Research Article

IJAC

International Journal of
Architectural Computing
1–25

DOI: 10.1177/1478077120943062
journals.sagepub.com/home/jac
SAGE

Beyond typologies, beyond optimization: Exploring novel structural forms at the interface of human and machine intelligence

Karla Saldana Ochoa, Patrick Ole Ohlbrock, Pierluigi D'Acunto and Vahid Moosavi

Abstract

This article presents a computer-aided design framework for the generation of non-standard structural forms in static equilibrium that takes advantage of the interaction between human and machine intelligence. The design framework relies on the implementation of a series of operations (generation, clustering, evaluation, selection, and regeneration) that allow to create multiple design options and to navigate in the design space according to objective and subjective criteria defined by the human designer. Through the interaction between human and machine intelligence, the machine can learn the nonlinear correlation between the design inputs and the design outputs preferred by the human designer and generate new options by itself. In addition, the machine can provide insights into the structural performance of the generated structural forms. Within the proposed framework, three main algorithms are used: Combinatorial Equilibrium Modeling for generating of structural forms in static equilibrium as design options, Self-Organizing Map for clustering the generated design options, and Gradient-Boosted Trees for classifying the design options. These algorithms are combined with the ability of human designers to evaluate non-quantifiable aspects of the design. To test the proposed framework in a real-world design scenario, the design of a stadium roof is presented as a case study.

Keywords

Structural design, machine learning, topology, graphic statics, form-finding, Combinatorial Equilibrium Modeling, Self-Organizing Map, Gradient-Boosted Trees

Introduction

Structural design can be regarded as an independent discipline at the interface between architecture and structural engineering in which architectural and structural aspects are conceptually interwoven

Institute of Technology in Architecture, ETH Zürich, Switzerland

Corresponding author:
Karla Saldana Ochoa, Institute of Technology in Architecture, ETH Zürich, HIB E 15, Stefano-Franscini-Platz 1, CH-8093 Zürich, Switzerland.
Email: saldana@arch.ethz.ch

in a non-hierarchical way. The main aim of structural design is to negotiate between freedom of formal expression and static requirements by establishing a relationship between the form of a structure and the distribution of its internal forces. In the following paragraphs, the notion of typologies and the role of optimization procedures in structural design are briefly discussed since both aspects play a dominant role in many structural design processes.

Research background

Beyond typologies. In the early design phase, structural design concepts are commonly derived from typological reference-based considerations.[1,2] The definition of typologies implies a categorization of the unique relationship between the form of a structure and the related load-bearing behavior (e.g. suspension bridge, cable-stayed bridge). Examples for such a categorization can be found in the work of Engel.[3] However, as pointed out by the renowned structural engineers Ney et al.,[4] this typological approach in structural design has a "perverse effect," since "the vocabulary freezes the object, and the object thus frozen assumes a sort of inviolable legitimacy." According to Ney et al.,[4] "it is necessary to leave the typological approach behind in order to arrive at novel solutions and innovative structural forms." The necessity to get away from rigid typological considerations in the early conceptual design phase is also supported by the work of structural designers from the past. In this regard, one of the most relevant examples is given by the Italian engineer Musmeci[5] (1926–1981), who investigated on new methodologies for the generation of structural forms emphasizing the role of science as a creative tool for design. As described in detail in *La statica e le strutture*,[6] Musmeci's scientific work set the basis for the creation of the so-called "forms without a name."[7] It is worth mentioning that the recourse to a non-typological approach does not generally fall on a fertile ground within the structural design community.[1,2] On the contrary, there is still a strong tendency that enforces recipe-like structural design procedures and solutions to become the only modus operandi.[8] This trend is reflected in many curricula in engineering schools around the world.[9–11]

Beyond optimization

> (Structural) art is solving problems which cannot be formulated before they have been solved. The search goes on until a solution is found, which is deemed to be satisfactory. There are always many possible solutions, the search is for the best—but there is no best—just more or less good. Ove Arup[12]

Problems in structural engineering are often regarded as being well defined, thus having only one best solution. Such a solution is usually searched for through the application of optimization routines.[13] However, in line with the words of Ove Arup, a conceptual design problem can seldom be described in an explicit form, using, for example, clear objective functions. A design problem can be rather characterized as a wicked problem,[14] in which qualitative and quantitative aspects co-exist.[15] In this regard, optimization routines can be useful for solving only very specific sub-problems in the engineering domain (e.g. sizing of members, topology optimization), but they cannot act as the primary strategy in a structural design problem that by nature is nonlinear and is characterized by multiple possible solutions.[13,16–18]

Objectives and contributions

This article describes an approach to structural design that tries to go beyond the use of typologies and the sole application of optimization routines. This approach is enabled through the combination of innovative generative methods from the field of equilibrium modeling with state-of-the-art machine learning (ML) algorithms. The article represents the outcome of a research collaboration between the Chair of Digital Architectonics and the Chair of Structural Design at ETH Zürich, the first results of which were published by Fuhrimann et al.[19]

The main objectives of this article are:

1. To introduce a theoretical and operative structural design framework that allows generating multiple novel, diversified, and structurally informed design options that go beyond the conventional canon of structural typologies.
2. To combine the strengths of human cognition with those of machine-driven computation, in order to bring together the subjective evaluation and selection capacity of humans with the ability of machines to handle large datasets.
3. To introduce a machine-based procedure that allows generating new structural design options by learning the design preferences of human designers.

Along with the description of the technical peculiarities, the article presents a case study for the conceptual design of a stadium roof. This case study demonstrates the potential of the proposed framework, while possibly contributing to the ongoing discussion on the human-machine interaction within the field of structural design. While several of the techniques used in the proposed framework have been presented before (see section "Proposed structural design framework"), the combination and sequence of these operations can be regarded as the main contribution of this work. Thanks to its flexibility, it is possible to apply the framework to different contexts and various structural design problems.

The remainder of this article is organized as follows. Section "Proposed structural design framework" introduces the underlying methodological scheme for the proposed design framework. Besides, this section also describes and contextualizes the four fundamental operations that are at the base of the framework. In section "Design experiment," each operation and the related algorithms are specified and visualized within the context of a design case study. Section "Discussion" discusses the contributions of the presented work, points out its limitations and indicates possible future developments.

Proposed structural design framework

As highlighted in section "Research background," the use of existing typologies as the starting point of structural design often limits the range of possible design options to a rather small solution space. Moreover, the application of pure optimization approaches generally represents an over-simplification of the problem, since in a design task, a quantifiable optimum can hardly exist. Thanks to the current developments in the field of computer-aided design and ML, both aspects can be questioned and possibly overcome. Strong attempts have been made in recent years to shift from pure computerization toward computing,[20] and from prediction to formation. Hence, changing the "focus from modelling objects towards modelling processes, from designing shapes to designing behavior and from defining static digital constructs to the definition of computing systems that are capable of reciprocal data exchanges and feedback information."[13]

In this regard, a generic scheme of design operations, relationships, and properties, which was formalized by Oxman[21] in 2006, has been used here as a starting point for the development of the proposed structural design framework. This scheme allows describing structural design as an iterative exploration and search process. Hence, Oxman's scheme can be used to encapsulate different design approaches with specific cycles of design operations using different media and environments. In the context of structural design, these operations are typically generation, evaluation, clustering, selection, and regeneration.

Although being derived from Oxman's scheme, the design framework proposed in this research (Figure 1, left) presents some conceptual and technical differences. In fact, the sequence of operations applied here during a design session is intended to be a variable and user-dependent in itself. That is, the designer is not forced to follow a specific, pre-defined sequential order, but is allowed to define individually the sequence, the algorithmic settings, and the feedback mechanisms of the operations themselves (Figure 1, right). The

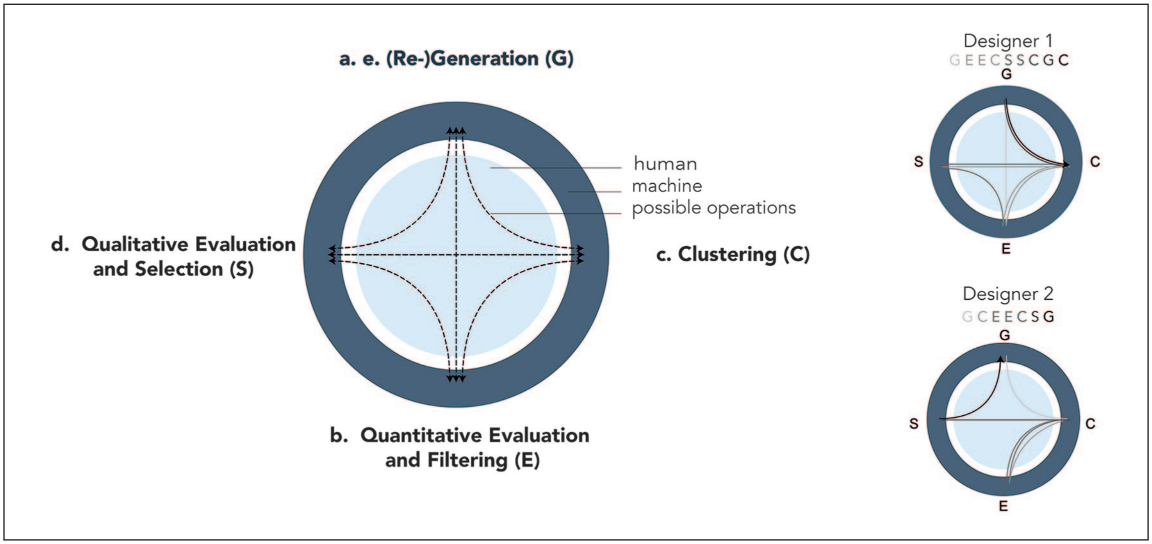

Figure 1. Diagram illustrating the proposed design framework (left) and personalized design procedures with different sequences and combinations of operations (right). The chronology of applied operations by each designer is described as a string (e.g. Designer 1 GEECSSCGC), where G stands for generation, E for quantitative evaluation and filtering, C for clustering, and S for qualitative evaluation and selection.

following subsections explain the specific role of each design operation within the overarching design framework and the related chosen algorithms.

Generation

Within the generative step, the notion of primary generators plays an important role.[22] Primary generators can be defined as a "broad initial objective or small set of objectives, self-imposed" by the designer, which reflect "a personal value judgement rather than the product of rationality."[22] Hence, primary generators have the purpose to reduce "the variety of potential solutions to a small class of solutions that is cognitively manageable."[22] Moreover, primary generators rely on multiple aspects of the design, such as problem framing, context, and model environment. They "form a starting point, a way into the problem,"[22] by defining the initial limits of the problem and suggesting the nature of possible solutions.[23] Primary generators not only have a strong influence on the process but also on the outcome itself. Moreover, it can be argued that in the conceptual design phase, a certain diversity within the set of possible solutions is generally beneficial in order to get a clearer picture of the "real nature" of the design task.[24]

A generative approach in structural design should be able to produce a multiplicity of very diverse, yet structurally sound options in a short time. The aim is to obtain a set of structurally informed forms that, on the one hand, go beyond the expectable typological references, and on the other hand, are intelligible and not disorganized in their appearance. Hence, methods in which structural principles such as static rigidity and equilibrium are explicitly formulated already in the computational generation of the model itself are of particular interest.

In the proposed framework, the Combinatorial Equilibrium Modeling (CEM)[25] is used as a primary generator and is the main algorithm within the generation step. The CEM is grounded on vector-based three-dimensional (3D) graphic statics,[26] and the equilibrium of a structure is constructed directly, based on a specific sequential decomposition of the equilibrium problem. The CEM is available as an open-source toolkit (https://github.com/OleOhlbrock/CEM, accessed April 2020), developed using the scripting

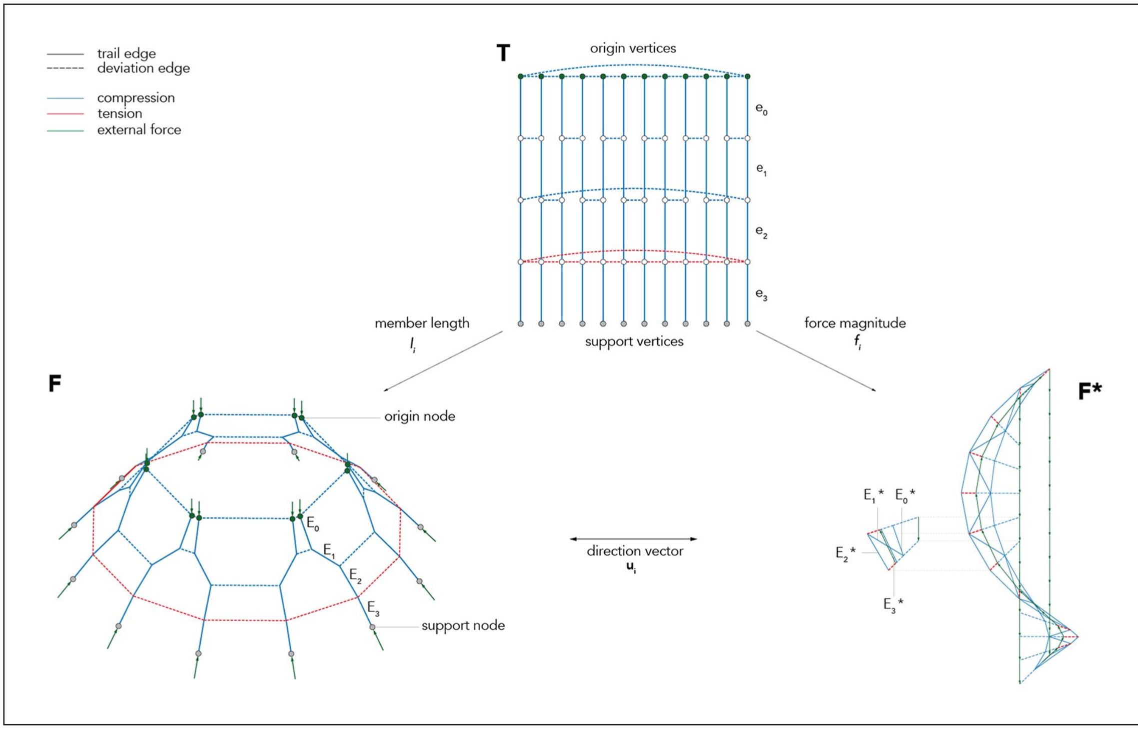

Figure 2. Relationships between the topological **T** (top), form **F** (bottom left) and force **F*** (bottom right) diagrams within the CEM framework. Tension is represented in red, compression in blue, and external forces in dark green.

language IronPython (https://ironpython.net/, accessed April 2020) as a plug-in of the commercial CAD software platform McNeel Rhinoceros (https://www.rhino3d.com/, accessed April 2020) and Grasshopper (https://www.grasshopper3d.com/, accessed April 2020).

For a given topological diagram **T** (Figure 2, top), which represents the topology of the structure, the CEM algorithm generates the form diagram **F** as pin-jointed frameworks in equilibrium (Figure 2, bottom left), and the related force diagram **F*** (Figure 2, bottom right). The CEM algorithm is initialized by introducing a series of topological and metric parameters. The topological parameters, which can be defined directly in **T**, are the connectivity of the structural elements and the combinatorial state of their internal forces (i.e. tension, compression, or null). The metric parameters are the trail lengths—that is the lengths of the members of **F** that correspond to the trail edges of **T**—and the deviation force magnitudes—that is the magnitudes of the internal forces of the members of **F** that correspond to the deviation edges of **T**. By enforcing the equilibrium of the nodes of the structure in **F**, the CEM algorithm seeks the equilibrium form associated with the given **T**, based on the input topological and metric parameters.[25] The equilibrium is imposed sequentially starting from the origin nodes of **F**, which correspond to the origin vertices of **T** up to the support nodes of **F**, which correspond to the support vertices of **T**. The advantage of the CEM over analysis-oriented approaches like the Finite Element Method (FEM) is that the CEM allows for a synthetic and intuitive representation of static equilibrium. Thus, the CEM can be easily used for the fast generation and transformation of equilibrium systems, in which the inner constellation of compression and tension forces is used as a key design generator. Moreover, in comparison with other form-finding methods such as the Force Density Method (FDM),[27] the CEM has possibly the advantage that the metric parameters can be assigned and varied directly in absolute physical units like lengths and forces.

Different methods exist to generate a multiplicity of topologically unpredictable, yet structurally valid design options beyond or in-between typological borders.[28,29] The contribution of Shea et al.[30] is one of the first examples for the use of generative methods in structural engineering. The works of Van Buelow[31] and Brown[2] are representative of recent research on interactive evolutionary optimization using parametric modeling. The investigations of Mueller[1] and Lee et al.,[32] on the contrary, exemplify the use of grammar-based generative methods in structural design.

The CEM focuses on the controlled diversity at the topological level while giving relevance to the combinatorial state of the internal forces. Hence, through simple variations of compression and tension forces in the generative step, a vast formal diversity can be obtained from a common ground topology.[19] Such diversity within an explicitly controllable topological set-up potentially helps to find a balance between the unpredictability of structurally driven bottom-up approaches and the challenges connected to the use of explicit top-down form-finding strategies.[33] As a result, the use of the CEM possibly allows the design process to produce structurally informed design options that go beyond a fixed canon of existing typologies while remaining explicitly controllable at the topological level.

Quantitative evaluation and filtering

With the use of fast generative methods, the designer has access to several thousands of design options in a short amount of time. In order to guide the design process, a generative engine is usually coupled to an optimization routine, which searches for optimal solutions for a given set of hard or/and soft constraints, expressed through the definition of objective functions. However, while optimization routines are useful in the final converging stages of the design process, in the conceptual phase, designers cannot rely solely on their application. In fact, as previously mentioned (see section "Research background"), the design problems are ill-defined, and they generally have more than one possible solution. The recourse to quantitative structural performance measures as the objective functions is not necessarily satisfying in real-world design scenarios, as they can drastically reduce the diversity of the possible design options too early. Therefore, rather than trying to find an optimum solution with an arbitrary objective function, it would be more appropriate to keep the acceptable solutions and to filter out the unacceptable ones.

Within the context of structural design, one can find several well-known structural quantitative measures, which are valuable for evaluating the performance of a structure. At the level of global structural behavior, particularly relevant are the global stability and the degree of static indeterminacy of a structure, which can be assessed based on the equilibrium matrix of the structure,[34] as well as the self-weight, and the load path (LP)[35]. At the level of the local structural behavior, of interest are the maximum and minimum internal forces, and the local stability of the structural members, which can be evaluated through the slenderness ratio (SR) of the structural members (see section "Quantitative evaluation and filtering based on structural performance").

Unlike the typical optimization procedures of structural engineering, in the quantitative evaluation and filtering step of the proposed design framework, these quantitative filters are used as hard boundaries. Diverse quantitative filters can be applied to a given set of design options sequentially. That is, rather than pointing to the best results, these filters are used to delete the unacceptable design options from the space of possible design solutions generated in the previous step. For example, the LP of a structure measures how the load is transferred through the structure.[35] This quantity can be used as an objective function to minimize the volume of a compression-only structure.[36] However, minimizing the LP of a structure as a design strategy does not necessarily help the designer to find innovative structural forms, especially at the early stages of design. This point is clearly illustrated in Figure 3, in which structural forms are created using the CEM, starting from the same ground topology. Although the structures possess a similar LP, they present a wide range of formal variation.

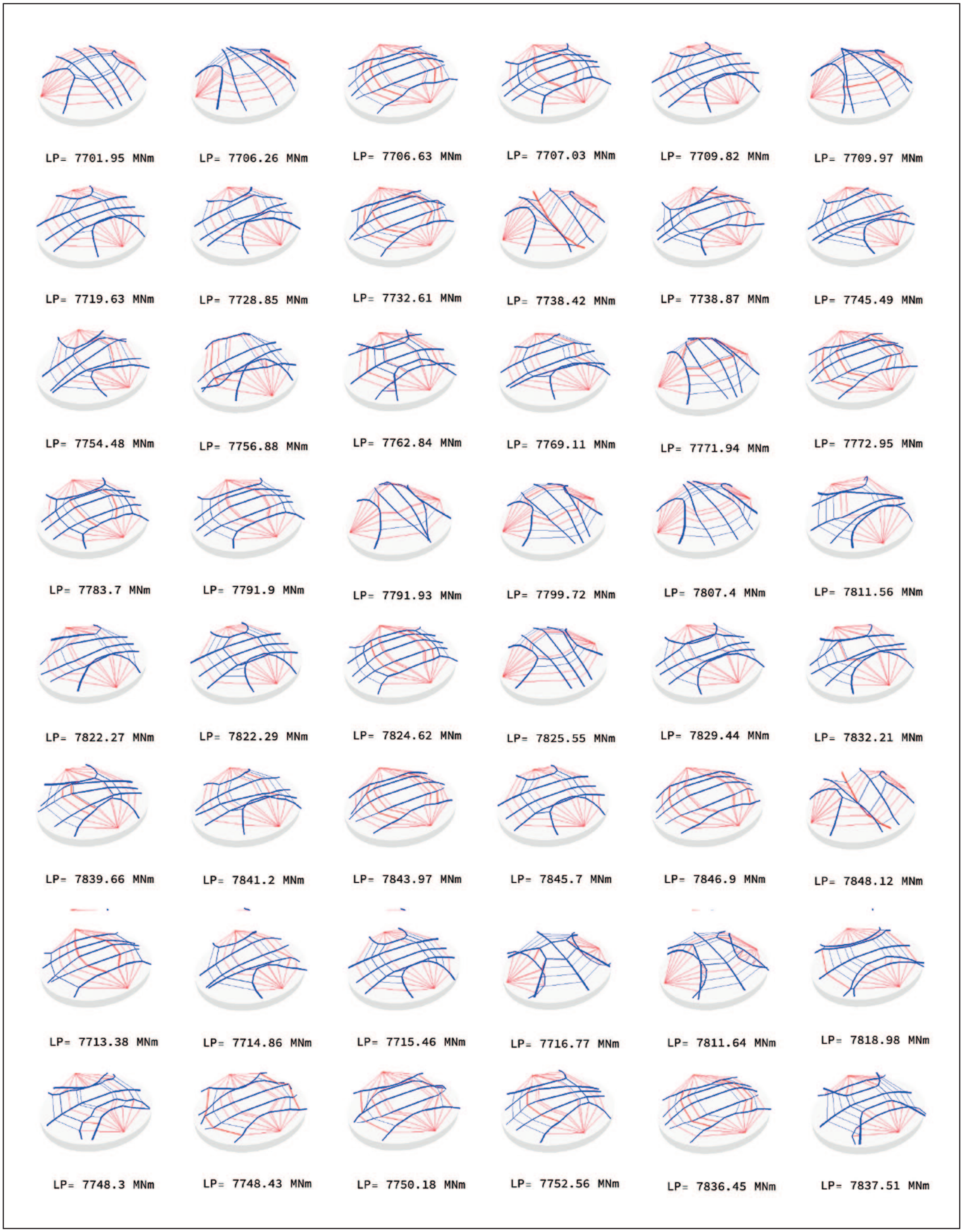

Figure 3. Randomly chosen design options for a roof structure with similar load paths (LP) do not necessarily have similar forms (structural members with tensile forces are in red and those with compressive forces are in blue).

Clustering

The scope of clustering operations is to group dynamically various design options into sets with similar design features. While in a classical design process based on standard structural typologies, the variation of the design parameters leads to slight variations within a given family of forms, by clustering thousands of design options, one can expect to obtain new emergent structural typologies. As shown by Fuhrimann et al.,[19] different clusters capture different families of forms, which were generated from the parametric variation of the design inputs. Such data-driven catalogs can give the designer valuable insights and new means to further explore the design space.

In the proposed framework, the machine provides an intuitive and interactive environment for the clustering of the design options, while the human designer actively decides how to navigate in the clustered design space. Within the literature of ML, one can find different clustering algorithms. For a given vector-based representation of the dataset and a choice of a distance (e.g. Euclidean, Cosine, any other L^P-norms), these algorithms assign a data point to each design option, while minimizing the similarity of data points within each cluster and maximizing the dissimilarity between different clusters. Although a typical clustering algorithm such as K-means[37] can be seen as a data-reduction method, other algorithms, which are developed around the concepts of dimensionality reduction and manifold learning, can also be used for the purpose of clustering. Among these algorithms, T-SNE[38] or the recently developed UMAP[39] are particularly relevant. While in algorithms like K-means, the data in each cluster is reduced to only K representative points, in other dimensionality reduction methods, the user can visualize a spectrum of all generated design options in relation to each other, giving a better overview of the overall design space.[19]

In the proposed framework, the algorithm of choice for clustering is Self-Organizing Map (SOM), which takes advantage of both clustering and dimensionality reduction.[40] SOM acts as a nonlinear data transformation in which data from a high-dimensional space is transformed into a low-dimensional space (usually a space of two or three dimensions), while preserving the topology of the original high-dimensional space. Topology preservation means that if two data points are similar in the high-dimensional space, they are necessarily close in the new low-dimensional space, and, hence, they are placed within the same cluster. This low-dimensional space, which is usually represented by a planar grid with a fixed number of points, is called a map. Each node of this map has specific coordinates ($x_{i,1}$, $x_{i,2}$) and an associated n-dimensional vector or Best Matching Unit (BMU), in such a way that similar data points in the high-dimensional space are given similar coordinates. Moreover, each node of the map represents the average of the n-dimensional original observed data that after iteration belong to this node. In various occasions, SOM has been employed in the context of architectural design,[41,42] and it can be conveniently used to visualize high-dimensional spaces.[43,44] The implementation of the SOM algorithm used in the present work, and all the other ML algorithms described in this and the following sections, have been coded using built-in functions within the software Wolfram Mathematica (https://www.wolfram.com/mathematica/, accessed April 2020). The entire code is an open source (https://github.com/sakarla/Beyond-typologies-beyond-optimization/, accessed April 2020).

Beside the choice of the clustering algorithm, in the context of structural design, the main challenge is how to compare and measure the similarity between two distinct design options, which, in turn, depends on how these design options are represented. Table 1 shows various design features related to the input and output design parameters. For each combination of design features selected by the human designer, a different representation of the generated design options evolves, which possibly leads to a different clustering pattern by the machine.

For quantitative features, such as the LP (see section "Quantitative evaluation and filtering"), a single scalar value is assigned to each design option. For features such as the distribution of the internal forces in the structure, it is necessary to describe the related data distributions for each design option statistically. Higher Order Statistics (HOS)[45] is used in the proposed design framework to capture the invariances of the

Table 1. Possible ways to represent and compare the design options based on various design features.

Feature	Vector representation	Description
Trail lengths (input)	$\boldsymbol{t} = [t_1, t_2, \ldots, t_T]$	where $\boldsymbol{t}$ is the vector containing the lengths of the trail members t_i, with T the number of trail members
Deviation force magnitudes (input)	$\boldsymbol{d} = [d_1, d_2, \ldots, d_D]$	where $\boldsymbol{d}$ is the vector containing the force magnitudes of the deviation members d_i, with D the number of deviation members
Position of origin nodes (input)	$\mathbf{O} = [(O_1x, O_1y, O_1z), \ldots, (O_nx, O_ny, O_nz)]$	where $\mathbf{O}$ is the vector containing the position of the origin nodes O_i in x, y, z coordinates, with n the number of origin nodes
Position of the nodes (output)	$\boldsymbol{P} = [(P_1x, P_1y, P_1z), \ldots, (P_mx, P_my, P_mz)]$	where $\boldsymbol{P}$ is the vector containing the position of the nodes P_i in x, y, z coordinates, with m the number of nodes
Members' lengths (output)	$\boldsymbol{l} = [l_1, l_2, \ldots, l_L]$	where $\boldsymbol{l}$ is the vector containing the lengths of the structural members l_i, with L the number of structural members
Internal forces (output)	$\boldsymbol{f} = [f_1, f_2, \ldots, f_F]$	where $\boldsymbol{f}$ is the vector containing the force magnitudes of the structural members f_i, with F the number of structural members
Any combination of structural quantitative measures (see section "Quantitative evaluation and filtering")		the vector representation depends on the chosen quantitative measures

The human designer dynamically chooses any combination of these features to describe the design options, and the clustering algorithm automatically identifies the patterns of emergent typologies.

features of the design options on a higher level. In HOS, for a sample of n observations, one could calculate up to n moments of data distribution. For example, the commonly used arithmetic mean and standard deviation are defined, respectively, as the first- and the second-order moments of the data distribution. Skewness is the third-order moment of the data distribution, which measures the direction of the tail of the distribution (at the right or left of the mean) in comparison with the normal distribution. If the resulting number is positive, the data is skewed to the left, leaving the tail pointing to the right side of the distribution. If the resulting number is negative, the tail is on the left side of the distribution. Kurtosis is the fourth-order moment of the data distribution, which measures how heavy the tails of a distribution are, or correspondingly, measures the pickiness (the shape of the curve, and specifically, the bell of the curve).[46] These measures have geometric interpretations and are commonly used as shape parameters, which capture information from the dataset.[47] To elaborate more on this point, we give an example of the different interpretations of the LP of a structure concerning the HOS measures in Appendix 1.

Qualitative evaluation and selection

In the proposed framework, the inclusion of the preferences by the human designer adds qualitative constraints on top of quantitative ones. This approach enables going beyond solely objective functions by considering those subjective formal qualities perceived by the human designer. Once the solution space is generated, evaluated, and clustered, the human designer can select those clusters or single design options that have the overall best performance concerning quantitative (measurable) and qualitative (non-measurable) aspects.

Techniques that represent high-dimensional data in a comprehensible manner ideally allow a designer to get a concise overview of the essence of all the possible options without suffering from evaluation fatigue.[48] Thus, it can be argued that through the recent developments in the field of ML, human designers can interact with an extensive set of possible options at a time. Examples of such a human–machine interaction can be found in the tool Biomorpher, developed by Harding and Brandt-Olsen,[29] in the work of Menges,[13] and the early work of Coates et al.[49] In the proposed design framework, the designer selects families of design options directly from the generated SOM through an interactive user interface. The time spent to select the preferred and non-preferred options (qualitative evaluation) is directly related to the intuitiveness of the user interface, the aim of which is to facilitate the interaction of the designer with the organized clustered of options. As this matter would require an extensive discussion, the design of an appropriate user interface is not addressed in the present work, but it will be developed in future work.

Regeneration

In case the selected design options are not yet satisfactory, the generation, representation, evaluation, and selection operations have to either be repeated or reformulated. A common procedure for iterative regeneration is based on evolutionary strategies. In these approaches, the parameters (genotypes) of the selected candidates (phenotypes) generate new candidates, which potentially combine the strengths of the parents, and hence outperform the candidates of the previous generation. A full review on the use of evolutionary computation in the field of structural design can be found in Kicinger et al.[50] In the case of Biomorpher,[29] for example, after the clustering operation, the human designer can select a cluster of options to continue the evolutionary process.

Another approach, which is employed in the proposed design framework, is to use the selected design options as labeled training data to be fed into supervised ML algorithms for classification. Commonly ML classifiers, such as Decision Tree,[51] Random Forest,[52] and Gradient-Boosted Trees (GBT),[53] aim to map from unlabeled instances to classes. A full review of classification techniques can be found in Kotsiantis et al.[54] In the present research, the selection made by the designer creates label instances to train the ML classifier to classify unlabeled data. Hence, the classifier is trained to learn the nonlinear relations between the input parameters of the generator and the preferences indicated by the feedback of the human designer. Subsequently, the outputs automatically produced by the ML classifier are used as new inputs for the generative routines process. In the proposed design framework, the choice of the classifier depends on the dataset under investigation since the performance of the classifier is strongly related to the characteristics of the dataset.

Design experiment

In this section, a representative design experiment is used to illustrate the potentials of the proposed structural design framework, and to describe the related operations and the chosen algorithms in more detail. The experiment is placed within a typical structural design task, which consists in the creation of a new roof for the undulating bowl of the Nya Ullevi Stadium in Gothenburg, Sweden. The stadium was initially built for the 1958 FIFA World Cup and has a seating capacity of 43,000 people. In the design experiment, only those design options whose supports match the existing undulating bowl were accepted in order to preserve the structural and formal characteristics of the stadium. Moreover, the accepted design options were the ones with a slenderness ratio (SR) defined by a quantile of 50% and a load path (LP) defined by a quantile of 80% in order to take into account the structural performance.

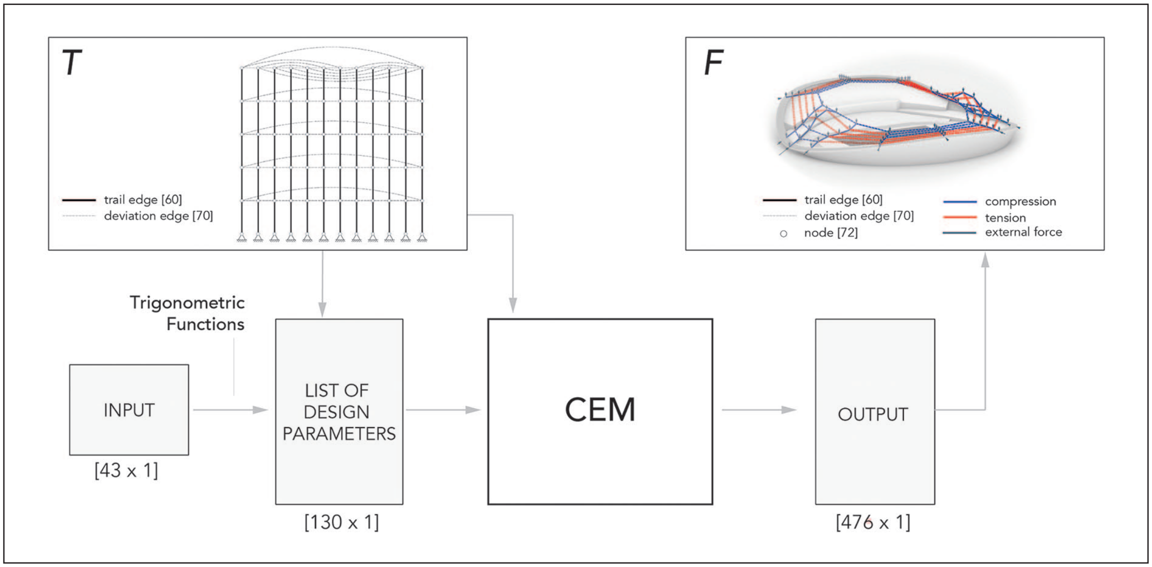

Figure 4. Trigonometric functions map the 43 randomized input values (left) to the list of 130 design parameters (center). Together with the topological definition, the design parameters are fed into the CEM algorithm. The latter returns a form in equilibrium that can be described by 476 values (right).

Machine-driven generation with CEM

Figure 4 (top left) shows the topological diagram used to initialize the generative process with the CEM. It consists of 72 vertices and 130 edges that are organized in 6 sequences. Within the CEM formulation, edges are subdivided into two distinct categories (see section "Generation"). On the one hand, trail edges (continuous line in Figure 4, top left), here 60 grouped in 6 sequences, defining 12 trails that connect each vertex with a direct load transfer (topologically) to the closest support. On the other hand, deviation edges (dashed lines in Figure 4, top left), here 70, that connect vertices on different trails. The metric values that relate to the trail edges are the trail lengths, while for the deviation edges are the deviation force magnitudes. In order to generate various structural forms as design options an automatic random variation of the 130 trail and deviation parameters was adopted. For each sequence, a trigonometric function[19] defined by three parameters (amplitude, frequency, and phase) was employed to control the 12 independent trail lengths and the 12 independent force magnitudes of the sequence at once. As a result, the number of 130 independent metric parameters could be reduced to 30; 13 additional parameters were used to describe special geometric features in the first and last sequence of the topological diagram and to describe the initial layout of the origin nodes in the form diagram. Hence, the random variation was carried out on a higher abstract level that ensured certain regularities among the individual values and thus prevented complete noise in the design space.[55,56]

Figure 4 highlights the relationship between the $30 + 13 = 43$ input parameters and the resulting 130 values for the trail and deviation edges that were taken as input for the generation. The output of the CEM (Figure 4, bottom, right) is a three-dimensional tensor containing 476 values: the position vector of the nodes (72×3), the length of the trail members (130, out of which 60 are equivalent to the input values) and the force magnitudes of the deviation members (130, out of which 70 are equivalent to the input values). Out of this information, a form (Figure 4, top, right) and a force diagram can be constructed unequivocally.

Given the design brief, a gradient-based optimization routine was run to fulfill given geometric boundary constraints.[25,57] Hence, the objective was to minimize the summed distance r_i (least-squares) between each

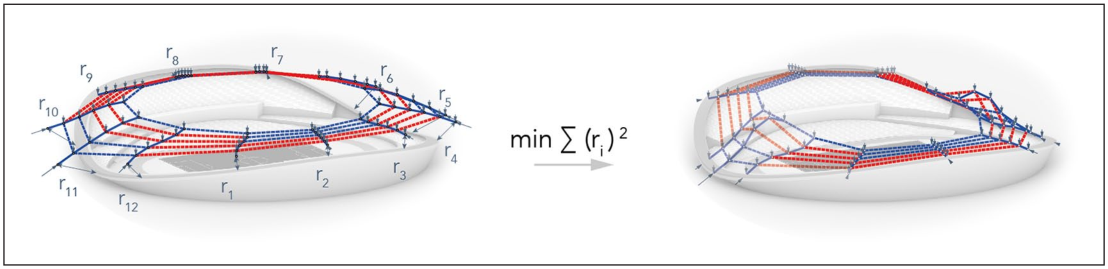

Figure 5. Initial form diagram with 12 residual distances r_i (left) and after the minimization of these distances to match the location of the support nodes (right).

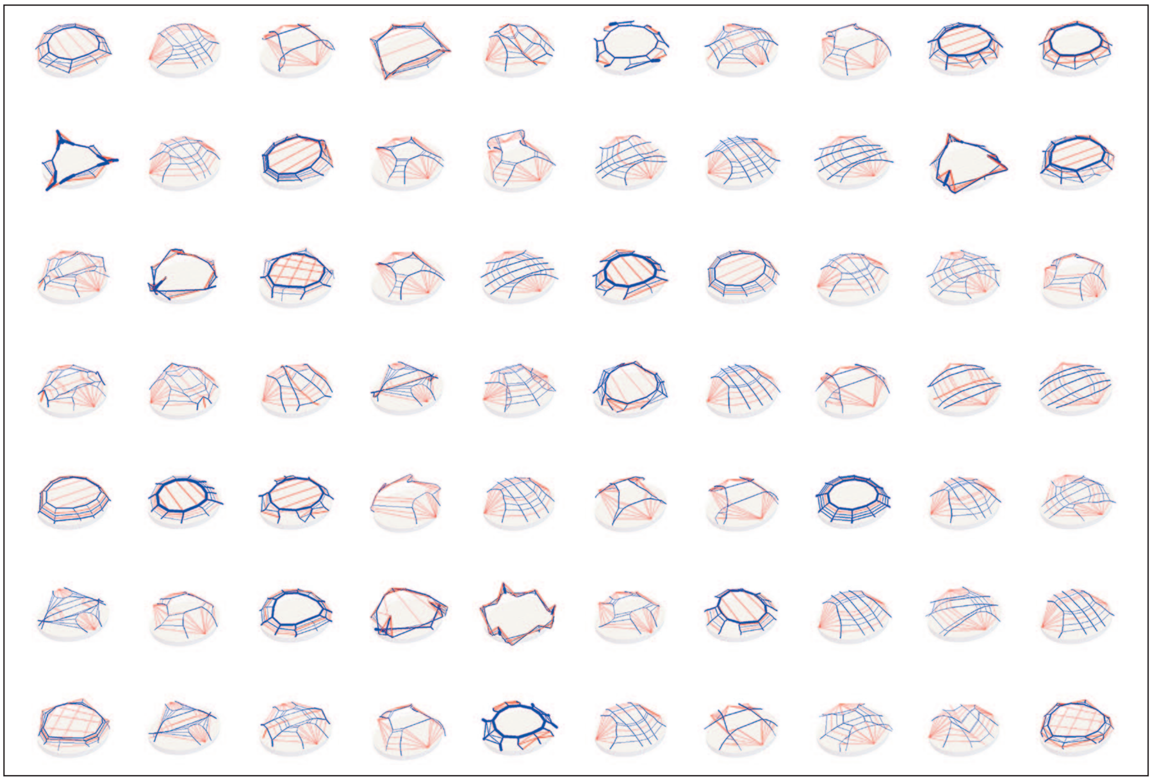

Figure 6. Random sample of 70 generated design options.

support node and a target strip at the border of the undulating bowl only through a variation of the position vectors (3 components each) of the 12 origin nodes. The gradients that contain $12 \times 3 = 36$ dimensions were obtained through finite difference approximation. Figure 5 (left) shows an initial form diagram before optimization, while Figure 5 (right) illustrates the form diagram after optimization.

For the automatic generation of the dataset of forms, only those design options that matched the target geometry with a predefined maximum number of iterations and within a given threshold were accepted and recorded. The given topological setup allowed generating a vast amount of radically different, unexpected, yet structurally informed options (Figure 6). In terms of computation time, each design option was calculated in 500 ms with a Quad-Core 2.8 GHz CPU. With parallel computing, 35,205 design options could be produced in less than 2 hours using a 10-Core 2.5 GHz CPU. These solutions were stored in a CSV file with a capacity of 2 GB.

Quantitative evaluation and filtering based on structural performance

In this step, different quantitative filters can be used (see section "Quantitative evaluation and filtering") to reduce the amount of design options. In this design experiment, the local stability was taken into consideration, knowing that members in compression generally require a larger cross-section than the ones in tension. Restricting the slenderness ratio (SR) of the members in compression regarded as elements with round hollow cross-section can be used to reduce the danger of buckling:

$$SR_i = \frac{l_i}{\sqrt{\frac{\pi\left(R_i^4 - \left(R_i - W_i\right)^4\right)}{4A_i}}} \tag{1}$$

where l_i is the length of each structural member, R_i is the outer radius of the round cross-section, w_i is the wall thickness, and A_i is the cross-section area. Within the design experiment, a quantitative filter was formalized selecting the best options with respect to SR (50% quantile). After applying this filter, the number of design options was reduced from 35,205 to 17,602.

The second quantitative filter is formulated in relation to the load path (LP), which is the sum of the products of the length l_i of each structural member and the absolute value of the axial force f_i acting on the member[35]

$$LP = \boldsymbol{f}^T \cdot \boldsymbol{l} = \sum_{i=1}^{n} |f_i| l_i \tag{2}$$

where $\boldsymbol{f} = [f_1, f_2, \ldots, f_n]$ is the vector containing the force magnitudes and $\boldsymbol{l} = [l_1, l_2, \ldots, l_n]$ is the vector containing the lengths of the structural members in a structure with $\boldsymbol{n}$ structural members.

In the design experiment, LP was used as a second filter (80% quantile). This further reduced the number of design options from 17,602 to 14,082.

Clustering with SOM

The 14,082 design options remaining from the previous step were transformed into numerical inputs using the four moments of HOS function of the members' lengths (mean, variance, skewness, and kurtosis; see section "Clustering"). This dataset of numerical inputs was fed as input into an SOM with a grid size of 20×20. The appropriate size of SOM's grid depends on the design task, the resolution of the model, and the intention of the designer.[33] It has been observed by the authors that the appropriate grid size usually does not exceed a dimension of 40×40 in order to keep a clear overview of the clustered design options. However, more research has to be done to generalize this observation.

As explained in section "Clustering," SOM can recognize similar patterns among the data and organize the corresponding options accordingly. The resulting SOM spans over 400 cells or BMUs (see section "Clustering"). In Figure 7 (top), the design options displayed are the ones with the closest Euclidean distance to its corresponding BMU value. Therefore, they can be considered as representative cases for each cell. The trained SOM can additionally visualize the four moments of HOS (Figure 7, bottom),

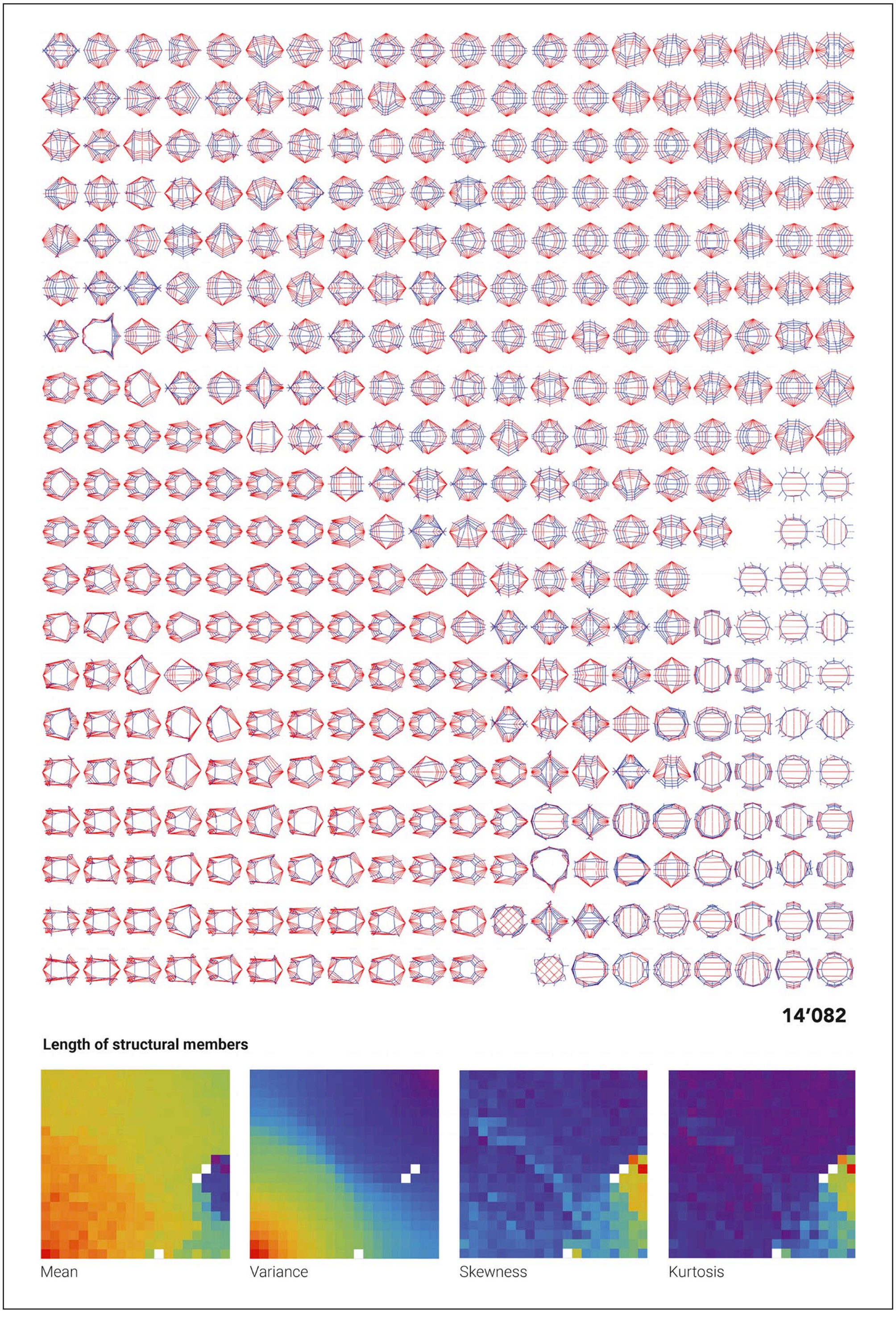

Figure 7. An SOM of 14,082 filtered design options based on HOS of length of structural members (top). The BMUs of the SOM are colored based on each HOS measure: mean, variance, skewness, kurtosis (bottom).

corresponding to each BMU. The visualization of the four moments of HOS provides a detailed exploration of the design space as it displays specific clusters of design forms highlighted for each moment of the HOS. When analyzing the SOM grid with design options, one can recognize at least two different families of design options emerging that belong to specific settings in the generative step. In addition, it can be observed that the first and last cells contain a larger quantity of design options than all the other cells (e.g. cell 1 = 116 and cell 400 = 271). The central cells tend to contain less design options (e.g. cell 190 = 1 and cell 152 = 2). The cells that do not contain information are considered as boundaries; this implies that no design options can be placed in-between.

User-based qualitative evaluation

Using the two-dimensional (2D) grid SOM defined in the previous step, the designer can not only get a fast and precise overview of possible design options but also distinguish between preferred and non-preferred options. As explained before, each cell of the SOM represents a set of multiple design options; therefore, by choosing any of them, its associated design options are selected as well. The design options are clustered in each cell of the SOM based on the design features selected by the designer. For example, if the design features used for clustering are related to the form and the grid size is large enough, the design options in each cell will be visually similar.

Figure 8 shows a screenshot of a selection procedure. The developed interface enabled the designer to select clusters of design options directly on the SOM. The interface visualizes a sample of structural forms contained in the selected node and indicates in real-time how many forms are present within each cell. This approach supports an interactive and informed selection. Within this illustrative design experiment, the designer selected 2,770 preferred and 2,574 non-preferred options, resulting in a dataset of 5,344 design options. Figure 8 depicts a sample of the selected options that belong to each list.

Regeneration with GBT and CEM

As explained in section “Regeneration”, in this operation, an ML classifier is trained to learn the nonlinear relation between the input parameters of the generator and the preferences indicated by the feedback of the human designer. The corresponding parameters that defined the input values to generate the structurally informed options with CEM (see section “Machine-driven generation with CEM”) were turned into input variables. The preferences of the designer (i.e. preferred = 1 and non-preferred = 0) were assigned labels, thus creating a binary numerical labeled dataset combining the designer preferences with the inputs used to generate the design options (Figure 9).

The dataset was then split into training and test data in a ratio of around 90:10, respectively (training data = 4,788 and test data = 556). The training data was fed as input to each ML classifier: Decision Trees (with a distribution smoothing of 1 and feature fraction of 1), Random Forest (with a distribution smoothing of 0.5, a feature fraction of 0.1524, leaves’ size of 5, and with 50 trees), and GBT (with maximum training rounds of 50, 500 leaves, leaves’ size of 35, and a maximum depth of 6). The performance of the obtained trained classifiers was then validated with the test data (556). Figure 10 depicts the confusion matrix for the three tested ML classifiers, while Table 2 shows detailed information of each classifier. A confusion matrix is a specific table layout that allows visualization of the performance of an algorithm. Each row of the matrix represents the instances in a predicted class, while each column represents the instances in an actual class (or vice versa). The use of a confusion matrix is beneficial for measuring accuracy (i.e. the percentage of correct prediction divided by the total number of predictions) and precision (i.e. how consistent the results are when measurements are repeated).

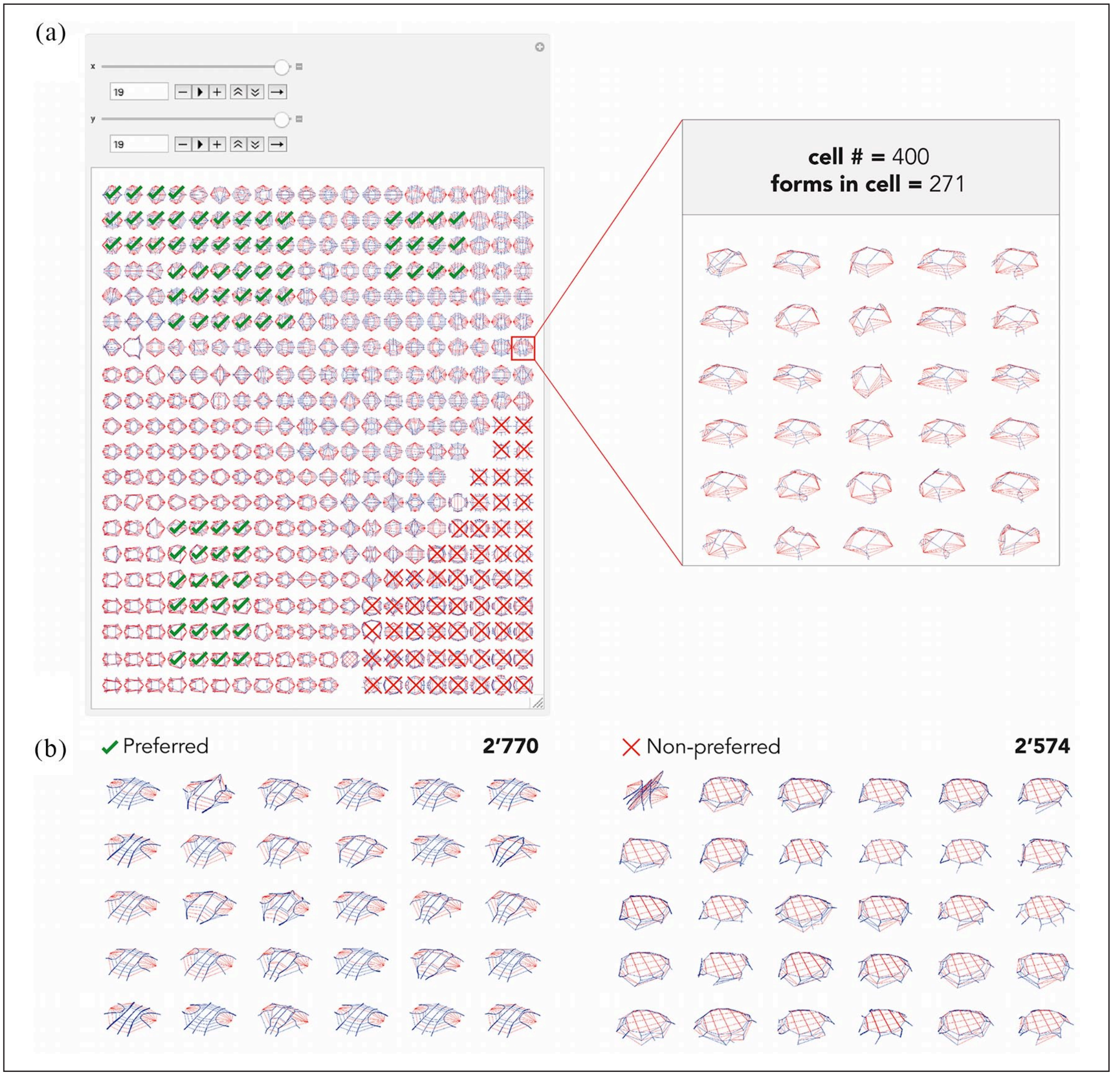

Figure 8. Screenshot of the interface with the 2D grid SOM and the corresponding selection list: (a) list of design options classified as preferred 2,770 and non-preferred 2,574 (b) out of the selected 5,344 options.

Compared to the other two classifiers, GBT was found to have the best performance with the given dataset, having an accuracy of 92% and precision for each class as follows: Class 0 $\rightarrow$0.9140625 and Class 1 $\rightarrow$0.926666 (Table 2). Commonly, GBT is used for regression and classification problems producing a prediction model in the form of an ensemble of trees. Trees or weak learners are trained sequentially to improve the accuracy and robustness of the final model. In the proposed model, LightGBM approach was employed.[58] This approach has the following advantages: a faster training speed and higher efficiency, lower memory usage, better accuracy, support of parallel learning, and capability to handle large-scale datasets.

{30.,2.,90.,0.,2.,0.,-6.,-12.,2.,0.,-1.1,-11.,2.,0.,-1.1,2.,0.,-1.1,2.,0.,-1.1,0.4,0.,0.,-5.,2.,0.,0.3,2.,0.,-2.,2.,0.,0.4,2.,0.,-0.6,1.,3.,40.,40.,500,0.} → 1
{30.,2.,90.,0.,1.,0.,-1.2,-4.,1.,0.,-4.8,-12.,1.,0.,-4.8,1.,0.,-4.8,1.,0.,-4.8,0.4,0.,0.,-6.,1.,0.,2.3,1.,0.,-1.4,1.,0.,-0.3,1.,0.,4.4,0.,3.,40.,40.,500.,0.} → 0
{30.,2.,90.,0.,0.,0.,-2.4,-8.,0.,0.,-0.8,-8.,0.,0.,-0.8,0.,0.,-0.8,0.,0.,-0.8,0.4,0.,0.,-4.,0.,0.,2.4,0.,0.,-2.9,0.,0.,-2.8,0.,0.,4.6,0.,3.,40.,40.,500.,0.} → 1
{30.,2.,45.,0.,4.,0.,-4.4,-11.,4.,0.,-1.6,-4.,4.,0.,-1.6,4.,0.,-1.6,4.,0.,-1.6,0.4,0.,0.,-6.,4.,0.,0.5,4.,0.,0.3,4.,0.,-0.6,4.,0.,0.2,3.,3.,40.,40.,500.,0.} → 0
{30.,2.,90.,0.,1.,0.,-3.3,-11.,1.,0.,-2.4,-8.,1.,0.,-2.4,1.,0.,-2.4,1.,0.,-2.4,0.4,0.,0.,-7.,1.,0.,0.6,1.,0.,-1.8,1.,0.,-2.1,1.,0.,4.,2.,3.,40.,40.,500.,0.} → 0
{30.,2.,90.,0.,1.,0.,-3.6,-12.,1.,0.,-0.4,-4.,1.,0.,-0.4,1.,0.,-0.4,1.,0.,-0.4,0.4,0.,0.,-6.,1.,0.,2.3,1.,0.,0.2,1.,0.,2.2,1.,0.,-3.9,2.,3.,40.,40.,500.,0.} → 0
{30.,2.,90.,0.,1.,0.,-1.5,-5.,1.,0.,-1.6,-4.,1.,0.,-1.6,1.,0.,-1.6,1.,0.,-1.6,0.4,0.,0.,-8.,1.,0.,1.5,1.,0.,2.6,1.,0.,-1.3,1.,0.,2.1,0.,3.,40.,40.,500.,0.} → 1

Figure 9. Example of the binary numerical labeled dataset.

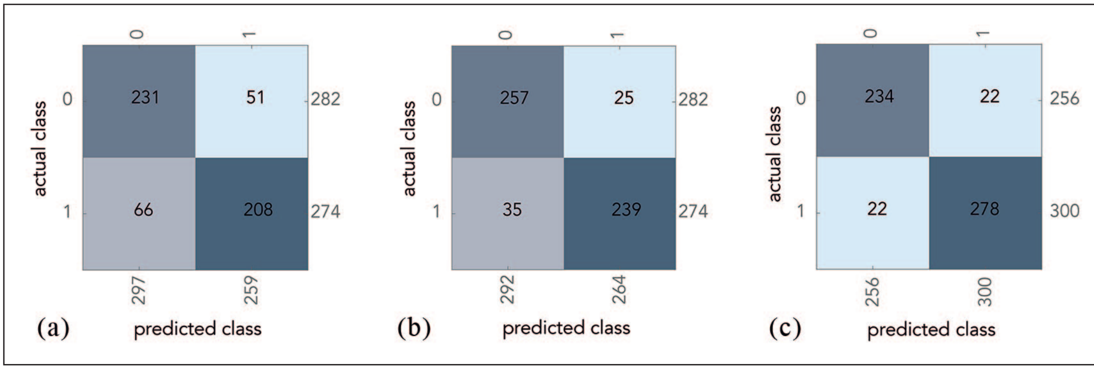

Figure 10. (a) Confusion matrix of decision trees, (b) confusion matrix of random forest, and (c) confusion matrix of gradient-boosting trees.

Table 2. Different algorithms for classification and their performance with the test dataset.

Classifiers	Test examples	Training time	Accuracy	Loss	Precision class 1	Precision class 2
Decision tree	556	3.05 s	0.79043	0.45	0.77777	0.80308
Random forest	556	2.65 s	0.89272	0.34	0.88013	0.90530
Gradient-boosted trees	**556**	**4.46 s**	**0.92036**	**0.27**	**0.91406**	**0.92666**

Gradient-boosted trees was selected since it offered the best performance with the given dataset.

For the regeneration routing, random input values were produced based on the range defined by each of the 43 parameters (30 + 13) of the CEM (see section "Machine-driven generation with CEM"). These parameters were fed in the trained GBT classifier, which then directed these input values into the two learned classes (1 or 0). As the output of the trained classifier is probabilistic, the label of each design option is based on probabilistic confidence. After setting a threshold on the confidence percentage such as 95%, only the options that were above the threshold were selected. The corresponding classified input values were then fed into the CEM to generate 6,751 new preferred design options. After regeneration, the design options were encoded (HOS) and clustered (SOM) to support a final selection, thus finalizing the design experiment.

All the computational tasks described in this section were performed with a Quad-Core 2.7 GHz CPU with no GPU acceleration. Figure 11 shows an overview of the design experiment and the time required for each design operation. Figure 12 illustrates the renderings of the final selection.

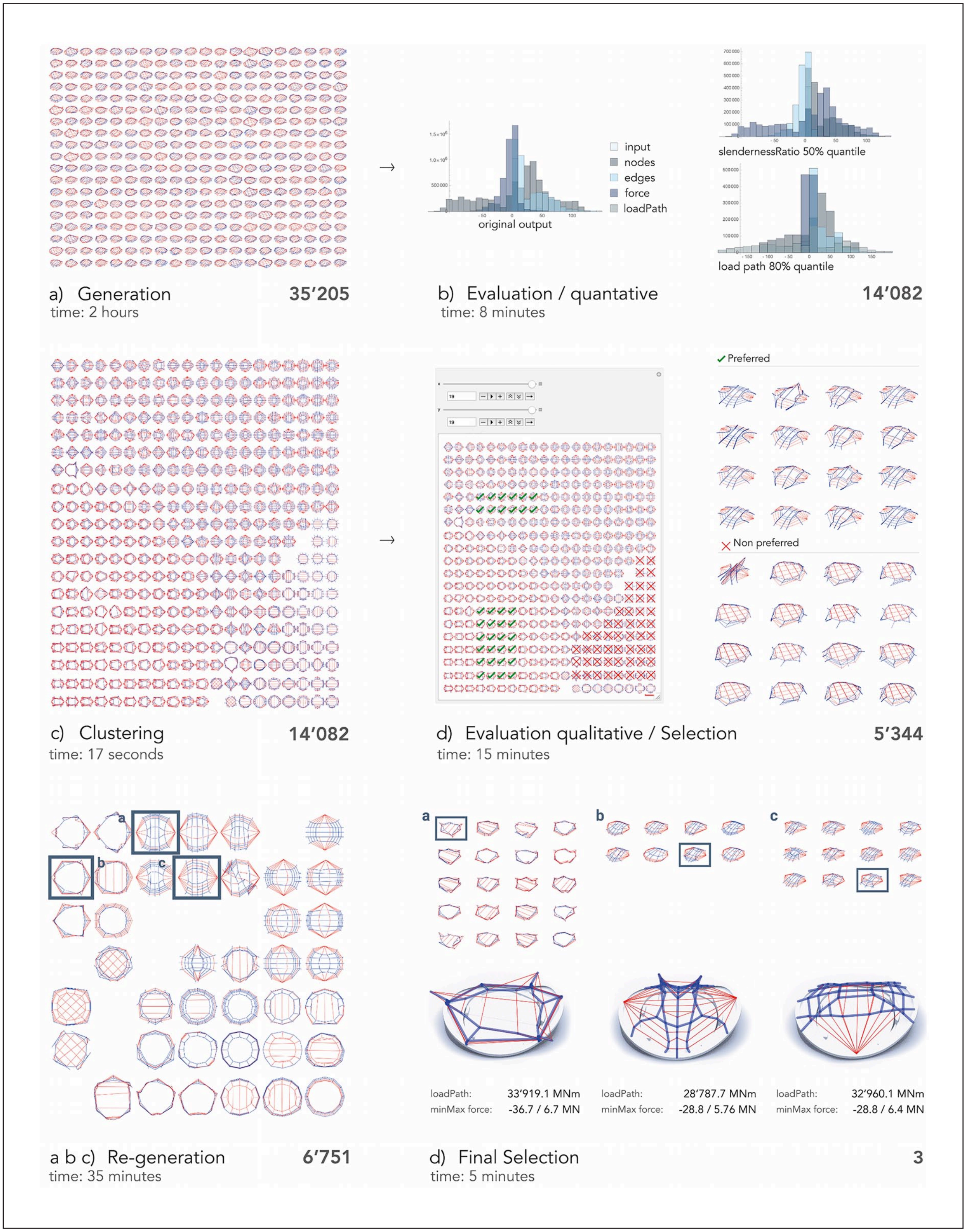

Figure 11. Overview of the design experiment and time required for each design operation.

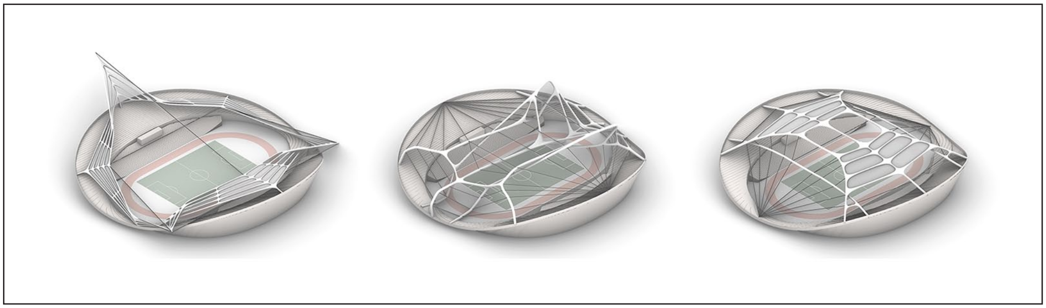

Figure 12. Renderings of the final selection.

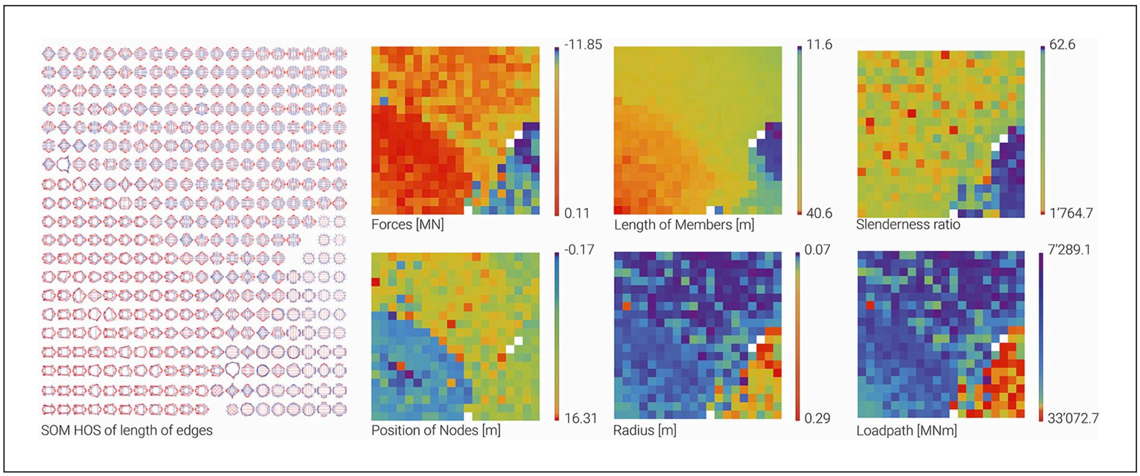

Figure 13. Self-organizing map and BMUs colored according to their data and performance features.

Discussion

The article introduced a computational design framework for the generation of non-standard structural forms in static equilibrium, which takes advantage of the interaction between human and machine intelligence. The framework is built upon basic design operations such as generation, evaluation, clustering, selection, and regeneration. More precisely, the generative operation is based on CEM for the creation of structural forms in static equilibrium as design options. In the operation of clustering, an SOM is employed to organize and represent the diversity of design options in a comprehensible manner according to a set of design features defined by the human designer (Figure 13). Hence, the organized representation enhances the understanding of the full design space and supports decision-making. In addition, HOS is used to encode the relevant aspects of the generated options with a high level of precision. Finally, the supervised classification algorithm GBT is used for the regeneration of the design options.

The proposed approach to structural design supports a shift from typological to topological thinking. Moreover, it proposes that the machine should not only be used to optimize a specific task but rather as an instrument that leverages the creativity of the designer.

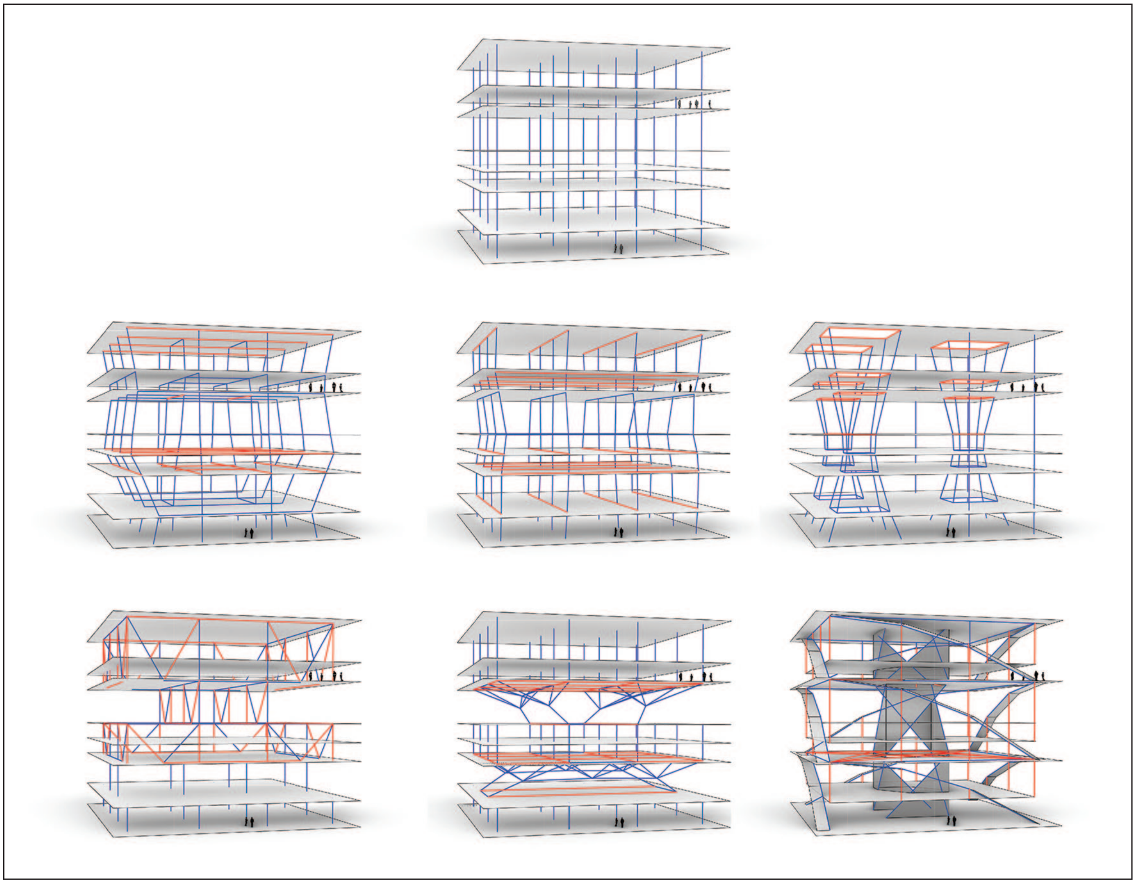

Figure 14. Conceptual design of a multi-story building with varying topology.[59]

The presented application of the proposed framework to the design of a roof for a stadium suggests that the framework is able to support the conception of diverse, yet structurally informed design options in the conceptual design phase without the need for a pre-defined, superimposed protocol of operations. To emphasize the flexibility of the proposed design framework, in Appendix 2, the design process of three additional structural designers, who performed the design task presented in section "Proposed structural design framework," is documented. In future research, the proposed design framework will be further refined and applied to different design scenarios. For example, a possible application could be the conceptual design of multi-story buildings (Figure 14).

Above all, this research tries to contribute to the open-ended question of how the machine could be integrated into a human-driven design exploration in a meaningful way. A common challenge that generally arises when using these generative approaches is the dichotomy between the desired diversity and completeness of the design space on the one hand, and the difficulty for humans to interact with a large amount of data on the other hand. As pointed out by Nicholas Negroponte on the relationship between machines and humans: "The best way to appreciate the merits and consequences of being digital is to reflect on the difference between bits and atoms."[60] One could thus argue that machines and humans have fundamental differences in their internal structure. Indeed, bits can very likely not be used to replace atoms, but they can be used to empower them.

Acknowledgements

The authors would like to thank the Reviewers and Editor for their helpful comments and constructive suggestions.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: K.S.O. receives financial support from the IFTH from the Government of Ecuador. P.O.O., P.D., and V.M. receive salaries from ETH Zürich.

ORCID iD

Karla Saldana Ochoa https://orcid.org/0000-0001-6092-4908

References

1. Mueller CT. *Computational exploration of the structural design space*. PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA, 2014.
2. Brown NC. *Multi-objective optimization for the conceptual design of structures*. Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA, 2016.
3. Engel H. *Structure systems*. 5th ed. Ostfildern: Hatje Cantz Verlag, 2013.
4. Ney L, Adriaenssens S, Devoldere S, et al. *Shaping forces*. Brussels: A+ Editions / Bozarbooks, 2010.
5. Musmeci S. La genesi della forma nelle strutture spaziali. *Parametro* 1979; 10(80): 13–33.
6. Musmeci S. *La statica e le strutture*. Roma: Edizioni Cremonese, 1971.
7. Ingold L and D'Acunto P. Structural folding as a source of research for Sergio Musmeci. In: *Proceedings of the international association for shell and spatial structures symposium 2017*, Hamburg, 25–28 September 2017.
8. Schwartz J. Tragwerkswissenschaft und Tragwerkslehre. In: Flury A (ed.) *Kooperation: Zur Zusammenarbeit von Ingenieur und Architekt*. Basel: Birkhäuser, 2011, p. 243.
9. Au F. A necessary resistance within architect-engineer collaboration. In: Kara H and Georgoulias A (eds) *Interdisciplinary design: new lessons from architecture and engineering*. New York: Actar, 2013, p. 237.
10. Nervi P. *Structures: aesthetics and technology in building*. New York: McGraw-Hill, 1956.
11. Vrontissi M, Castellón González J, D'Acunto P, et al. "Constructing equilibrium": a methodological approach to teach structural design in architecture. In: *Proceedings of the IV international conference on structural engineering*, Madrid, 20–22 June 2018.
12. Addis B. Ove Arup and partners: engineering the built environment/Ove Arup 1895–1988. *Constr Hist* 1995; 11: 103.
13. Menges A. Biomimetic design processes in architecture: morphogenetic and evolutionary computational design. *Bioinspir Biomim* 2012; 7(1): 015003.
14. Rittel H and Webber M. Dilemmas in a general theory of planning. *Pol Sci* 1973; 4: 155–169.
15. Lawson B. *How designers think*. Oxford: Butterworth Architecture, 1990.
16. Lachauer L. *Interactive equilibrium modelling*. PhD Thesis, ETH Zürich, Zürich, 2015.

17. Brown NC and Mueller CT. Design for structural and energy performance of long span buildings using geometric multi-objective optimization. *Energ Build* 2016; 127: 748–761.
18. Erbatur F, Hasançebi O, Tütüncü I, et al. Optimal design of planar and space structures with genetic algorithms. *Comput Struct* 2000; 75(2): 209–224.
19. Fuhrimann L, Moosavi V, Ohlbrock PO, et al. Data-driven design: exploring new structural forms using machine learning and graphic statics. In: *Proceedings of the international association for shell and spatial structures symposium 2018*, Boston, MA, 16–20 July 2018.
20. Terzidis K. *Algorithmic architecture*. Oxford: Oxford Architectural Press, 2006.
21. Oxman R. Theory and design in the first digital age. *Des Stud* 2006; 27(3): 229–265.
22. Darke J. The primary generator and the design process. *Des Stud* 1979; 1(1): 36–44.
23. Cross N. *Designerly ways of knowing*. London: Springer, 2006.
24. Marples D. *The decisions of engineering design*. London: Institute of Engineering Designers, 1960.
25. Ohlbrock PO and D'Acunto P. A computer-aided approach to equilibrium design based on graphic statics and combinatorial variations. *Comput Aid Des* 2020; 121: 102802.
26. D'Acunto P, Jasienski JP, Ohlbrock PO, et al. Vector-based 3D graphic statics: a framework for the design of spatial structures based on the relation between form and forces. *Int J Solid Struct* 2019; 167: 58–70.
27. Schek HJ. The force density method for form finding and computation of general networks. *Comput Meth Appl Mech Eng* 1974; 3(1): 115–134.
28. Geyer P. *Multidisziplinäre Entwurfsoptimierung für das Bauwesen*. PhD Thesis, TU Berlin, Berlin, 2009, p. 34.
29. Harding J and Brandt-Olsen C. Biomorpher: interactive evolution for parametric design. *Int J Archit Comput* 2018; 16(2): 144–163.
30. Shea K, Cagan J and Fenves SJ. A shape annealing approach to optimal truss design with dynamic grouping of members. *J Mech Des* 1997; 119(3): 388–394.
31. Van Buelow P. ParaGen: performative exploration of generative systems. *J Int Assoc Shell Spat Struct* 2012; 53(4): 271–284.
32. Lee J, Fivet C and Mueller C. Modelling with forces: grammar-based graphic statics for diverse architectural structures. In: Thomsen M, Tamke M, Gengnagel C, et al. (eds) *Modelling behaviour*. Cham: Springer, 2015, pp. 491–504.
33. Harding J. Dimensionality reduction for parametric design exploration. In: *Proceedings of the conference on advances in architectural geometry 2016*, Zürich, 9–13 September 2016.
34. Pellegrino S and Calladine CR. Matrix analysis of statically and kinematically indeterminate frameworks. *Int J Solid Struct* 1986; 22: 409–428.
35. Baker W, Beghini L, Mazurek A, et al. Structural innovation: combining classic theories with new technologies. *Eng J* 2015; 52: 203–217.
36. Liew A, Avelino R, Moosavi V, et al. Optimising the load path of compression-only thrust networks through independent sets. *Struct Multidisc Optim* 2019; 60: 231–244.
37. Wong HJ and Hartigan JA. Algorithm AS 136: a k-means clustering algorithm. *J Roy Stat Soc Ser C Appl Stat* 1979; 28(1): 100–108.
38. Maaten LVD and Hinton G. Visualizing data using t-SNE. *J Mach Learn Res* 2008; 9: 2579–2605.
39. McInnes L, Healy J and Melville J. UMAP: uniform manifold approximation and projection for dimension reduction, 2018, https://arxiv.org/pdf/1802.03426.pdf
40. Kohonen T. Self-organized formation of topologically correct feature maps. *Biol Cybern* 1982; 43(1): 59–69.
41. Ireland T and Derix C. An analysis of the poly-dimensionality of living: an experiment in the application of 3-dimensional self-organising maps to evolve form. In: *Proceedings of the 21st international eCAADe conference on digital design (eCAADe 21)*, 2003, pp. 449–455. Graz: Graz University of Technology, http://papers.cumincad.org/data/works/att/ecaade03_449_44_ireland.content.pdf
42. Derix C and Jagannath P. Near futures: associative archetypes. *Archit Des* 2014; 84(5): 130–135.
43. Moosavi V. Computing with contextual numbers, 2014, https://arxiv.org/ftp/arxiv/papers/1408/1408.0889.pdf
44. Moosavi V. Contextual mapping: visualization of high-dimensional spatial patterns in a single geo-map. *Comput Environ Urb Syst* 2017; 61: 1–12.

45. Farid H. Detecting hidden messages using higher-order statistical models. In: Proceedings of the international conference on image processing, Rochester, NY, 22–25 September 2002.
46. DeCarlo LT. On the meaning and use of kurtosis. *Psychol Meth* 1997; 2(3): 292–307.
47. Welling M. Robust higher order statistics. In: *Proceedings of the AISTATS 2005*, vol. 3, p. 7, https://www.ics.uci.edu/~welling/publications/papers/RobCum-aistats.pdf
48. Shackelford M and Simons CL. Metaheuristic design pattern: interactive solution presentation. In: *Proceedings of the companion publication of the 2014 annual conference on genetic and evolutionary computation*, Vancouver, BC, Canada, 12–16 July 2014, pp. 1431–1434. New York: ACM.
49. Coates P, Broughton T and Jackson H. Exploring three-dimensional design worlds using Lindenmayer systems and genetic programming. In: Bentley PJ (ed.) *Evolutionary design by computers*. San Francisco, CA: Morgan Kaufmann Publishers, 1999, pp. 323–341.
50. Kicinger R, Arciszewski T and De Jong K. Evolutionary computation and structural design: a survey of the state-of-the-art. *Comput Struct* 2005; 83(23–24): 1943–1978.
51. Rokach L and Maimon O. *Data mining with decision trees: theory and applications*. Singapore: World Scientific, 2008.
52. Breiman L. Random forests. *Mach Learn* 2001; 45(1): 5–32.
53. Freund Y and Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci* 1997; 55(1): 119–139.
54. Kotsiantis SB, Zaharakis I and Pintelas P. Supervised machine learning: a review of classification techniques. *Emerg Artif Intell Appl Comput Eng* 2007; 160: 3–24.
55. Harding J. *Meta-parametric design: developing a computational approach for early stage collaborative practice*. Doctoral Thesis, University of Bath, Bath, 2014.
56. Ibrahim N and Joyce S. User directed meta parametric design for option exploration. In: *Proceedings of the ACADIA 2019, 2019*, https://www.researchgate.net/publication/336798906_User_Directed_Meta_Parametric_Design_for_Option_Exploration
57. Ohlbrock P, D'Acunto P, Jasienski J, et al. Constraint-driven design with combinatorial equilibrium modelling. In: *Proceedings of the international association for shell and spatial structures symposium 2017*, Hamburg, 25–28 September 2017.
58. Fateman RJ and Gray J. Mastering mathematica: programming methods and applications. *Math Comput* 2006; 64(210): 903.
59. Ohlbrock PO and Schwartz J. Computerunterstützter Entwurf von Stabwerkmodellen. In: *Proceedings of the Tagungsband Baustatik–Baupraxis 14*, Stuttgart, 23–24 March 2020.
60. Negroponte N. Bits and atoms. *Wire Mag* 1995; 1: 1–9.

Appendix 1

The load path of a structure (LP) can be formulated as follows

$$LP = \sum_{i=1}^{n} |f_i| l_i = \sum_{i=1}^{n} LP_i = n\widehat{LP} \tag{3}$$

where LP_i is the member load path and $\widehat{LP}$ is the mean load path. Hence, the load path of a structure LP is an average value proportional to the mean value $\widehat{LP}$. Its capacity to capture the shape of the distribution of member load paths in the structure is limited. Using higher order statistics, it is possible to capture the standard deviation, skewness, and kurtosis of the distribution of the member load paths. Figure 15 shows two design options based on the same ground topology. While having relatively similar load paths (LP), these designs have completely distinct forms as they have completely different distributions of member load paths (LP_i).

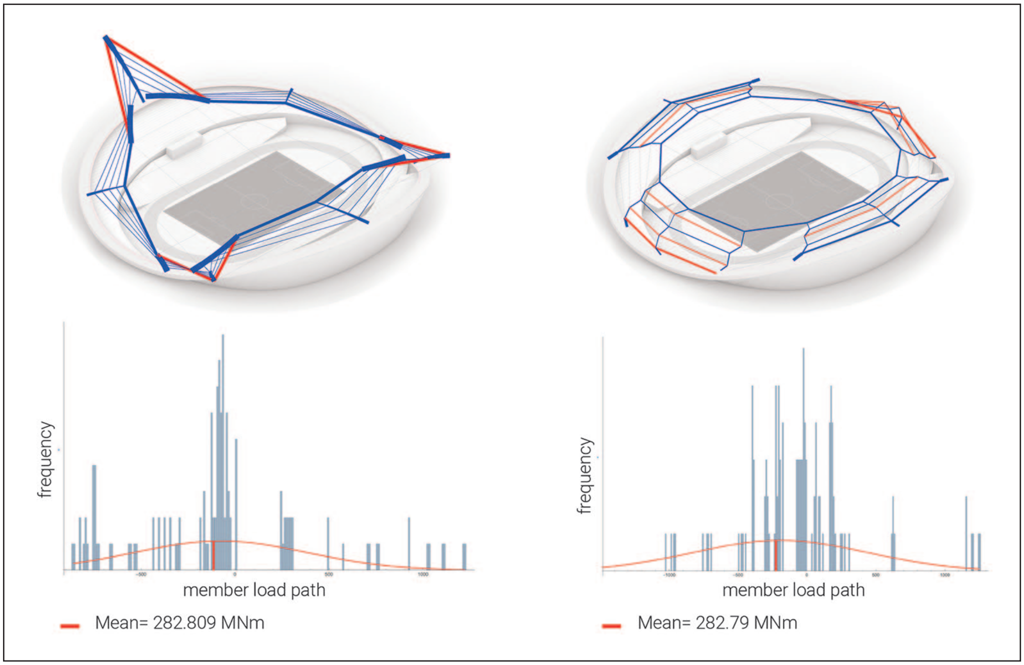

Figure 15. Two randomly generated forms based on the same ground topology with similar load paths (*LP*), but different distribution of member load paths (LP_i).

Appendix 2

To demonstrate the flexibility of the proposed design framework, the design task used as an example in section "Design experiment" is proposed to three different structural designers. The only aspect that is kept constant in all three design sessions is the initial set of the 35,205 generated design options.

Figure 16 shows the individual outcomes for the three different design sessions within the proposed framework. On the left side of each row is a final self-organizing map (SOM) of 7×7 retrieved after regeneration. Through visualizing the elements contained in each selected SOM-cell, the designer could select the preferred ones (top right of each row) from a group of similar design options. Each designer follows an individual sequence of operations (bottom right of each row).

As shown in Figure 16, designers 1 and 2 carried out two design cycles, while designer 3 went through one more cycle before converging to the final three design options. The diversity of these nine options indicates that the present framework can support personalized design sessions with highly differentiated results. Future research will investigate the response of the framework when more design cycles and more designers are involved.

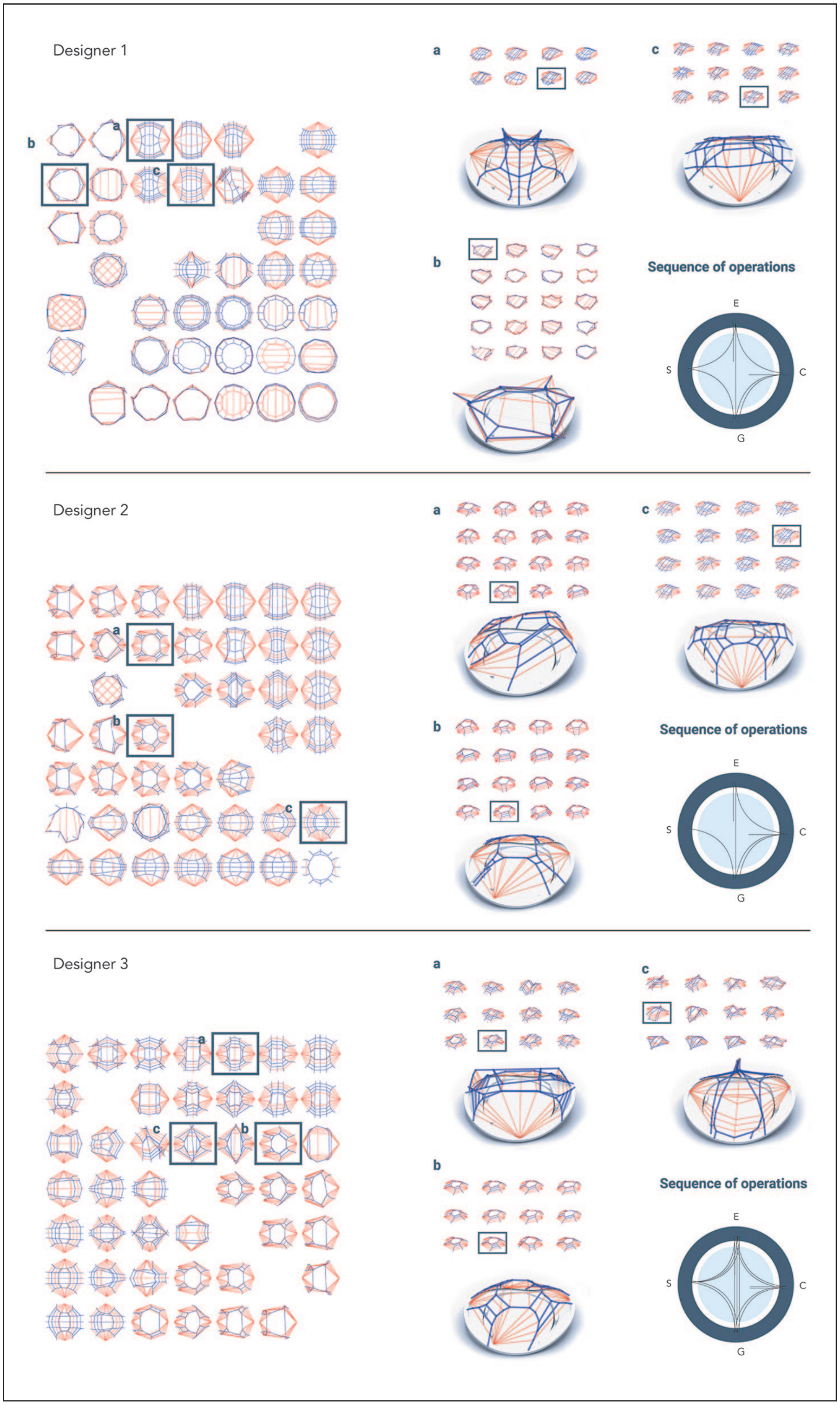

Figure 16. SOM of 7 × 7 after regeneration (left). Three design options selected (middle and top right) and a diagram representing the sequence of operation (bottom, right) for each designer.