Proceedings of the IASS Annual Symposium 2023
Integration of Design and Fabrication
10–14 July 2023, Melbourne, Australia
Y.M. Xie, J. Burry, T.U. Lee and J. Ma (eds.)

Computational conceptual design – typological exploration of spatial truss systems through optimization

Jonas WARMUTH[*], Pierluigi D'ACUNTO[a], Corentin FIVET[b]

[*] Structural Xploration Lab, Ecole Polytechnique Fédérale de Lausanne (EPFL)
Fribourg Switzerland, Passage du Cardinal 13B, CH-1700 Fribourg
jonas.warmuth@epfl.ch

[a] Professorship of Structural Design, Technical University of Munich, Munich, Germany
[b] Structural Xploration Lab, Ecole Polytechnique Fédérale de Lausanne (EPFL), Fribourg Switzerland, Passage du Cardinal 13B, CH-1700 Fribourg

Abstract

Exploring a wide range of design options is crucial for every sound conceptual design process. In the conventional structural design workflow, optimization techniques are generally employed to generate well-performing design options. However, focusing only on performative criteria may result in a lack of design exploration and diversity, narrowing down the design brief too early and neglecting essential aspects beyond pure performance. In response, this paper introduces a new method for the generation of structural forms that emphasizes both performance and the exploration of a diverse set of design briefs. The method employs layout optimization in a new way and aims to find close-to-optimal yet diverse structures. First, a parametric ground structure is generated. Second, this ground structure is used to produce a structure in equilibrium based on different user-defined goals. By varying ground structures and balancing the different goals, the method produces a series of designs that are both diverse and well-performing. The applicability of the method is demonstrated through several case studies. Results show that this method allows for the real-time generation of multiple design options, including both well-known and less well-known, but not 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

1. Introduction

Traditionally, the process of designing structures involved reliance on hand-sketches, physical models, and well-established typologies. With advancements in computational power this process was extended to include digital format. As a result, designers have been enabled to integrate geometric and performance metrics in the design.

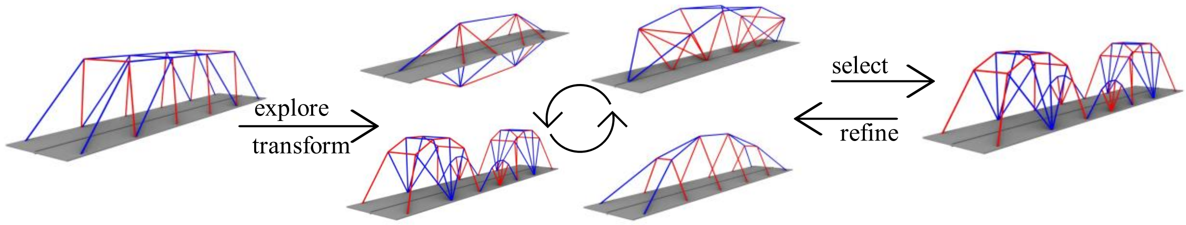

Figure 1. Designing structures by transformation operations and layout optimization

Computational design has opened up a new realm of design possibilities, where feedback on an intended design is instantly available, leading to an entirely new approach of exploring and designing structures. This also brought the benefit of generating a significantly larger number of design alternatives, thereby expanding the design space and promoting exploration. With these increased possibilities, the fundamental goals of conceptual structural design could be addressed more effectively – exploring structures beyond pre-defined catalog types while ensuring adequate performance.

1.1 Related work

In this regard, computational tools have been developed to assist designers in this task. Typically, so called meta-heuristic optimization methods, such as 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 plug-in 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 capability to handle a diverse array of problems due to their independence from differentiable (in-)equations. Nonetheless, a significant challenge associated with their use is that the underlying computations 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 good solutions, it is essential to carefully balance design constraints to ensure practical outcomes. If constraints are excessively rigid, it may hinder design exploration. Furthermore, the process of generating results is a black box, making it challenging to establish a clear link between input and output. 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. 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 important, as it is the case in the conceptual design stage.

1.2 Contributions

Optimization approaches can still be beneficial if they are integrated as sub-routines within a broader design process. Therefore, this paper aims to showcase how optimization can be leveraged without being the primary driving force, but rather a helpful 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. This allows to generate a diverse set of close-to-optimal structural forms that are yet visually distinct. A significant advantage of the presented method is that it operates in real-time allowing designers to directly manipulate and transform design briefs, resulting in an intuitive and seamless workflow. The method achieves a balance between exploration and rationality by imposing customized node-placing in combination with discrete layout optimization. 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 by using discrete layout optimization to create strut-and-tie models in static equilibrium, as they are known from graphic statics [10] 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. In Section 2.3, the mathematical formulation for the discrete layout optimization is presented.

2.1 General framework

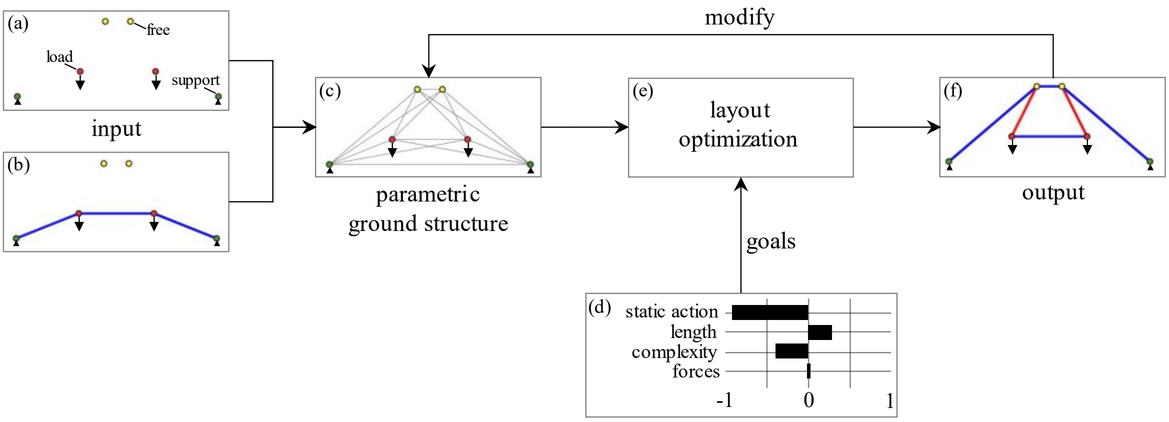

Figure 2. Workflow of the presented method

The general procedure of the method is summarized in Figure 2. The initial step in this process involves supplying input data. Figure 2(a) illustrates the three types of nodes that comprise the input data: support nodes (green), load nodes (red), and free nodes (yellow). Additionally, as illustrated in Figure 2(b), a pre-existing structure can be incorporated as needed. Once the input data is provided, a ground structure is generated by interconnecting all nodes (Figure 2c). It is important to note that the positioning of the nodes serves as one of two junctures where the designer can influence the resulting structure. The second point of influence is through the delineation of design goals, as shown in Figure 2(d). In this paper, these goals comprise four distinct categories: volume, length, complexity, and forces. However, additional, less trivial goals are available and would add even more relevance to the search process.

In order to weigh these design goals against each other, users can assign a real value between -1 and 1 for each of them, where a value of -1 represents a minimization, a value of +1 a maximization, and a value of 0 a neutral value indicating that the goal is not a priority for the optimization. When optimizing for volume, only values between -1 and 0 make sense since maximizing the volume of a structure is usually not desirable. However, for the sake of consistency, the range 0 to +1 can also be used. Regarding length, a value of -1 aims for shorter bars in the output structure, a value of +1 for longer bars. Complexity is derived from the number of bars and nodes existing in the output structure, where -1 aims for as less bars and nodes as possible and +1 for as many as possible. The 'Forces' input works slightly differently. A value of -1 requests all bars in compression, while a value of +1 requests all bars in tension. Values in between interpolate between both extremes. This means that a value of +0.8 tries to have most of the bars in tension but allows for some bars in compression. In summary, these goals influence the outcome of the layout optimization (Figure 2e) together with the custom ground structure from Figure 2(c). Another peculiarity of the method used here is that instead of volume, the so-called static action is used:

$$\sum_{\mathrm{T}} \mathrm{F_T L_T} + \sum_{\mathrm{C}} \mathrm{F_C L_C} \tag{1}$$

Static action is calculated as the sum of the products of bar lengths by their force magnitude, separating bars in tension (T) from bars in compression (C). In early-stage design where no material has yet been decided upon, or where a constant material resistance in tension and compression is assumed throughout, static action can be understood as a substitute for volume. Thus, the terms volume and static action are used equally in this paper. Finally, the result of the process is shown in Figure 2(f). This result can be further modified by jumping back to step 2(c) i.e., by adding or removing nodes or by moving nodes to obtain a new custom ground structure. This enables designers to explore different kinds of structures for as many iterations wanted.

2.2 Extended ground structure approach

The discrete layout optimization used here is based on the ground structure approach, first proposed by Dorn et al. [11]. It consists in 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 were 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. [12]. 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 cost. However, it can only deal with statically determinate 2D problems. The approach proposed here can be perceived as a combination of the two aforementioned strategies. 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 interconnected to create a custom ground structure for the subsequent topology optimization. Therefore, the optimized ground structure is significantly influenced by the nodes selected by the designer. By this method, the designer has the ability to enforce their preferences and specify the locations where nodes and bars may or may not occur. The nodes that are placed beforehand can be altered or removed at any point during the process, and the designer 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. [11] as a LP problem. Building on this, a mixed integer linear programming (MILP) formulation is proposed to allow for the definition of the goals explained in Figure 2(d).

2.3.1 Linear programming formulation

As already mentioned, 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:

$$\min_{\mathbf{a},\mathbf{q}} V = \mathbf{l}^T \mathbf{a} \tag{2a}$$

subject to

$$\mathbf{B}\mathbf{q} = \mathbf{f} \tag{2b}$$

$$\sigma_T \mathbf{a} - \mathbf{q}_k \geq 0 \quad (2c)$$

$$\sigma_C \mathbf{a} + \boldsymbol{q}_k \geq 0 \quad (2d)$$

$$\mathbf{a} \geq 0 \quad (2e)$$

where V is the total static action (≈ volume) of all members, **l** is a vector of all bar lengths, and **a** is a vector of all bar cross section areas. **B** is the equilibrium matrix containing the direction cosines of all bars. **q** is the vector of bar internal forces, and **f** the vector of external nodal loads. σ_T and σ_C are stress limits in tension and compression. The advantage of using this LP formulation is that it is computationally efficient which is an important feature for any interactive tool for the conceptual design stage. Design goals, as described in Figure 2(d), 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, binary assignment variables for nodes and bars are used, as they have already been used by Fairclough and Gilbert [13] and Brütting et al. [14]. The formulation used here is similar to the one of Fairclough and Gilbert. This extends the formulation to a MILP problem which can mean that computational complexity increases. However, it allows the addition of several goals, herein formulated as constraints:

$$M\mathbf{w} - \mathbf{a} \geq 0 \quad (2f)$$

$$Mv_j - \sum_{i \in N_j} a_i \geq 0 \quad (2g)$$

$$\mu_L \leq \sum_{i=1}^{n} w_i \leq \mu_U \quad (2h)$$

$$\vartheta_L \leq \sum_{j=1}^{m} v_j \leq \vartheta_U \quad (2i)$$

$$\mathbf{w}, \mathbf{v} \in \{0, 1\} \quad (2j)$$

Equation 2f 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 sufficiently large number which can be seen as an upper bound for cross section areas. Equation 2g 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 allows to control the number of bars and nodes in the output structure.

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 [15] 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 scratch with no additional input, while Section 3.2 demonstrates the local transformation of an existing structure to achieve structures with similar performance and appearance. 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 3(a) shows the setup for this case study with three loads of value *F* and two supports. The design domain is indicated by the gray box and has a length of *L* and a height of *L/2*. Figure 3(b) shows the goals as black bars ranging from -1 to +1 (see Section 2.1). These goals can be changed to obtain different results. Support and load nodes do not move, while free nodes may move, vanish, or appear.

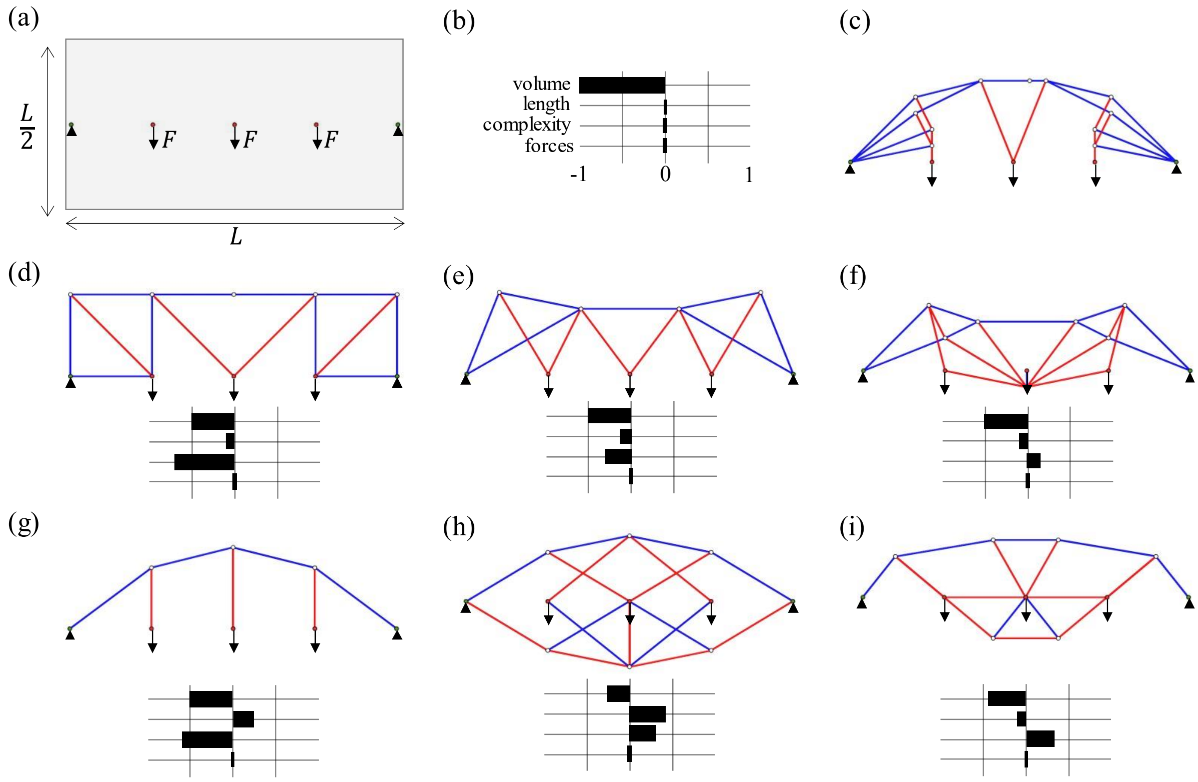

Figure 3. Case study for a structural exploration from ground up: (a) initial set up of load and support conditions; (b) input goals leading to (c) first iteration; (d-i) other results (top) resulting from other choices of input goals (bottom).

Table 1. Result metrics for the ground up exploration presented on Figure 3.

Name	Static action [FL]	Nodes [-]	Bars [-]	avg. Length [L]	Difference [-]
(c)	**21.3**	16	24	0.186	- 42%
(d)	**36.7**	10	14	0.280	± 0%
(e)	**26.1**	9	13	0.283	- 29%
(f)	**30.2**	12	20	0.211	- 18%
(g)	**21.7**	8	7	0.252	- 41%
(h)	**37.5**	11	17	0.284	+ 2%
(i)	**28.5**	11	16	0.208	- 22%

The obtained structures can be observed in Figure 3(c)-(i). Result (c) shows the 'optimal' structure if only the minimization of static action is a goal and a fully connected dense ground structure is available. It is rather a theoretical result used to benchmark results (d)-(i). Structure (d) denotes a 'classical' truss structure. Therefore, (d) is used as a reference to measure the difference of static action. The black bars underneath indicate the values chosen for the design goals. Table 1 shows the results belonging to the structures in Figure 3, containing their static action, normalized by *FL*, the number of nodes and bars, the average length of the bars, normalized by L, and the difference of static action compared to the one of the reference structure (d), i.e. 36.7FL. All obtained structures perform similarly or significantly better than the truss in (d), while (c) performs best with 21.3FL. Except for (h) all structures perform 18% to 41% better. The amount of bars and nodes used is varying from 7 to 24 and 8 to 16, respectively.

3.2 Structural transformation

In this case study, the structure from Figure 3(g) is taken as a starting point and shown in Figure 4(a). Support and load conditions remain the same as in the previous example. Again, support and load nodes do not move, while free nodes may move, vanish, or appear. The intention is to transform Figure 4(a) to obtain similar arch-like looking results and explore the range of possibilities for a given type of structure. Figure 4(b)-(f) show the obtained transformed structures with the respective goals underneath. Table 2 shows the result metrics in the same way as Table 1 does.

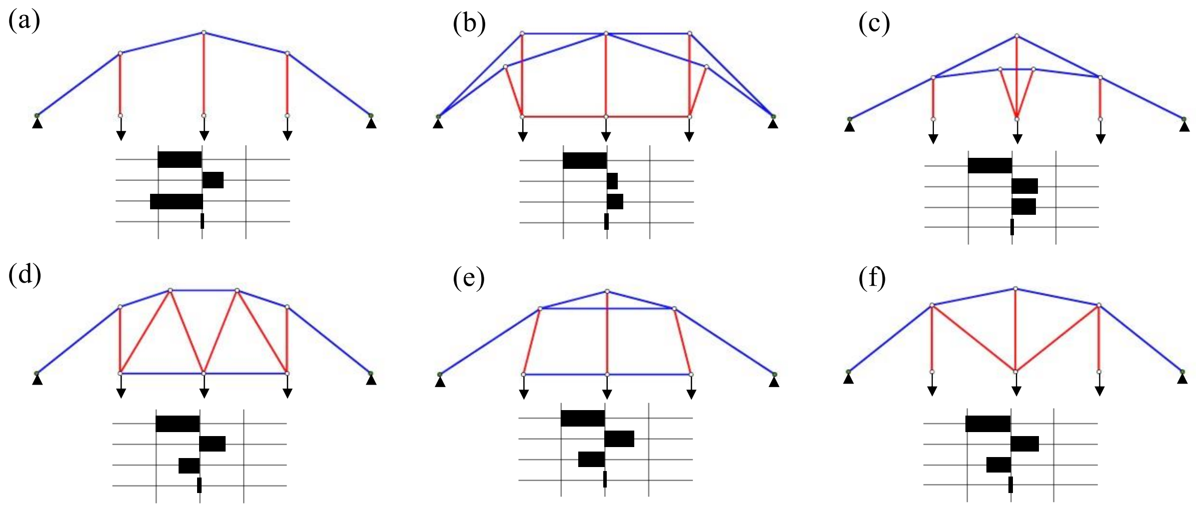

Figure 4. Case study for a structural transformation: (a) initial structure; (b-f) transformations (top) resulting from a change of input goals (bottom).

Table 2. Result metric for the structural transformation

Name	Static action [FL]	Nodes [-]	Bars [-]	avg. length [L]	Difference [-]
(a)	**21.7**	8	7	0.252	± 0
(b)	**22.2**	10	13	0.300	+ 2%
(c)	**25.9**	10	13	0.252	+ 19%
(d)	**22.2**	9	13	0.244	+ 2%
(e)	**24.1**	8	9	0.299	+ 11%
(f)	**21.9**	8	9	0.271	+ 1%

Moreover, all obtained structures perform similarly with static action deviations up to +11%, except for (c) deviating by +19% compared to the starting structure. Their amount of nodes and bars ranges from 8 to 10 and 7 to 13, respectively, while the average lengths ranges from 0.244 to 0.300.

3.3 3D bridge structure

This case study serves as an illustrative example of the extension of the introduced method to the third dimension, as demonstrated in Figure 5 where 40 different bridge structures of varying layout and geometry are displayed. Underneath, a diverse selection of four structures for potential further investigations is shown.

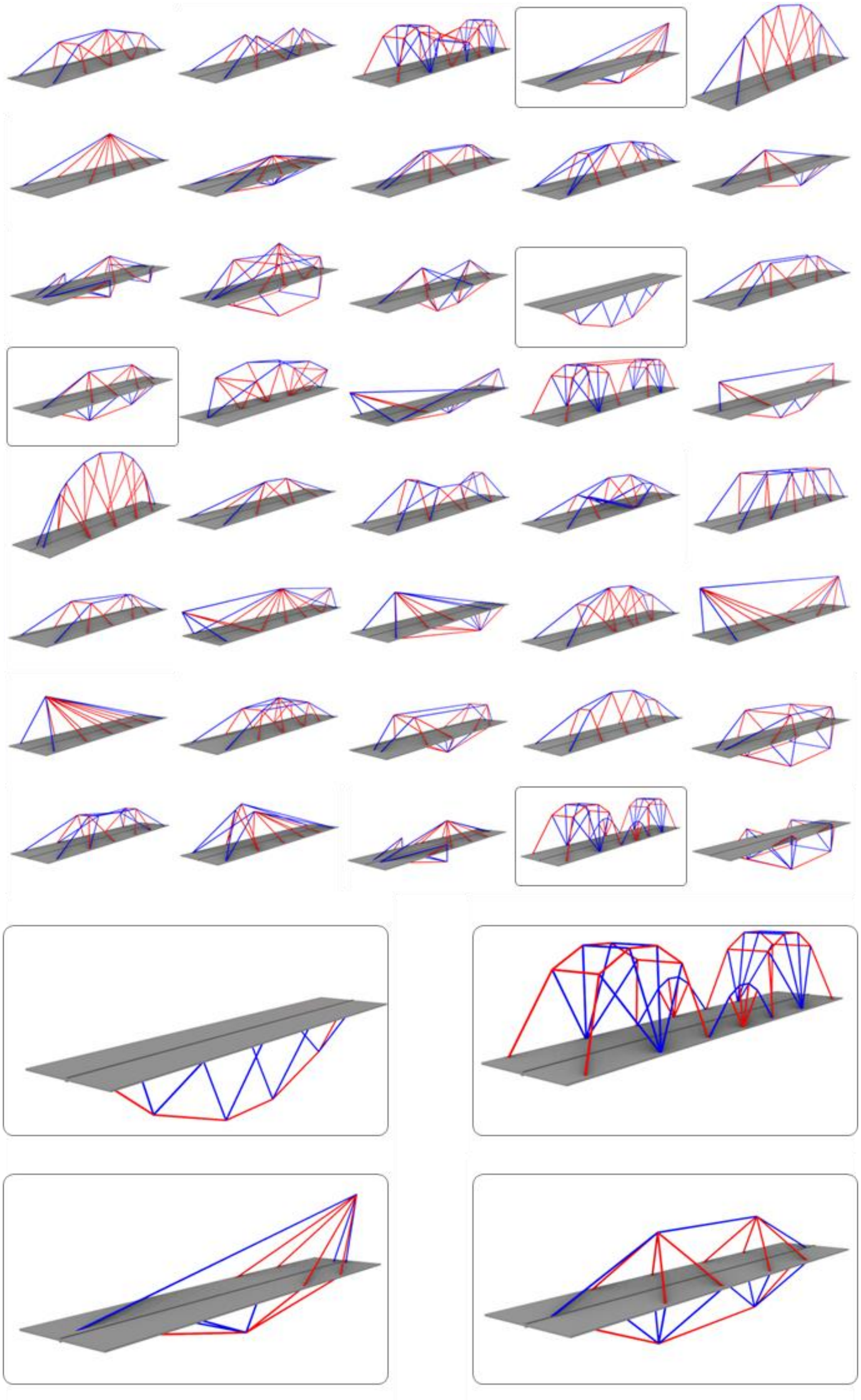

Figure 5. Sample of 40 generated structures with four selected structures to further investigate

4. Discussion

The present study reveals a diverse range of forms and types for spatial structures, as demonstrated by the case studies. These structures exhibit a variety of known types, such as arches and trusses, as well as new, uncategorized forms. Remarkably, the study finds that the diverse array of structures performs within a similar range in terms of static action. This suggests that designers may choose from a variety of forms and types for qualitative preferences without compromising performance. Moreover, the MILP problem introduced in this paper requires significantly less computational resources compared to similar MILP formulations in the literature, e.g. the one from Fairclough and Gilbert [13]. 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 larger number of nodes and bars may pose challenges in control, 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, the integration of early stability checks and construction feasibility checks would make the transition to the next phases of the design process more straightforward. Finally, the development of 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.

5. Conclusion

This paper introduced a new method for the generation and exploration of structural forms for the conceptual structural phase of the design process. The approach provides designers with the flexibility 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 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 only generate structural typologies that make sense from both points of views of static equilibrium and construction rationality or other custom goals.

Future developments of this method relate to the implementation of 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 of 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 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 may also be 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, Eds., "Grammar-based generation of bar networks in static equilibrium with bounded bar lengths," *Proc. IASS Annu. Symp. 202021 7th Int. Conf. Spat. Struct.*, 2021.

[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] 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.

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

[12] 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.

[13] 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.

[14] 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.

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