COMPUTATIONAL CONCEPTUAL DESIGN – TYPOLOGICAL EXPLORATION OF SPATIAL TRUSS SYSTEMS THROUGH OPTIMIZATION

Jonas WARMUTH[1], Pierluigi D'ACUNTO[2] and Corentin FIVET[1]

[1]Structural Xploration Lab, Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland
Corresponding Author: jonas.warmuth@epfl.ch
[2]Professorship of Structural Design, Technical University of Munich, Germany

Editor's Note: The first author of this paper is one of the four winners of the 2023 Hangai Prize, awarded for outstanding papers that are submitted for presentation and publication at the annual IASS Symposium by younger members of the Association (under 30 years old). It is published here with permission of the editors of the proceedings of the IASS Symposium 2023 "Integration of Design and Fabrication", that was held in July 2023, in Melbourne, Australia.

DOI: https://doi.org/10.20898/j.iass.2023.026

ABSTRACT

Exploring a wide range of relevant design options from the outset is crucial for every sound conceptual design process. Optimization techniques are generally employed to generate well-performing structural design options. However, focusing only on performative criteria may narrow the design brief too early and neglect essential aspects beyond pure performance. In response, this paper introduces a new method for generating structural forms that emphasizes both performance and structural diversity. Applying mixed integer linear programming onto strut-and-tie models, the method employs layout optimization in a new way by (1) generating and modifying custom ground structures and (2) using them to produce systems in static equilibrium that optimize user-balanced sets of custom goals. While the first feature forces a broad exploration of the solution space, the second ensures the generation of only close-to-optimal solutions. Combined, both features provide a new means for generating a trans-topological set of diverse and well-performing designs. The applicability of the method is demonstrated through several case studies. Results show that our formulations allow for the real-time generation of multiple design options, including well-known and uncommon, but no less valid, typologies. Using this approach, designers can move beyond the limitations of established typologies and explore a new variety of structural forms.

Keywords: *computational design, design exploration, interactive design, conceptual design, layout optimization, form-finding*

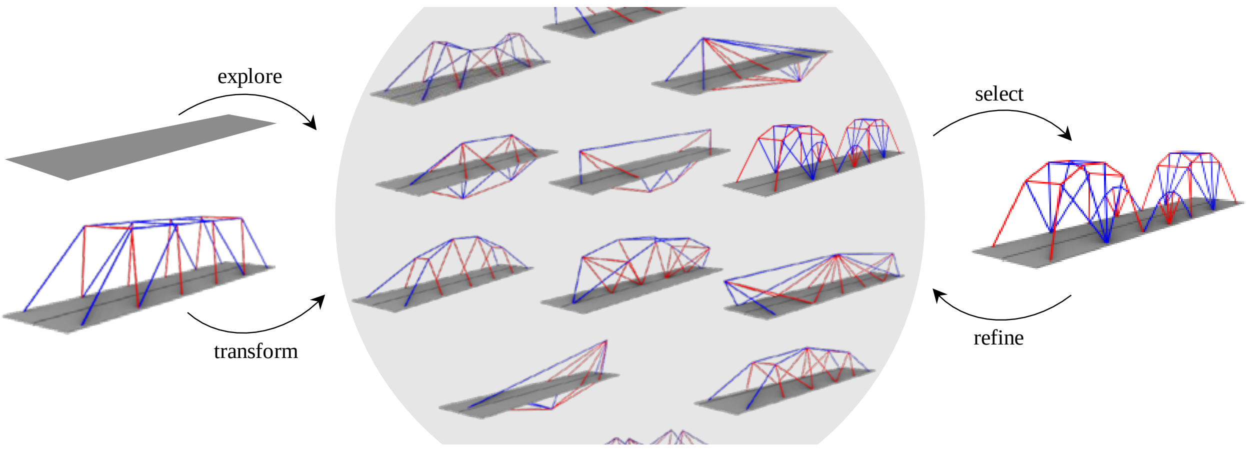

Figure 1. *Designing trans-topological structures by transformation operations and layout optimization.*

1. INTRODUCTION

Traditionally, the process of designing structures involved reliance on hand sketches, physical models, and well-established typologies. With advancements in computational power, computational design opened up a new realm of possibilities, where feedback on an intended design is instantly available, leading to an entirely new approach to exploring and designing structures. Alongside came the opportunity to generate many design alternatives at once, thereby expanding the design space and promoting exploration. With these increased possibilities, the fundamental goals of conceptual structural design could be addressed more effectively, i.e., exploring structures beyond predefined catalogue types while ensuring high performance and fitness to given architectural, economic, and environmental contexts.

1.1. Related work

Over the past decades, various computational tools have been developed to assist structural designers in this conceptual task. Typically, so-called meta-heuristic optimization methods, like genetic algorithms, have been employed to explore structural forms. An early example is ParaGen [1] by von Buelow. This tool operates on structures whose topology is predefined. The evolutionary exploration mechanism controls nodal coordinates and imposes the nodes' relocation until specific design criteria are satisfied.

Mueller and Ochsendorf [2] contributed to the development of performance-informed design, which resulted in the creation of StructureFit. This approach employs genetic algorithms to enable designers to investigate and enhance design candidates by selecting parents for crossover and mutation. The topology of the design candidates is determined by the designer's input, with the examples presented in a 2D format.

Harding and Shepherd [3] worked on meta-parametric design, resulting in Biomorpher [4], a plugin for Grasshopper that employs interactive genetic algorithms to explore and optimize any parametric definition created by the designer.

Mirtsopoulos and Fivet [5] presented a generative approach based on grammar rules and combined it with the interactive genetic algorithms in Biomorpher.

A major advantage of employing meta-heuristic optimization algorithms is their ability to handle diverse problems due to their independence from differentiable (in-)equations. Nonetheless, a significant challenge associated with their use is that the underlying formulations may lead to computationally expensive procedures.

More recently, a generative structural design workflow has been proposed that integrates the Combinatorial Equilibrium Modelling form-finding method [6] with a machine-learning-based clustering technique to facilitate the exploration of multi-dimensional spaces of structural design solutions [7]. While these methods can provide reasonable solutions, it is essential to carefully balance design constraints to ensure practical outcomes. If constraints are excessively rigid, it may hinder design exploration. Furthermore, results generation operates in a black box, making a clear link between input and output challenging.

Regarding performance only, mathematically sound optimization techniques are often used to generate well-performing structures. He et al. [8] have proposed a computationally efficient optimization framework, in which a linear programming truss layout optimization is employed to generate close-to-optimal designs. Park et al. [9] discussed the potential use of layout optimization at the conceptual design stage. Both methods utilize optimization routines as a primary means of generating design alternatives. Bleker [10] developed a hybrid truss layout optimization framework that combined layout optimization with population-based meta-heuristic algorithms to find various design alternatives. This approach introduced a parametrized ground structure approach which allows for more flexibility. However, relying solely on optimization techniques can lead to prematurely narrowed designs without sufficient exploration of other alternatives. This can result in an excessive focus on optimization itself, which may be appropriate for pure engineering tasks but may not allow for sufficient exploration of structural forms when other factors beyond performance are essential, as is the case at the conceptual design stage.

1.2 Contributions

Optimization approaches can still be beneficial if integrated as sub-routines within a broader design process. Therefore, this paper aims to showcase

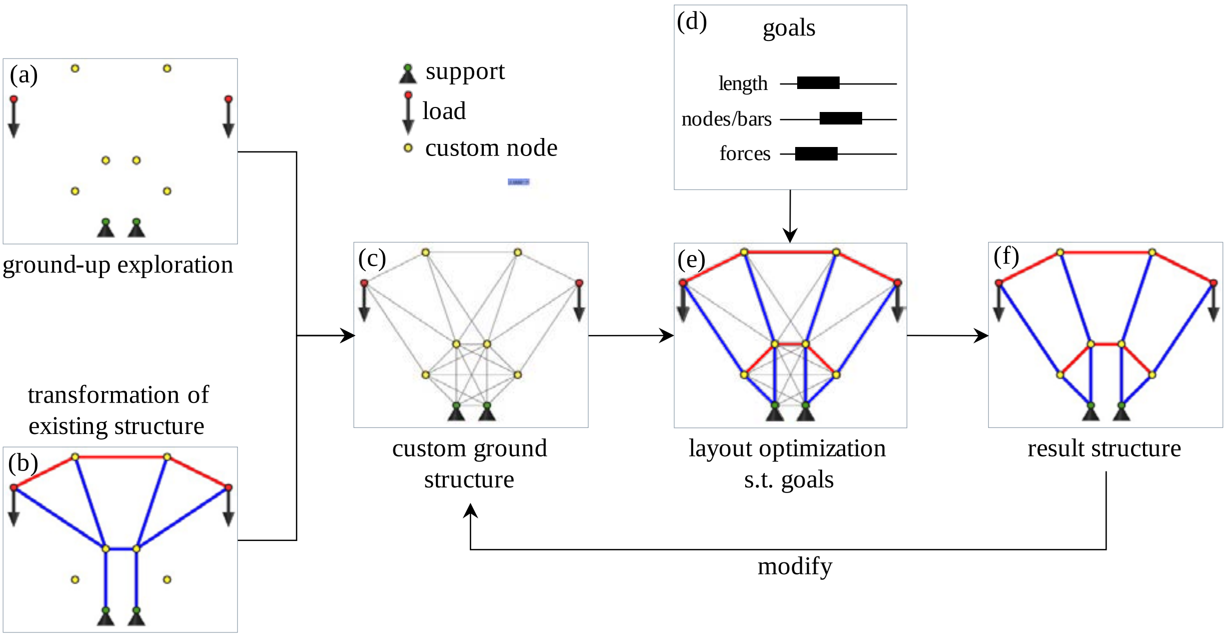

Figure 2. *workflow of the proposed method*

optimization can be leveraged without being the primary driving force but rather a helpful supporting tool in generating structurally sound designs while simultaneously promoting exploration. The focus lies on connecting structural performance with individual design preferences by using layout optimization in a new way. It allows the generation of a diverse set of close-to-optimal structural forms that are visually distinct. A significant advantage of the presented method is that it operates in real-time, allowing designers to manipulate and transform design briefs directly, resulting in an intuitive and seamless workflow. To summarize, this paper's contributions are: (1) the introduction of a new feedback-driven design framework for exploring structural forms in real-time, and (2) the use of discrete layout optimization in a new application focused on typological exploration.

2. METHOD

The method presented in this paper can be employed in two distinct ways: (1) for the creation and exploration of structural forms from the ground up, or (2) for the refinement and local transformation of an already existing design brief. This is achieved using discrete layout optimization to create strut-and-tie models in static equilibrium, as they are known from graphic statics [11], convenient for abstracting a wide range of structures and structural behaviors, and used by many architects and structural engineers. Section 2.1 of this paper outlines the overarching framework of the method, while Section 2.2 delves into the customized ground structure approach. Section 2.3 presents the mathematical formulation for the discrete layout optimization.

2.1 General framework

The general procedure of the method is summarized in Figure 2. The initial step in this process involves supplying input data. For that, two options exist: In option (1) designers can provide support nodes (green) and load nodes (red), as shown in Figure 2a. In addition, custom nodes (yellow) can be placed according to a designer's preference. In option (2) designers can provide an already existing design brief, as shown in Figure 2b. In addition, custom nodes can be added, too. Once the input data is provided, regardless of whether option (1) or (2) was chosen, a custom ground structure is generated by interconnecting all nodes to each other (Figure 2c). It is important to note that the positioning of free nodes serves as one of two junctures where a designer can influence the resulting structure. The second point of influence is through the delineation of design goals, as shown in Figure 2d. In this paper, design goals are limited to three distinct categories: length, number of nodes and bars, and force magnitudes. However, additional, less trivial goals are available and would add even more relevance to the search process. In order to handle these design goals, users can assign intervals, i.e., upper and lower bounds, for each of them. In short, these goals constrain the outcome of the consequent layout optimization step (Figure 2e). Finally, the result of the process is shown in Figure 2f. This result can be further modified by jumping back to step 2c, i.e., by adding, removing or moving nodes to obtain a new custom ground structure. This enables designers to explore different structures for as many iterations as desired.

2.2 Extended ground structure approach

The discrete layout optimization explained in Figure 2 is based on the ground structure approach, first proposed by Dorn et al. [12]. It consists of a fixed grid of nodes interconnected by bars. The optimal structure is found by only keeping a sub-set of all available bars and removing all obsolete bars. Over the years, other strategies for discrete layout optimization have been developed. Some start with a sparse structure and iteratively add nodes or bars to the structure. One well-known method, the growth method, was developed by Martinez et al. [13]. Instead of starting with a densely connected ground structure where all nodes are already available, it adds one node at a time to reduce computational costs. However, it can only deal with statically determinate 2D problems. The approach proposed here can be perceived as combining the two aforementioned strategies and has been shown recently to be similar to the approach proposed by Bleker [10]. Unlike the conventional ground structure approach that employs a predefined grid of nodes, the proposed approach allows the user to place nodes in desired locations, which are then used as nodes of the custom ground structure for the subsequent layout optimization. Therefore, the optimized ground structure is significantly influenced by the nodes placed by a designer. By this method, designers have the ability to enforce their preferences and specify the locations where nodes and bars may occur. However, no guarantee is provided that every node will be kept in the solution. The nodes placed beforehand can be altered or removed anytime during the process, and designers can observe the effects on the final structure in real-time. This allows direct interaction with the design brief. Alternatively, the nodal positions could also be defined randomly in an automated way, which leads to the direct generation of a large set of diverse design outputs.

2.3 Layout optimization

The discrete layout optimization employed in this paper is formulated as an extension of the already mentioned ground structure approach by Dorn et al. [12] as a Linear Programming (LP) problem. Building on this, a Mixed Integer Linear Programming (MILP) formulation is proposed to accommodate the goals shown in Figure 2d.

2.3.1 Linear programming formulation

The discrete layout optimization used here is based on the plastic layout optimization formulation for volume minimization. It consists of setting up a ground structure and solving the following LP problem:

$$\min_{\mathbf{a},\mathbf{q}} V = \mathbf{l}^{\mathrm{T}}\mathbf{a} \tag{1a}$$

subject to

$$\mathbf{B}\mathbf{q} = \mathbf{f} \tag{1b}$$

$$\sigma_T\mathbf{a} - \mathbf{q} \geq 0 \tag{1c}$$

$$\sigma_C\mathbf{a} + \mathbf{q} \geq 0 \tag{1d}$$

$$\mathbf{a} \geq 0 \tag{1e}$$

where V is the volume of all members, $\mathbf{l}$ is a vector of all bar lengths, and $\mathbf{a}$ is a vector of all bar cross-section areas. $\mathbf{B}$ is the equilibrium matrix containing the direction cosines of all bars. $\mathbf{q}$ is the vector of bar internal forces, and $\mathbf{f}$ the vector of external nodal loads. σ_T and σ_C are stress limits in tension and compression.

In general, the objective function of a classical layout optimization is to minimize the volume of a structure. However, in early-stage design, usually no material has yet been decided upon and therefore no stress limits can be stated. This means that no volume can be computed, as, according to Equation 1a, volume is defined as the multiplication of bar lengths with their cross-section areas, while the cross-section area of a bar i is defined as

$$a_i = \frac{q_i}{\sigma} \tag{2}$$

Assuming a constant stress limit in tension and compression, the relation of a bar internal force q_i and its cross section area a_i is constant, too. By choosing a value for this constant, e.g., $\sigma = 1$, then

$$a_i = |q_i| \tag{3}$$

Equation 1a can therefore be rewritten as

$$\min_{\mathbf{q}} V = \mathbf{l}^{\mathrm{T}}|\mathbf{q}| \tag{4}$$

The multiplication of bar internal force magnitudes and corresponding bar lengths was already introduced by Sergio Musmeci in his book *La Statica e le Strutture* [14], where he called this value *total static action Φ*. There, the computation of *Φ* is defined as the sum of the products of bar lengths by their force magnitudes, separating bars in tension (T) from bars in compression (C):

$$\Phi = \sum_{\mathrm{T}} q_{\mathrm{T}} l_{\mathrm{T}} + \sum_{\mathrm{C}} |q_C| l_{\mathrm{C}} \tag{5}$$

Therefore, *total static action* can be seen as a substitute for volume which leads to the following optimization problem to be solved:

$$\min_{\mathbf{q}} \Phi = \mathbf{l}^{\mathrm{T}}|\mathbf{q}| \tag{6a}$$

subject to

$$\mathbf{Bq} = \mathbf{f} \tag{6b}$$

The rewritten optimization problem is a LP problem, too. The advantage of using an LP formulation is that it is computationally efficient, an important feature for any interactive tool at the conceptual design stage. Design goals, as shown in Figure 2d, are not described yet in this formulation. Therefore, the current formulation needs to be extended.

2.3.2 Mixed integer linear programming formulation

One way to extend the previous formulation is to introduce additional constraints and variables. For this, upper and lower bounds for bar lengths and force magnitudes are introduced. In addition, binary assignment variables for nodes and bars are used, as they have already been used by Fairclough and Gilbert [15] and Brütting et al. [16]. The formulation used here is similar to the one of Fairclough and Gilbert. This extends the formulation to a MILP problem, meaning increased computational complexity. However, this extension allows the addition of the goals described in Figure 2d, herein formulated as optimization constraints:

$$\boldsymbol{\rho_L} \leq \boldsymbol{l_k} \leq \boldsymbol{\rho_U} \tag{6c}$$

$$\varphi_L \leq q_k \leq \varphi_L \tag{6d}$$

$$M\mathbf{w} - |\mathbf{q}| \geq 0 \tag{6e}$$

$$Mv_j - \sum_{i \in N_j} |q_i| \geq 0 \tag{6f}$$

$$\mu_L \leq \sum_{i=1}^{n} w_i \leq \mu_U \tag{6g}$$

$$\vartheta_L \leq \sum_{j=1}^{m} v_j \leq \vartheta_U \tag{6h}$$

$$\mathbf{w}, \mathbf{v} \in \{0, 1\} \tag{6i}$$

ρ_L and ρ_U are lower and upper bounds for the length of a bar l_k, and φ_L and φ_U are lower and upper bounds for the internal bar force q_k. Equation 6e denotes whether a potential bar in the ground structure is assigned to the final structure ($w_i = 1$) or not ($w_i = 0$), where **w** is the vector of binary assignment variables for each bar in the ground structure. *M* is a large number, which can be seen as an upper bound for cross-section areas. Equation 6f denotes the existence of a node following the same principle. To provide this, the sum of the areas of all bars ($\sum a_i$) connected to the respective node is used. Here, N_j is the set of bar indices for all connected bars. **v** is another vector of binary assignment variables indicating the existence of a node *j* where $v_j \in \{0,1\}$. μ_L and μ_U are lower and upper bounds for the number of bars and nodes, respectively. This controls the number of bars and nodes in the output structure. Having introduced this, it is now possible to define the aforementioned goal intervals from Figure 2d.

3. RESULTS

All examples in this section were run on an Intel Core i9-11900H @ 2.50GHz with 32.0 GB of RAM. To solve the MILP formulation, Gurobi 10.0.1 [17] was used. Sections 3.1 and 3.2 present a case study in 2D, where Section 3.1 demonstrates the exploration of structures generated from ground-up with no additional input, while Section 3.2 demonstrates the local transformation of an already existing structure. Section 3.3 extends the study to 3D, highlighting the capabilities of the proposed method. Notably, all results presented in this section were obtained in real-time, taking only a few milliseconds to compute.

3.1 Structural exploration

Figure 3a shows the set-up for a bridge case study with three loads of value F = 1MN and two supports. The design domain has a length of L = 20m. In this case study, forces can range from −5F to +5F. Bar lengths are bound between 0 and 5/4L. The maximum number of nodes and bars are 15 and 25, respectively. Figure 3b-3h show the chosen goal intervals on the left-hand side and the resulting structure on the right-hand side. In Figure 3b a classical truss is shown, while Figure 3c shows the result of a layout optimization performed classically with a regular grid of 20×5 nodes and no constraint. Structures 3d-h are then generated following the proposed method, with varying goal intervals.

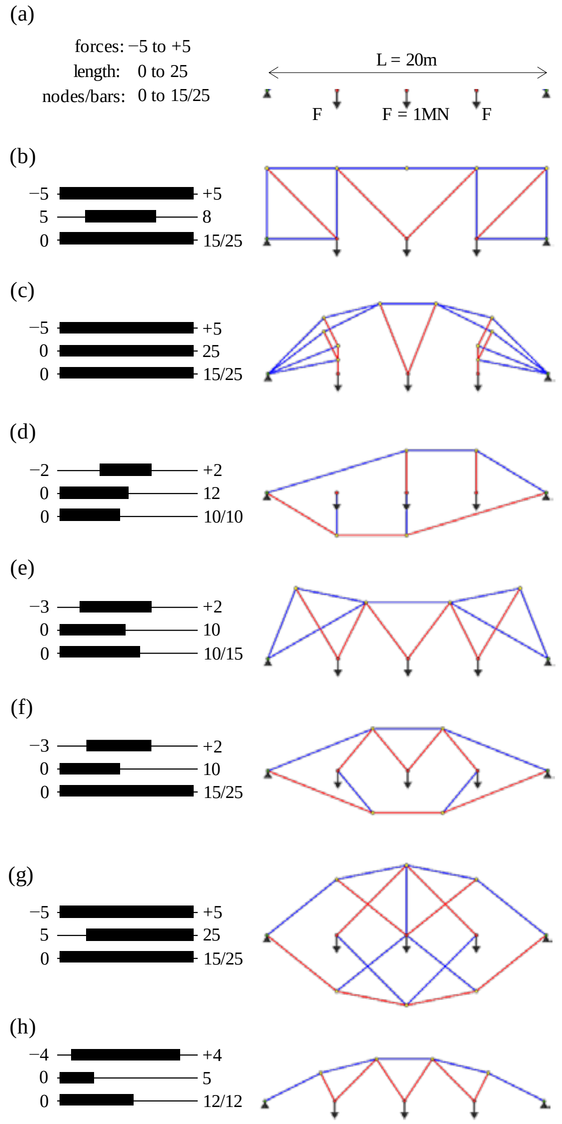

Figure 3: *Case study for a structural exploration from the ground up: (left) structures of different topology and type, (right) goal intervals for length, complexity, and forces.*

Table 1: *Result metrics for the ground-up exploration shown in Figure 3.*

	static action [FL]	nodes [-]	bars [-]	avg. length [L]	gain of stat. act. [-]
(a)	**36.7**	10	14	0.280	± 0%
(b)	**21.3**	16	24	0.186	− 42%
(c)	**28.2**	9	10	0.273	− 23%
(d)	**26.1**	9	13	0.283	− 30%
(e)	**32.4**	9	12	0.274	− 12%
(f)	**34.8**	11	17	0.310	− 4%
(g)	**27.8**	9	11	0.188	− 24%

Table 1 shows the results corresponding to the structures in Figure 3, i.e. their total static action in normalized by FL, the number of nodes and bars, the average bar length normalized by L, and the gain of total static action compared to Structure 3b, the classical truss, i.e. 36.7FL. All obtained structures perform similarly or significantly better than the truss in (b), while (c) performs best with 21.3FL. Except for (g), all structures perform 12% to 42% better. The amount of bars and nodes varies from 11 to 24 and 9 to 16, respectively.

3.2 Structural transformation

In the second case study, the structure from Figure 3h is taken as a starting point and shown in Figure 4a. Support and load conditions remain the same as in the previous example. Also, the goals intervals are the same for each design brief and only custom nodes are added, removed or moved. The intention is to transform the structure in Figure 4a and obtain variations of the same arch-like structural type. Figures 4b-f show the transformed structures with the respective goal intervals. Table 2 shows the result metrics in the same way as Table 1 does. Moreover, all obtained structures perform similarly with total static action deviations up to −22%, compared to the starting structure. Their nodes and bars range from 8 to 10 and 7 to 14, respectively, while the average lengths range from 0.188L to 0.299L.

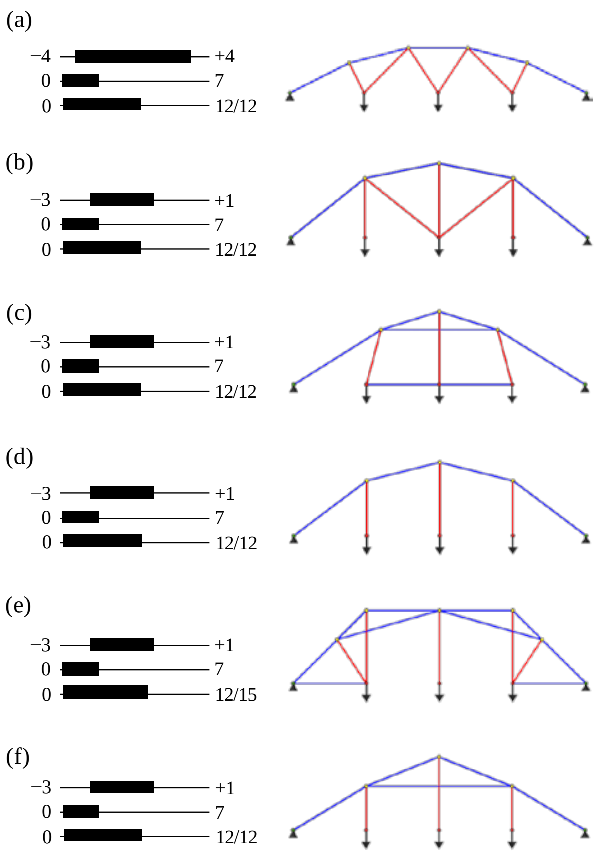

Figure 4: *Case study for a structural transformation: (left) structures of different topology but same type, (right) goal intervals for length, complexity, and forces.*

Table 2: *Result metrics for the structural transformation presented in Figure 4.*

	static action [FL]	nodes [-]	bars [-]	avg. length [L]	gain of stat. act. [-]
(a)	**27.8**	9	11	0.188	± 0%
(b)	**21.9**	8	9	0.271	− 21%
(c)	**24.1**	8	9	0.299	− 13%
(d)	**21.7**	8	7	0.252	− 22%
(e)	**22.7**	10	14	0.253	− 18%
(f)	**24.0**	8	8	0.271	− 14%

3.3 3D bridge structure

The third case study extends the method to the third dimension. Figure 5 displays 40 different bridge structures of varying layout and geometry. The length of the bridges remains at L=20m. Instead of 2 supports and 3 loads there are now 4 supports and 6 loads. The magnitude of each load remains at F=1MN, too. While support and load nodes remain at their position, custom nodes are added, removed and moved by the designer to create alternating structures. A diverse selection of four designer-chosen structures for potential further investigations is shown on the right-hand side.

4. DISCUSSION

The case studies exhibit new variations of forms and types for plane and spatial structures. Some are known types, such as arches and trusses. Others are new, uncategorized forms. Despite this diversity, the study reveals that the resulting structures all perform within a similar range of total static action. This suggests that the method allows designers to choose from diverse forms and types while integrating qualitative preferences without compromising performance.

In addition, the MILP problem introduced in this paper requires significantly fewer computational resources than similar MILP formulations in the literature, e.g. the one from Fairclough and Gilbert [15]. One explanation for this is that the current approach does not employ a densely connected ground structure, resulting in the utilization of fewer nodes and bars.

However, there currently are several limitations and trade-offs. Form-active structures such as arches or cables are not always readily identifiable, as their nodes must be conveniently pre-positioned, which calls for combining current algorithms with complementary geometric search. Furthermore, more complex structures with a significantly bigger number of nodes and bars may pose control challenges, and changing one node at a time only may not lead to a significant change. To address these limitations, the development of methods that bundle sets of nodes for efficient analysis and transformation is required. Also, even though this tool is meant for early-stage conceptual design, integrating early stability checks and construction feasibility checks would make the transition to the subsequent phases of the design process more straightforward. Finally, developing additional goal parameters should allow users to gain control over more complex formal features and, hence, an even more straightforward control over the search process. These improvements will be the subject of future publications by the authors.

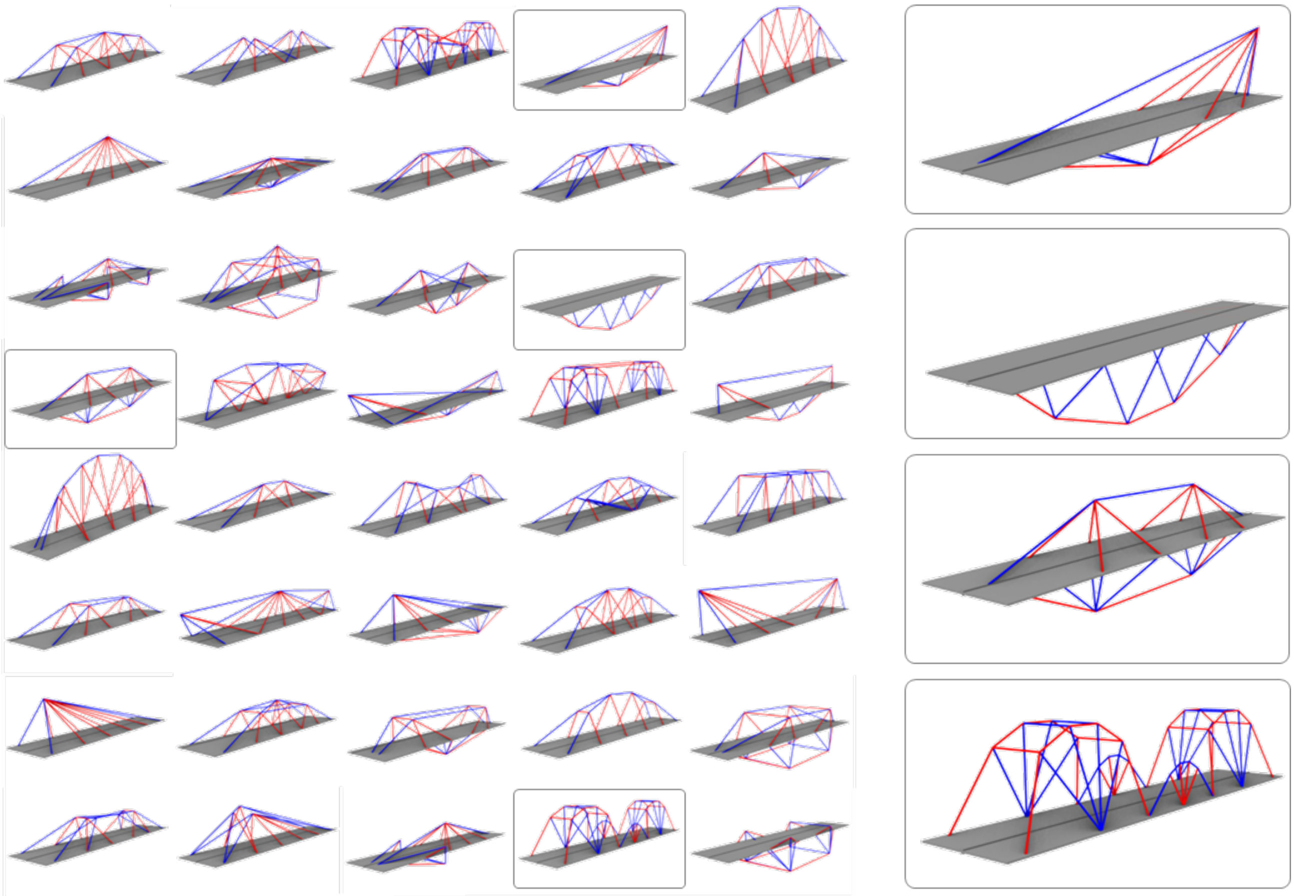

Figure 5: *Sample of 40 generated structures with four selected structures to further investigate.*

5. CONCLUSION

This paper introduced a new method for generating and exploring structural forms for the conceptual structural phase of the design process. The approach allows designers to choose between the unbridled exploration of various forms or the refinement of an already advanced design brief. The method uses layout optimization in a new way, which involves the creation of a bespoke, irregular, and parametric ground structure and the subsequent generation of structures in equilibrium based on different design goals. Primary advantages of this method are (1) its ability to navigate a solution space that is unrestrained from predefined typologies, (2) its ability to operate in real-time, enabling designers to modify their intended design interactively, and (3) its ability to generate only structural typologies that make sense from both points of view of static equilibrium and construction rationality or other custom non-quantifiable goals.

Future developments of this method relate to implementing a geometry optimization step to allow for the generation of form-active structures and the implementation of additional design goals, hence allowing more design opportunities to tame the search for new structural typologies. The algorithms will soon be made available as a Grasshopper plugin for Rhino3D.

With this method, designers may break away from established typologies and quickly explore a vast new set of sound structural forms. Furthermore, this method facilitates the simultaneous consideration of both performance and diversity, ensuring that the final design is not only well-performing but also distinctive and inventive. Therefore, this introduced method constitutes a valuable addition to the computational conceptual design process, providing designers with a persuasive tool for generating and exploring structural forms. The approach offers flexibility and adaptability to designers, empowering them to develop creative, high-quality designs that cater to the specific requirements of their projects.

REFERENCES

[1] P. von Buelow, "Paragen: Performative Exploration of Generative Systems," *J. Int. Assoc. Shell Spat. Struct., vol. 53, no. 4, pp. 271–284, Dec. 2012.*

[2] C. T. Mueller and J. A. Ochsendorf, "Combining structural performance and designer preferences in evolutionary design space exploration," *Autom. Constr., vol. 52, pp. 70–82, Apr. 2015,* (DOI : 10.1016/j.autcon.2015.02.011)

[3] J. E. Harding and P. Shepherd, "Meta-Parametric Design," *Des. Stud., vol. 52, pp. 73–95, Sep. 2017,* (DOI : 10.1016/j.destud.2016.09.005)

[4] J. Harding and C. Brandt-Olsen, "Biomorpher: Interactive evolution for parametric design," Int. *J. Archit. Comput., vol. 16, pp. 144–163, Jun. 2018,* (DOI: 10.1177/1478077118778579)

[5] I. Mirtsopoulos and C. Fivet, "Structural topology exploration through policy-based generation of equilibrium representations," *Comput.-Aided Des., vol. 160, p. 103518, Jul. 2023,* (DOI : 10.1016/j.cad.2023.103518)

[6] P. O. Ohlbrock and P. D'Acunto, "A Computer-Aided Approach to Equilibrium Design Based on Graphic Statics and Combinatorial Variations," *Comput.-Aided Des., vol. 121, p. 102802, Apr. 2020,* (DOI : 10.1016/j.cad.2019.102802)

[7] K. Saldana Ochoa, P. O. Ohlbrock, P. D'Acunto, and V. Moosavi, "Beyond typologies, beyond optimization: Exploring novel structural forms at the interface of human and machine intelligence," *Int. J. Archit. Comput., vol. 19, no. 3, pp. 466–490, Sep. 2021,* (DOI : 10.1177/1478077120943062)

[8] L. He et al., "Optimization-driven conceptual design of truss structures in a parametric modelling environment," *Structures, vol. 37, pp. 469–482, Mar. 2022,* (DOI : 10.1016/j.istruc.2021.12.048)

[9] P. Park, M. Gilbert, A. Tyas, and O. Popovic-Larsen, "Potential Use of Structural Layout Optimization at the Conceptual Design Stage," Int. J. *Archit. Comput., vol. 10, no. 1, pp. 13–32, Mar. 2012,* (DOI : 10.1260/1478-0771.10.1.13)

[10] L. Bleker, "Hybrid truss layout optimization for generating multiple design alternatives," 2021. (DOI : 10.13140/RG.2.2.34230.88648)

[11] P. D'Acunto, J.-P. Jasienski, P. O. Ohlbrock, C. Fivet, J. Schwartz, and D. Zastavni, "Vector-based 3D graphic statics: A framework for the design of spatial structures based on the relation between form and forces," *Int. J. Solids Struct., vol. 167, pp. 58–70, Aug. 2019,* (DOI : 10.1016/j.ijsolstr.2019.02.008)

[12] W. Dorn, R. Gomory, and H. J. Greenberg, "Automatic design of optimal structures," *J. Mècanique, vol. 3, pp. 25–52, 1964.*

[13] P. Martínez, P. Martí, and O. M. Querin, "Growth method for size, topology, and geometry optimization of truss structures," *Struct. Multidiscip. Optim., vol. 33, no. 1, pp. 13–26, Jan. 2007,* (DOI : 10.1007/s00158-006-0043-9)

[14] S. Musmeci, La statica e le strutture. 1971.

[15] H. Fairclough and M. Gilbert, "Layout optimization of simplified trusses using mixed integer linear programming with runtime generation of constraints," *Struct. Multidiscip. Optim., vol. 61, no. 5, pp. 1977–1999,* May 2020, (DOI: 10.1007/s00158-019-02449-7)

[16] J. Brütting, J. Desruelle, G. Senatore, and C. Fivet, "Design of Truss Structures Through Reuse," *Structures, vol. 18, pp. 128–137, 2019,* (DOI : 10.1016/j.istruc.2018.11.006)

[17] "Gurobi Optimization LLC." 2023. Accessed: Mar. 30, 2023. [Online]. Available: https://www.gurobi.com/