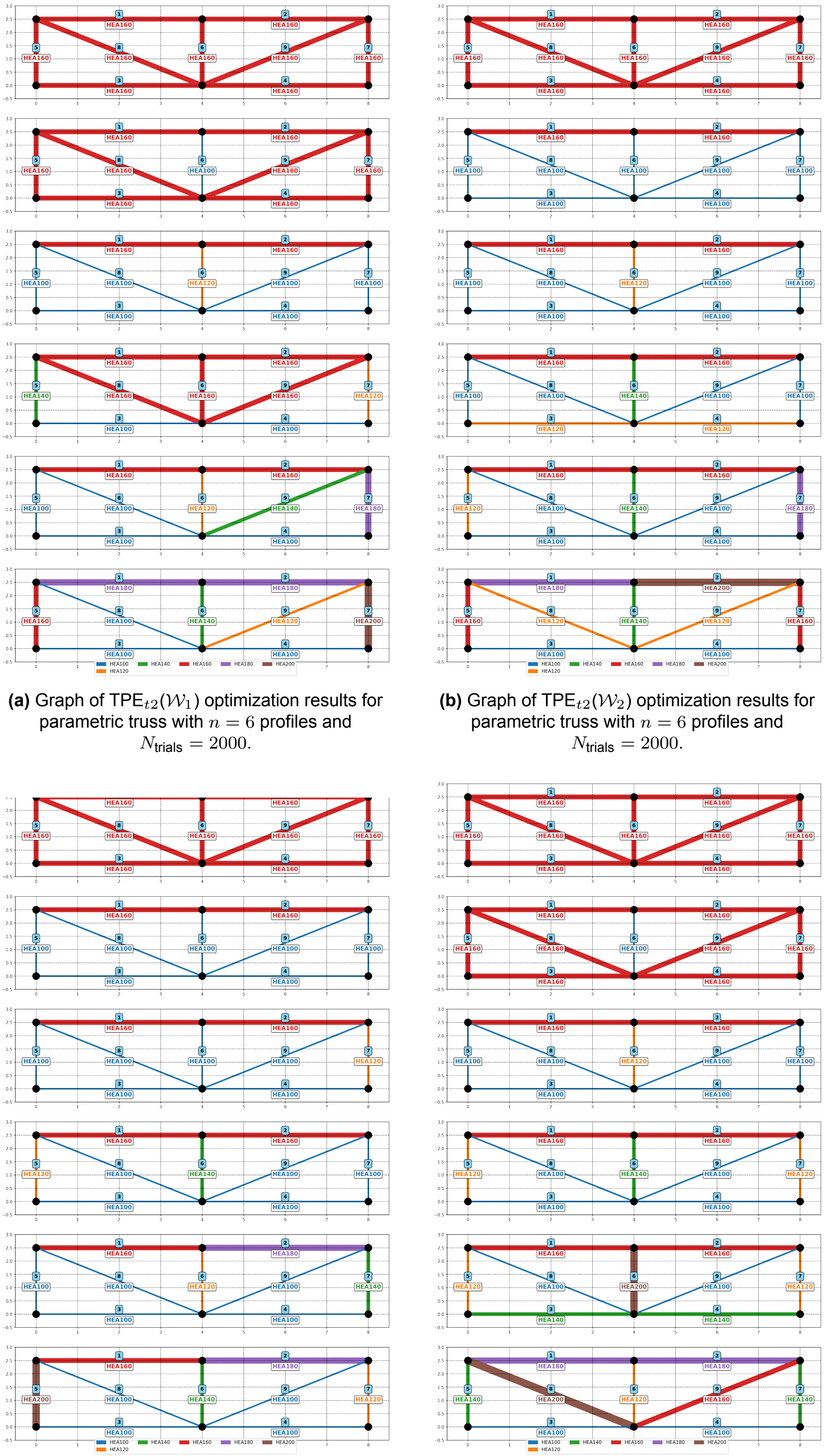
(a) Mass

(b) Connection Degree

(c) Symmetry

(d) Beam Continuity

Figure H.4: Results of TPE$_{t2}$(*SYM*) Optimization across Objectives for $N_{\text{trials}} = 2000$.



(a) Graph of TPE$_{t2}$($\mathcal{W}_1$) optimization results for parametric truss with $n = 6$ profiles and $N_{\text{trials}} = 2000$.

(b) Graph of TPE$_{t2}$($\mathcal{W}_2$) optimization results for parametric truss with $n = 6$ profiles and $N_{\text{trials}} = 2000$.

(c) Graph of TPE$_{t2}$(*M*) optimization results for parametric truss with $n = 6$ profiles and $N_{\text{trials}} = 2000$.

(d) Graph of TPE$_{t2}$(*SYM*) optimization results for parametric truss with $n = 6$ profiles and $N_{\text{trials}} = 2000$.

Figure H.5: Graph of TPE$_{t2}$($\mathcal{W}_1$)($\mathcal{W}_2$)(*M*)(*SYM*) optimization results for parametric truss with $n = 6$ profiles and $N_{\text{trials}} = 2000$.

I

Truss typology 3

This appendix presents the results of TPE$_{t3}$ optimization for a truss Typology 3, evaluated across (n from 1 to 8), using four different weight distributions: TPE$_{t3}(\mathcal{W}_1)$, TPE$_{t3}(\mathcal{W}_2)$, TPE$_{t3}$(*M*) and TPE$_{t3}$(*SYM*) as represented in Table I.1. The analysis investigates the impact of each of the selected weight distribution on the four target objectives mass, connection degree, symmetry, and beam continuity under a fixed ($N_{\text{trials}} = 2000$).

Table I.1: Comparison of different weights combination for TPE$_{t3}$ distributions in parametric truss optimization

	w_M	w_{CN}	w_{SYM}	w_{BCN}
TPE$_{\mathbf{t3}}(\boldsymbol{\mathcal{W}_1})$	0.25	0.25	0.25	0.25
TPE$_{\mathbf{t3}}(\boldsymbol{\mathcal{W}_2})$	0.4	0.1	0.4	0.1
TPE$_{\mathbf{t3}}$(*M*)	1	0	0	0
TPE$_{\mathbf{t3}}$(*SYM*)	0	0	1	0

Table I.2: TPE$_{t3}(\mathcal{W}_1)$ results for parametric truss optimization across $n = 1$ to 8, $N_{\text{trials}} = 2000$.

		$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$
Objective	$\boldsymbol{\mathcal{W}_1}$	**TPE$_{\mathbf{t3}}(\boldsymbol{\mathcal{W}_1})$**	**TPE$_{\mathbf{t3}}(\boldsymbol{\mathcal{W}_1})$**	**TPE$_{\mathbf{t3}}(\boldsymbol{\mathcal{W}_1})$**	**TPE$_{\mathbf{t3}}(\boldsymbol{\mathcal{W}_1})$**	**TPE$_{\mathbf{t3}}(\boldsymbol{\mathcal{W}_1})$**	**TPE$_{\mathbf{t3}}(\boldsymbol{\mathcal{W}_1})$**	**TPE$_{\mathbf{t3}}(\boldsymbol{\mathcal{W}_1})$**	**TPE$_{\mathbf{t3}}(\boldsymbol{\mathcal{W}_1})$**
Mass	0.25	1584.46	1550.21	1042.81	1089.86	1328.41	1308.48	1423.19	1800.29
Connection Degree	0.25	10	12	16	18	21	21	23	27
Symmetry	0.25	0	0	0	1	0	3	3	3
Beam Continuity	0.25	2	2	2	2	2	2	3	4

Table I.3: TPE$_{t3}(\mathcal{W}_2)$ results for parametric truss optimization across $n = 1$ to 8, $N_{\text{trials}} = 2000$.

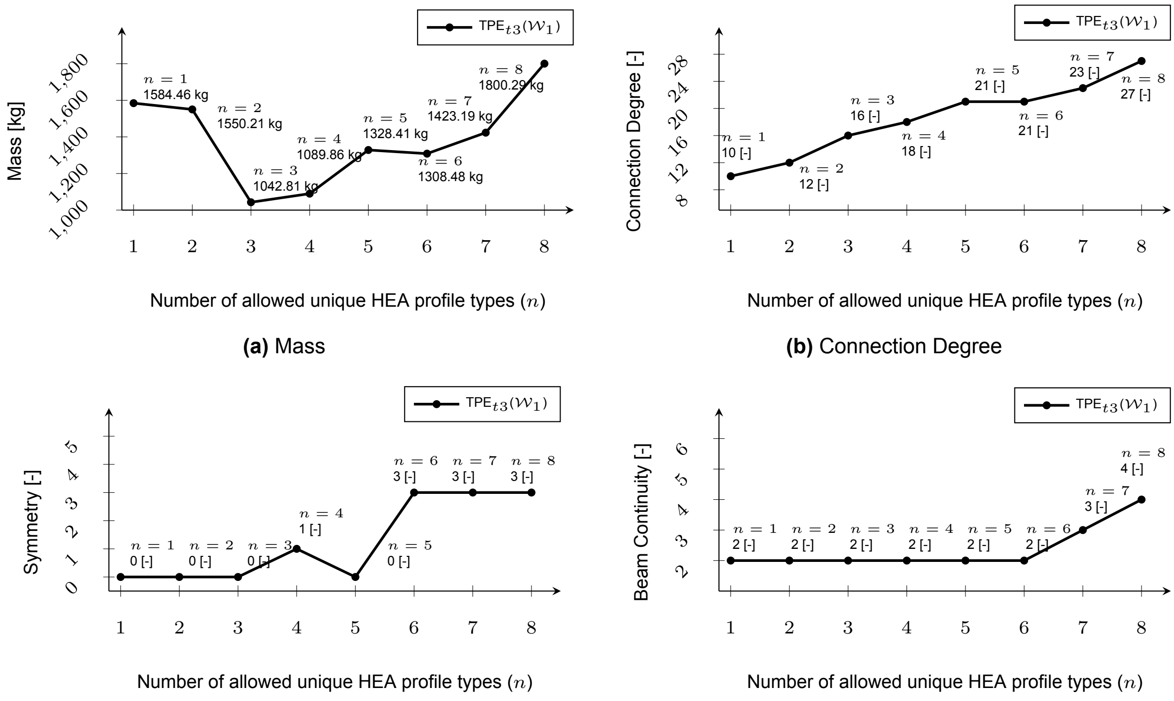
		$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$
Objective	$\boldsymbol{\mathcal{W}_2}$	**TPE$_{\mathbf{t3}}(\boldsymbol{\mathcal{W}_2})$**	**TPE$_{\mathbf{t3}}(\boldsymbol{\mathcal{W}_3})$**	**TPE$_{\mathbf{t3}}(\boldsymbol{\mathcal{W}_2})$**	**TPE$_{\mathbf{t3}}(\boldsymbol{\mathcal{W}_2})$**	**TPE$_{\mathbf{t3}}(\boldsymbol{\mathcal{W}_2})$**	**TPE$_{\mathbf{t3}}(\boldsymbol{\mathcal{W}_2})$**	**TPE$_{\mathbf{t3}}(\boldsymbol{\mathcal{W}_2})$**	**TPE$_{\mathbf{t3}}(\boldsymbol{\mathcal{W}_2})$**
Mass	0.4	1584.46	1034.81	1042.81	1184.51	1232.31	1311.79	1423.19	1459.6
Connection Degree	0.1	10	15	16	20	24	23	23	28
Symmetry	0.4	0	0	0	0	0	1	3	4
Beam Continuity	0.1	2	2	2	2	2	4	3	5

Table I.4: TPE$_{t3}$(*M*) results for parametric truss optimization across $n = 1$ to 8, $N_{\text{trials}} = 2000$.

		$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$
Objective	$\boldsymbol{\mathcal{W}_M}$	**TPE$_{\mathbf{t3}}$(*M*)**	**TPE$_{\mathbf{t3}}$(*M*)**	**TPE$_{\mathbf{t3}}$(*M*)**	**TPE$_{\mathbf{t3}}$(*M*)**	**TPE$_{\mathbf{t3}}$(*M*)**	**TPE$_{\mathbf{t3}}$(*M*)**	**TPE$_{\mathbf{t3}}$(*M*)**	**TPE$_{\mathbf{t3}}$(*M*)**
Mass	1	1584.46	1034.81	1042.81	1066.11	1122.25	1195.91	1297.04	1420.01
Connection Degree	0	10	15	16	21	20	23	26	27
Symmetry	0	0	0	0	3	2	4	4	5
Beam Continuity	0	2	2	2	4	3	5	4	6

Table I.5: TPE$_{t3}$(*SYM*) results for parametric truss optimization across $n = 1$ to 8, $N_{\text{trials}} = 2000$.

		$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$
Objective	$\mathcal{W}_{SYM}$	**TPE$_{t3}$(*SYM*)**	**TPE$_{t3}$(*SYM*)**	**TPE$_{t3}$(*SYM*)**	**TPE$_{t3}$(*SYM*)**	**TPE$_{t3}$(*SYM*)**	**TPE$_{t3}$(*SYM*)**	**TPE$_{t3}$(*SYM*)**	**TPE$_{t3}$(*SYM*)**
Mass	0	1584.46	1034.81	1042.81	1326.24	1232.31	1785.14	1798.25	1674.46
Connection Degree	0	10	15	16	20	24	28	28	26
Symmetry	1	0	0	0	0	0	0	1	3
Beam Continuity	0	2	2	2	4	2	6	4	6



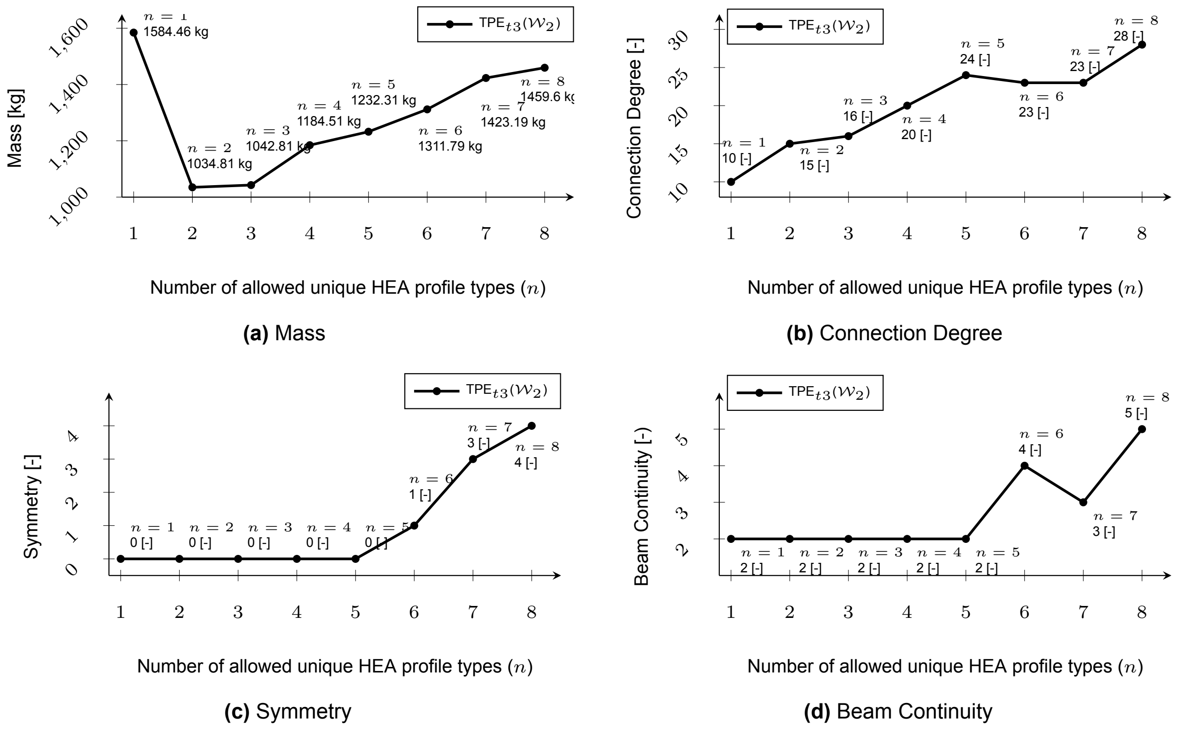
(a) Mass

(b) Connection Degree

(c) Symmetry

(d) Beam Continuity

Figure I.1: Results of TPE$_{t3}(\mathcal{W}_1)$ Optimization across Objectives for $N_{\text{trials}} = 2000$.



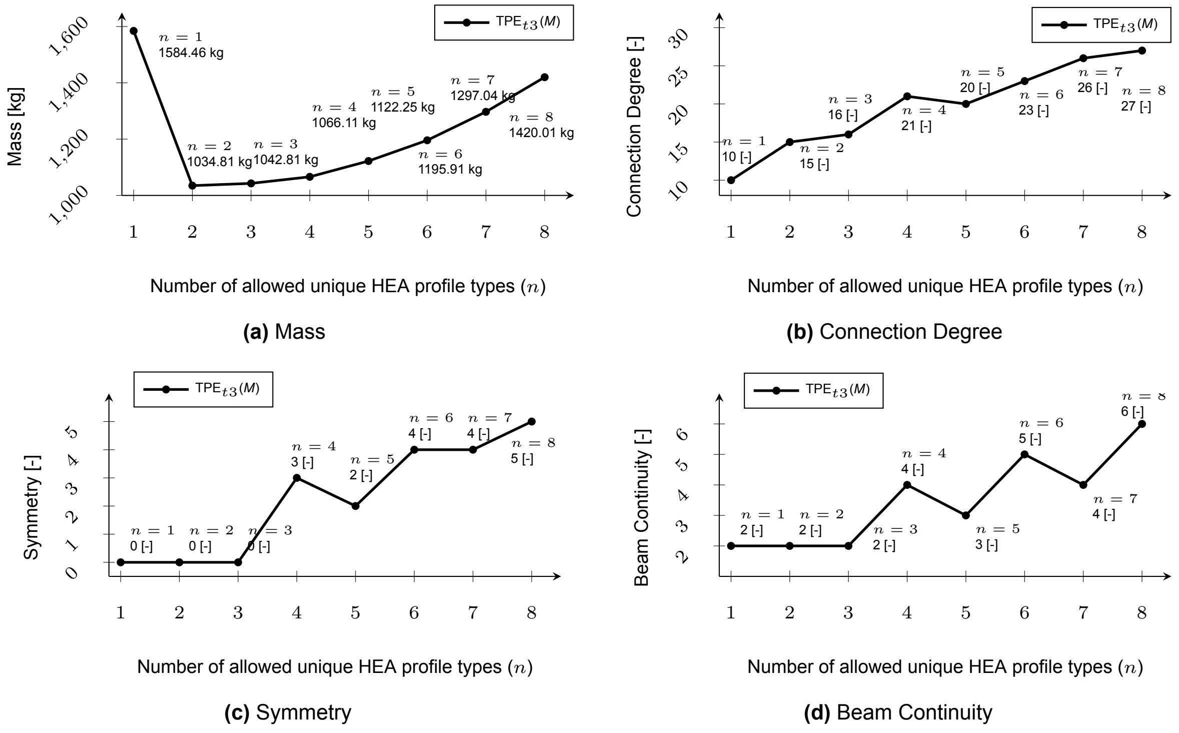
(a) Mass

(b) Connection Degree

(c) Symmetry

(d) Beam Continuity

Figure I.2: Results of TPE$_{t3}(\mathcal{W}_2)$ Optimization across Objectives for $N_{\text{trials}} = 2000$.



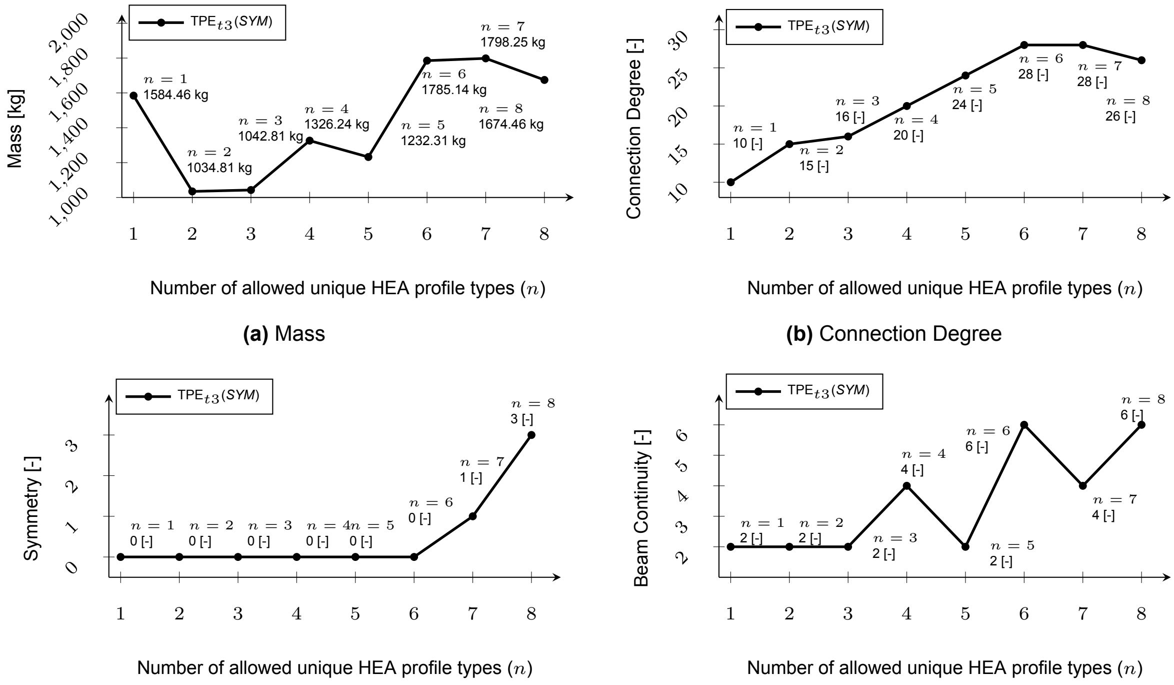
(a) Mass

(b) Connection Degree

(c) Symmetry

(d) Beam Continuity

Figure I.3: Results of TPE$_{t3}(M)$ Optimization across Objectives for $N_{\text{trials}} = 2000$.



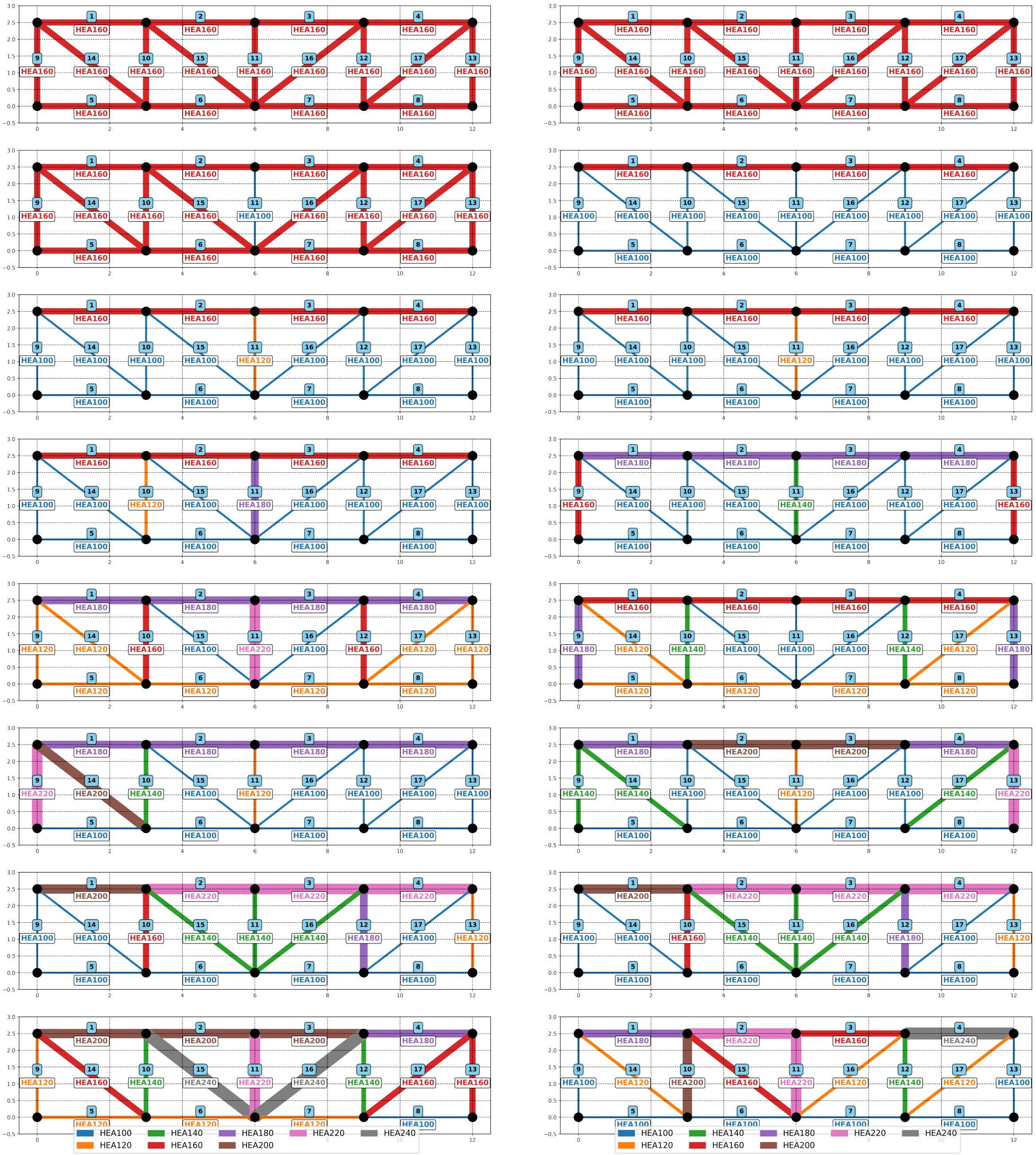
(a) Mass

(b) Connection Degree

(c) Symmetry

(d) Beam Continuity

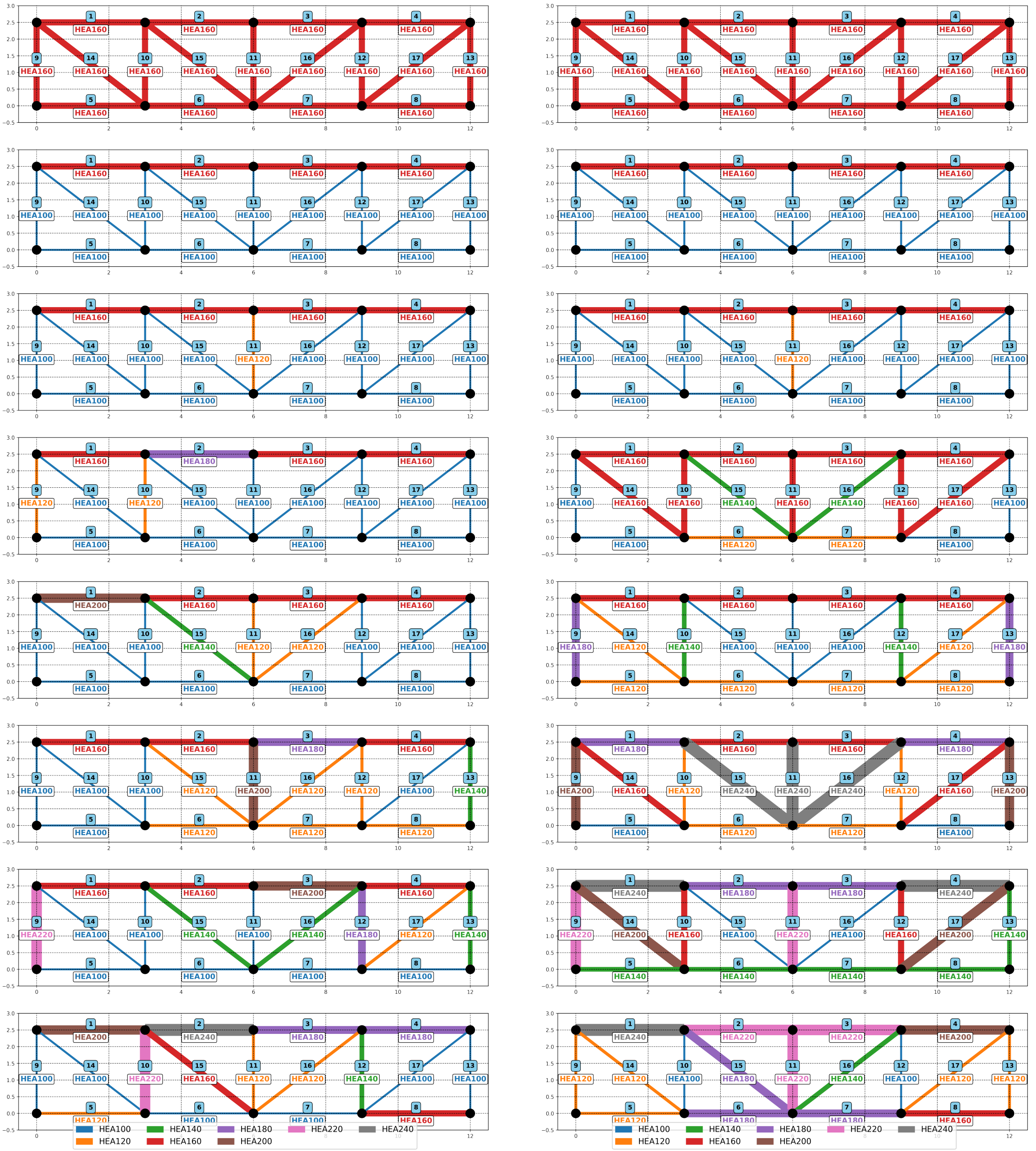
Figure I.4: Results of TPE$_{t3}$(*SYM*) Optimization across Objectives for $N_{\text{trials}} = 2000$.



(a) Graph of TPE$_{t3}(\mathcal{W}_1)$ optimization results for parametric truss with $n = 8$ profiles and $N_{\text{trials}} = 2000$.

(b) Graph of TPE$_{t3}(\mathcal{W}_2)$ optimization results for parametric truss with $n = 8$ profiles and $N_{\text{trials}} = 2000$.

Figure I.5: Graph of TPE$_{t3}(\mathcal{W}_1)(\mathcal{W}_2)$ optimization results for parametric truss with $n = 8$ profiles and $N_{\text{trials}} = 2000$.



(a) Graph of TPE$_{t3}$(*M*) optimization results for parametric truss with $n = 8$ profiles and $N_{\text{trials}} = 2000$.

(b) Graph of TPE$_{t3}$(*SYM*) optimization results for parametric truss with $n = 8$ profiles and $N_{\text{trials}} = 2000$.

Figure I.6: Graph of TPE$_{t2}$(*M*)(*SYM*) optimization results for parametric truss with $n = 8$ profiles and $N_{\text{trials}} = 2000$.