International Journal of Space Structures
XX(X):1–16

DOI: 10.1177/ToBeAssigned
www.sagepub.com/

Automatic generation of diverse equilibrium structures through shape grammars and graphic statics

Juney Lee[1], Caitlin Mueller[2] and Corentin Fivet[2]

Abstract

This paper presents a computational design methodology that integrates generative (architectural) and analytical (engineering) procedures into a simultaneous design process. By combining shape grammars and graphic statics, the proposed methodology enables: 1) rapid generation of diverse, yet statically equilibrated discrete structures; 2) exploration of various design alternatives without any biases toward pre-existing typologies; 3) customization of the framework for unique formulations of design problems and a wide range of applications; and 4) intuitive, bidirectional interaction between the form and forces of the structure through reciprocal diagrams. Design tests presented in this paper illustrate the creative potential of the proposed approach, and demonstrate the possibility for unbiased explorations of richer and broader design spaces during early stages of design, with much more trial and less error.

Keywords

Grammar-based design, generative structural design, shape grammars, graphic statics, discrete topology generation, conceptual structural design, structural form finding

Introduction

Most commonly used parametric tools in architectural design provide extensive geometric freedom in absence of structural performance, while engineering analysis software mandates pre-determined forms before it can perform any numerical analysis. Digital models generated by architects typically have to be abstracted and re-modelled by an engineer in a file format that is appropriate for numerical analysis software in order to evaluate static equilibrium. This trial-and-error process is not only time intensive, but it also hinders free exploration beyond standard designs. Furthermore, this setup makes it difficult to explore multiple designs at once. While the rapidly advancing capabilities of computational tools have empowered architects to generate almost any form, and engineers to analyze almost any structure, it has not enabled designers to easily generate and explore new structural forms or typologies. More meaningful investment of the computational resources that are available today may be in investigating new structural possibilities, rather than developing better or faster ways of analyzing what may be inherently bad forms.

There is a need for a computational design methodology that can not only generate forms, but simultaneously process structural information so that the outcome does not need to be constantly remodeled and checked with numerical analysis software. In order to explore a wide range of diverse alternatives during early stages of design, computational power along with controlled randomness can be used to aid the designer in unbiasedly exploring alternative solutions that are unexpected, visually interesting and yet performatively adequate.

Background

Parametric structural design: optimality over diversity

In the conventional parametric modelling paradigm, forms are generated and controlled by parameters or variables. Similarly, in computational design and optimization of structures, the objective function is mathematically formulated and numerical parameters are clearly defined. This means that the design space contains all possible solutions to a given problem, but only includes variations of one particular, pre-determined typology.

Within the parametric design space, sophisticated topology optimization methodologies have been developed to generate solutions that are not only efficient in their performance, but also interesting in its topology. One of the most commonly used approaches in topology optimization is based on the Solid Isotropic Material with Penalization (SIMP) model[1;2], also known as continuum topology optimization. In this approach, elements of resulting designs can have any cross-sectional shape, size and connectivity (Figure 1(b)). The results are significantly unconventional in its shape and topology. However, its practical usefulness in an architectural scale is substantially limited because of challenges regarding fabrication and construction of continuously connected members with varying depths.

[1]Block Research Group, ETH Zurich, Switzerland
[2]Department of Architecture, MIT, USA

Corresponding author:
Juney Lee, ETH Zurich, Institute of Technology in Architecture, Block Research Group, Stefano-Franscini-Platz 5, HIL H 47, 8093 Zurich, Switzerland
Email: juney.lee@arch.ethz.ch

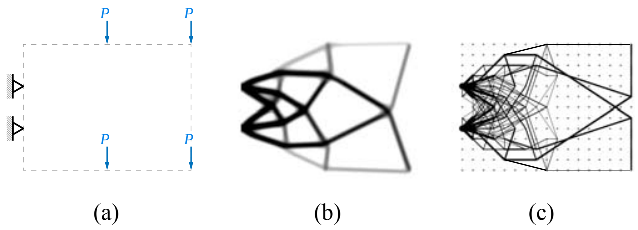


Figure 1. (a) A simple design problem, with solutions generated by: (b) continuum topology optimization; and (c) ground structures method.

Alternative method for continuum topology optimization is the ground structures approach, where the optimization process is conducted using a mass array of discrete members and the cross-sectional areas of members as optimization parameters. Sizing optimization is iteratively executed, and members with smallest sectional areas are gradually eliminated and subsequently guides the structure towards an optimal layout[1–6]. Unlike continuum topology optimization, the final designs using this approach are already discretized but often with hundreds of overlapping members (Figure 1(c)). The ground structure method relies on a large number of design parameters, and the designer often needs to subjectively simplify the initial setup which significantly influences the results.

While powerful in finding a single optimal topology for a given problem, these methods are not capable of rapidly generating a wide range of diverse topologies for comparative design evaluations or explorations. In addition, architectural design objectives are often difficult to integrate into optimization algorithms that are mathematically constructed. Furthermore, resulting designs offer no intuitive platform through which a designer can interactively modify the structure for further design and study. In general, parameter-based topology optimization methods are differently formulated approaches that eventually result in similar optimal designs as demonstrated in Figure 1. During early stages of design, a parametric structural design space does not contain the wide range of design possibilities that the designer may want to consider[7].

Grammar-based design

A grammar-based approach can be used in place of the conventional parameter-based design paradigm in order to broaden the design space. Grammar-based design, also known as shape grammars, uses a set of geometric rules to automate form generations and transformations that are guided by a desired logic, style or objective[8]. It has been used frequently in an architectural context to not only analyze existing design styles and languages, but also to generate new ones. The potential applicability of shape grammars to other fields such as engineering was demonstrated by Mitchell, who incorporated functional attributes and structural criteria to grammar rules in the form of functional grammars[9]. While robust in concept, functional grammars relies on combinatorial variation of standardized structural elements, and can not be used for generating and exploring new structural typologies.

Structural grammars

Application of shape grammars in structural engineering has been made most notably by a method called shape annealing. Shape annealing is a generate-and-evaluate method that combines shape grammars[10] with simulated annealing[11]. Shape annealing iteratively applies a series of geometric grammar rules to transform a structure, and uses a stochastic search algorithm to guide the transformation process to satisfy an optimization criterion[12]. Applied in context of structural engineering, shape annealing can be used to generate various geometric forms that are not only structurally feasible but also address other design objectives[13]. Shea has demonstrated a variety of structural design applications using shape annealing, including generative design of roof trusses[14;15], spatial domes[16] and transmission towers[17;18].

While successful in generating unconventional designs that satisfy practical objectives, shape annealing relies on grammar rules that are entirely geometric without any embedded structural information, therefore a numerical analysis of the entire structure is required after every step of the process. In addition, all transformations are regulated by a stochastic optimization algorithm, which means that unless the selected grammar rule and its subsequent transformation improves the overall performance of the structure, the algorithm will continue to search for on that does. Shape annealing is ultimately an optimization driven method that outputs one solution for each computational run, and it is resource-intensive to generate large quantities of diverse designs at once for comparative analysis.

Alternatively, shape grammars can be used to explore trans-typological structures, by randomly mix-and-matching elements of different structural typologies[7;19]. A wide range of unexpected yet structurally equilibrated solutions can be found using a small kit of pre-defined parts and a simple set of grammar rules. However, such mix-and-match approach is limited by the set of typologies that are pre-defined by the user, and can only address specific building or structural typologies.

Graphic statics

Graphic statics is a graphical method of computing forces and equilibrium for discrete structures under axial loads[20]. It is based on the construction of two reciprocal diagrams[21]: the form diagram that describes the topology and geometry of the structure, and the force diagram that represents equilibrium of that structure through closed polygons constructed from vector representation of forces. Because forces are graphically computed, no further numerical analysis is required to verify the equilibrium of the structure's internal forces.

Combined with modern day computational platforms, interactive applications such as Active Statics[22], eQuilibrium[23], Constraint-based Graphic Statics[24] and Rhino-VAULT[25] have shown how graphic statics can become a powerful design tool by automating the drawing process, and enabling real-time interaction between the reciprocal diagrams.

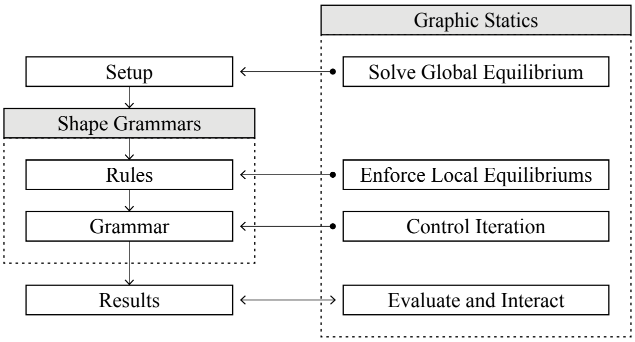


Figure 2. Conceptual overview of integrating shape grammars and graphic statics.

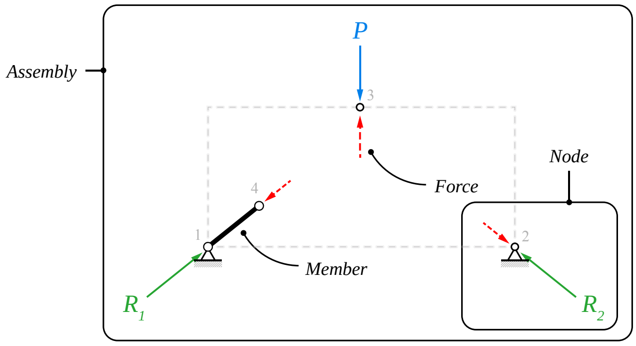


Figure 3. Four basic elements of the methodology

Combining grammars and graphic statics

Shape grammars and graphic statics have been explored previously in the field of structural design, but never in combination. When shape grammars and graphic statics are combined, several key benefits emerge. First, geometric rules can have direct relationship with corresponding force diagrams so that any transformation results in equilibrium. Because local and global equilibriums are always guaranteed, randomness can be introduced during the generation process to explore design diversity. Second, because force diagrams are constructed in parallel with every transformation, there is no need for numerical analysis during or after the generation process to verify equilibrium. In addition, the grammar rules have no boundary-specific parameters, which enables the method to be applied to a wide range of design problems. Lastly, the graphical representation and evaluation of equilibrium offer explicit, bidirectional interaction between the geometry of the structure and its internal forces, which may potentially lead to new insights and better understanding of the design problem. By harnessing the intelligent, generative potency of shape grammars, and the computational graphic statics that can transform forces into equilibrated forms, architecture and structure can be integrated more seamlessly during conceptual design.

Methodology

Conceptual overview

The proposed methodology generates designs by iteratively applying a series of randomly chosen generative rules to a structure that remains in static equilibrium at every step. The conceptual overview of the computational setup is illustrated in Figure 2. Generally, the shape grammars engine is responsible for choosing grammar rules, deciding where to apply them and creating the geometry. Graphic statics ensures equilibrium and functions as visualizer and evaluator for all procedures to be performed by the shape grammar engine. In this paper, several key benefits of computational graphic statics is exploited:

- solving global and local equilibrium
- direct materialization of force vectors into the geometry of the structure using the reciprocal relationship between form and force diagrams
- simplification of the *extended Maxwell rule* using graph interpretation of the form diagram to evaluate stability of the structure (see section: Structural feasibility criteria)
- seamless integration with the approximation of the structure's total load path for performative evaluation (see section: Evaluation metric)
- use of multiple possible configurations of force polygons as randomness generator
- visualization of the structure's internal forces
- bi-directional control of the form and force diagrams for post-generation design modifications and explorations without breaking the structure's equilibrium

Elements

The proposed methodology operates on four types of computational classes: 1) a *Force* class that is a vector, with a type parameter (applied, reaction or temporary), direction value (+1 for compression or -1 for tension) and magnitude; 2) a *Node* class that includes a coordinate, state parameter (active or inactive), type parameter (support, load, float, end, corner or inner), and a list of *Forces* acting on that node; 3) a *Member* class that is a line, with information about its internal force; and 4) an *Assembly* class that includes a list of *Nodes*, list of *Members*, the overall system state (start, go, close or end), and other information about the entire structure. The four class elements are summarized in Figure 3.

In this paper, italicized words will be used to refer to an object class in the programming environment of the proposed methodology. For example: the word "force" simply refers to its literal definition. A *Force* is a custom programming object that is a digital representation of an actual force, with several layers of parameters and information as described in the previous paragraph.

Assumptions

To develop this methodology, the following key assumptions are made:

1. All structures begin with user-defined or automatically computed reaction *Forces* that are in global equilibrium with the applied *Forces*.
2. This paper only considers discretized truss-like structures, and flexural (bending) stresses onto

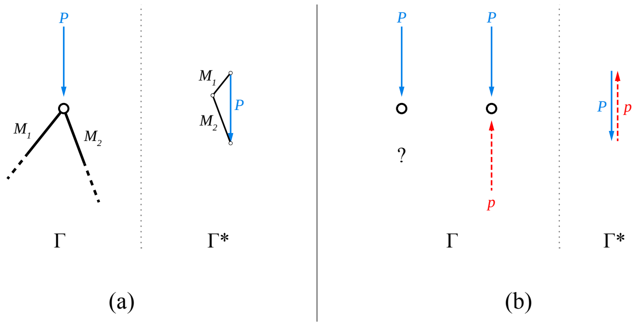


Figure 4. (a) Equilibrium of a typical *Node* with an applied load P and *Members* M_1 and M_2; and (b) the use of a temporary *Force* (shown in red, dotted arrows) to equilibrate a *Node* without any *Members*.

bar elements are not considered. Therefore, all connections are hinges, or pins.

3. Crossing of members is allowed.
4. The buckling capacity of each member and of the structure is not analyzed in this study However, buckling considerations can be incorporated as a global constraint if desired. In order to impose this constraint, the user will need to make preliminary decisions on material, section sizes and shapes of the *Members*. Then, the individual rules can determine the maximum allowed length of a member according the *Force* that it is operating on.
5. All geometric operations are based on local equilibrations of *Forces* of a *Node*.

Concept 1: temporary Forces

For a *Node* of a discrete structure to be in a state of static equilibrium, the sum of the force vectors acting on that *Node* must equal zero. In two-dimensional graphic statics, equilibrium is verified when the force vectors form a closed polygon[20;21;26]. If the geometry of the structure at a *Node* is already known as shown in the form diagram Γ of Figure 4(a), the force polygon construction for that *Node* is simple, and the equilibrium can be easily verified with the corresponding force diagram, Γ*. However, when the structure at a *Node* is not yet known, the *Node* can be equilibrated with a temporary *Force*, shown as a dotted red arrow (Figure 4(b)).

Similarly, if a structure is assumed to be in global static and rotational equilibrium, the sum of the external force vectors (applied loads and reactions) must equal zero by forming a closed global force polygon (Figure 5(a)). This means that regardless of the geometry of the structure that is in static equilibrium, the sum of all internal force vectors (including any temporary *Forces*) also equals zero, as shown in the force diagram Γ* of Figure 5(b).

Concept 2: graphic statics enforced rules

Because temporary *Forces* are initially used to enforce the internal equilibrium, the temporary *Forces* can then be used to generate the geometry of the structures that are subsequently in equilibrium. Specific geometric generations or transformations can be formulated as a rule that operates on the magnitudes and orientations of temporary *Forces*

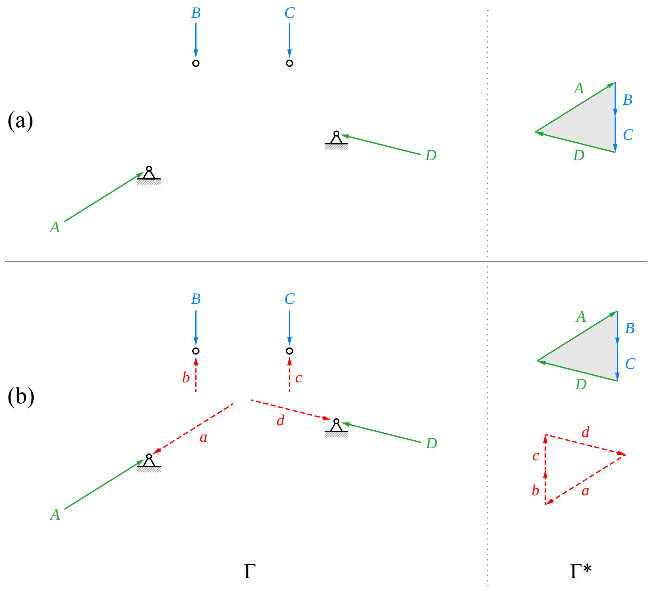


Figure 5. (a) Global equilibrium of external *Forces* (applied loads shown in blue, reactions in green, global force polygons with gray fills); and as a corollary, (b) the equilibrium of internal *Forces* (in this instance, just temporary *Forces* shown as red, dotted arrows).

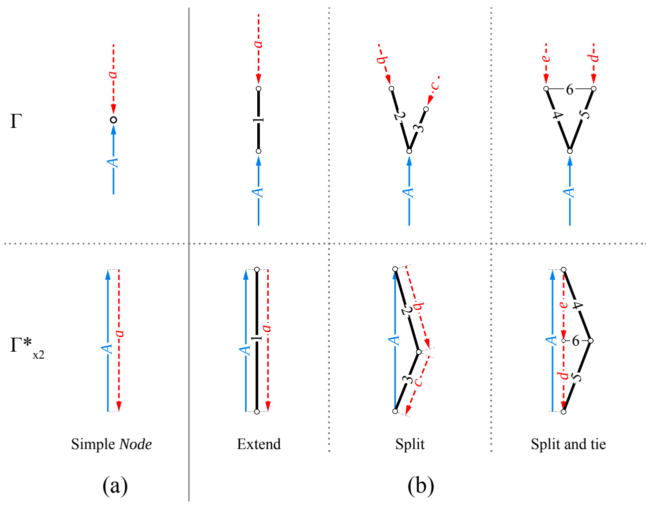


Figure 6. (a) a free *Node* in equilibrium; and (b) three simple rules showing how equilibrated structure can be generated using the temporary *Force*.

at a *Node*. Figure 6 shows three examples of elementary rules that can generate new *Members* and *Nodes* by using the temporary *Force* of a floating *Node*. Force diagrams are scaled by 200% for clarity. Thick black lines represent *Members* that are in compression, and thin black lines as *Members* in tension.

Concept 3: Node types

Six different types of *Nodes* are used in this paper (Figure 7). Any *Node* with a temporary *Force* is in an "Active" state, and can be selected for the application of a rule. In shape grammars, rules have a Left-Hand Side, or LHS (the shape prior to the rule application) and Right-Hand Side, or RHS (the shape after the rule application)[8;10;27;28]. Typically, a shape recognition procedure is necessary to recognize

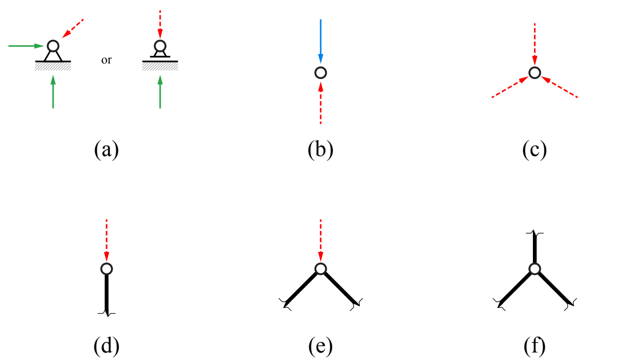


Figure 7. Six different types of *Nodes*: (a) support; (b) load; (c) float; (d) end; (e) corner; and (f) inner

the shapes and evaluate the state of LHS. However, the *Node* type as a class attribute is used as LHS and RHS in the proposed approach, and therefore no additional shape recognition procedure is necessary.

Concept 4: controlled randomness as diversity

This paper introduces controlled randomness to explore topological diversity in three different ways: 1) the sequence of force polygon construction (Figure 8(a)); 2) randomness of rule and the location to apply the rule (Figure 8(b)), and 3) variance of rule parameters (Figure 10). First, given a set number of forces that are in equilibrium, there are multiple ways of constructing a simply-connected structure in equilibrium with all the forces. Figure8(a) shows three possible forms generated from the same set of forces in equilibrium by altering the order of forces during the force polygon construction. In Figure 8(b), same rule is applied at different subset of forces, which result in forms with different topologies.

Constraints

Generative grammars can be a powerful tool in discovering and exploring new structural typologies. However, without intelligent constraints, the rules may be too broad and generate forms that have limited practical feasibility. In addition, the grammar rules can potentially be applied recursively, or repeated uncontrollably without an end. The following strategies are used to control the automated generation.

- Set reasonable ranges of angles for the initial pinned support reactions. Within this range, the initial angles can be either manually defined by the user, or automatically selected.
- Set reasonable local bounds for rule parameters, such as minimum and maximum angles or lengths.
- Set global termination conditions, such as generation count and recursion control mechanisms.

Framework

Workflow

Unlike most conventional engineering tools, this methodology begins without a starting geometry. First, the user sets up the problem by defining the magnitudes and locations of applied loads, and solving the reaction forces by completing the global force diagram (Figure 9(a) and 5(a)). The user then chooses the rules to apply, and weights for each rule, which defines how likely it is for that rule to be randomly selected to be applied (Figure 9(b)). Finally, the user defines how many options to produce (Figure 9(c)). Results can be further diversified by modifying the following global parameters: 1) minimum number of rule applications for each generation; 2) rule sensitivity towards the beginning, the middle or the end of the generation cycle; 3) termination conditions; and 4) random seed.

Rules

The eight rules used for generating designs in this paper are summarized in Figure 10. Geometric rules incorporate structural logic and information, and equilibrium is always verified by constructing the force diagrams. While the precise values of the parameters are randomly determined, it is constrained by user-defined lower and upper bounds.

Grammar (automatic generation algorithm)

Figure 11 summarizes the grammar portion of the computational framework: the automatic random generation algorithm. Steps 1 and 2a are where the user sets up the design problem. Then, the initial *Assembly* is constructed in step 2b. Steps 3a through 3e, where the algorithm randomly chooses a *Node* to apply a random rule, are repeated until the system reaches a terminating condition defined by the user. Once the terminating condition is reached, the structure then enters the finalizing phase, where Rule 6 is applied until no temporary *Forces* remain.

Example generation

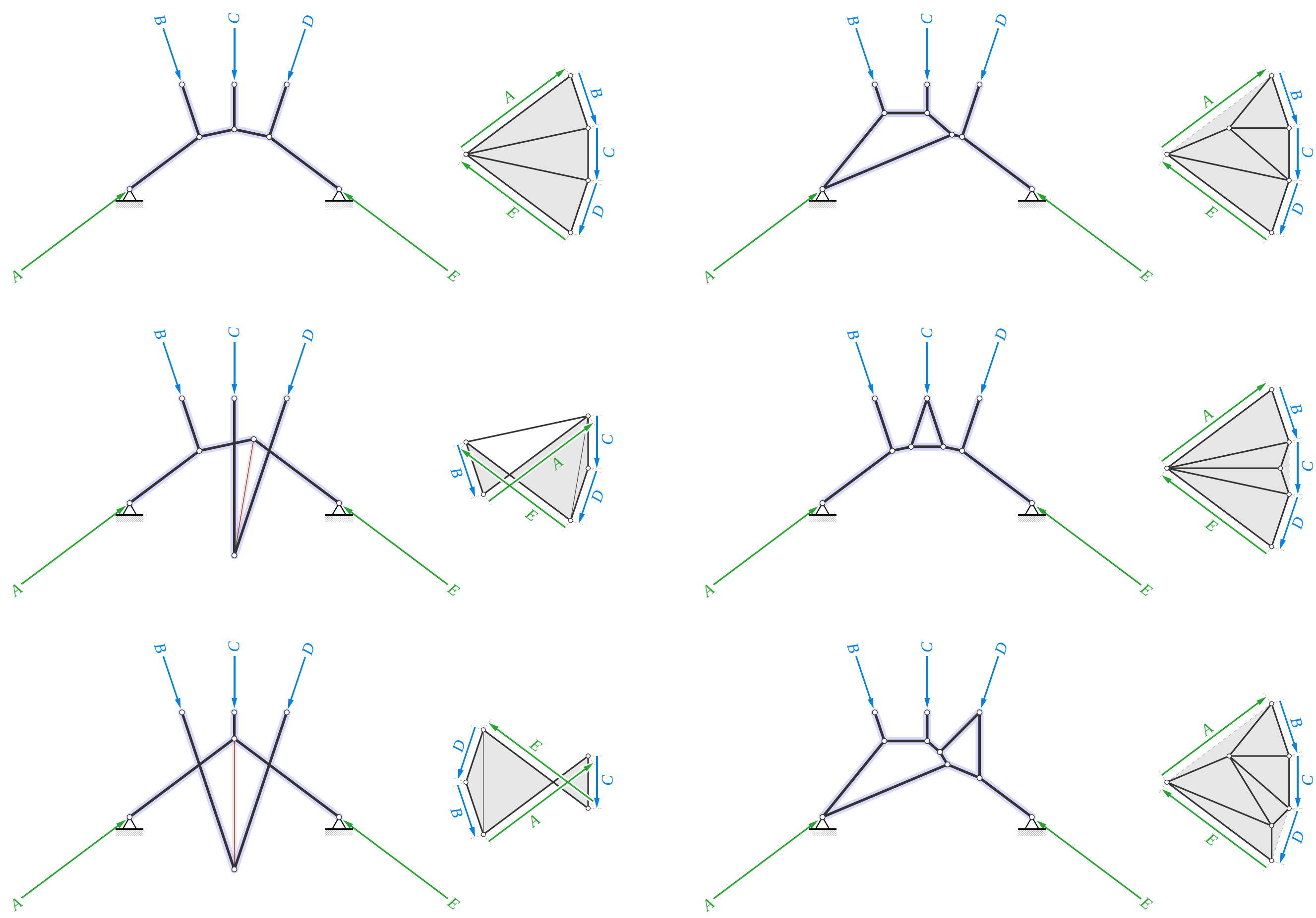
Figure 12 shows an example problem, and the full generation sequence. Step-by-step explanation of the generation is described here.

Results

In this section, the proposed methodology is tested on several design scenarios to demonstrate how this approach can be used to generate a wide range of diverse discrete structures for various problems.

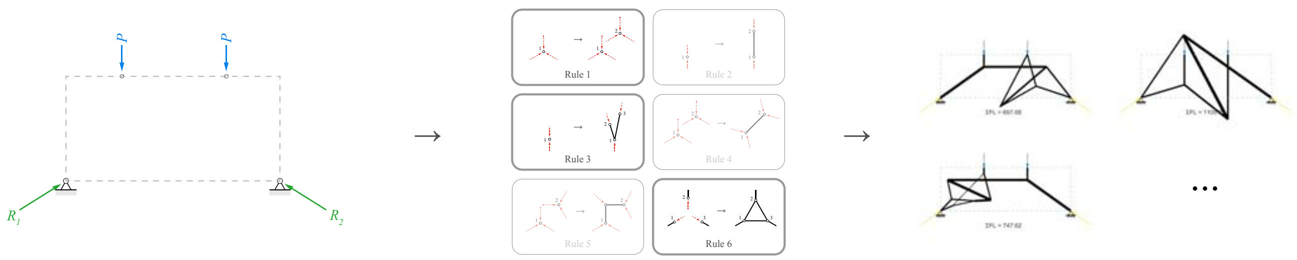
Implementation

The proposed methodology was implemented using Rhinoceros[29], Grasshopper[30] and Python[31] scripting language. Within the Rhinoceros interface, each iteration instantaneously generates: 1) corresponding force diagrams for every *Node*, 2) a complete force diagram for the entire structure, if possible (to be discussed in detail in section: Real-time interaction with results), 3) a form diagram with clear labels, and 4) rule history, information and evaluation metrics for the shown design. Visual representation of the forces, the evaluation metric, and the rule history which summarizes how the structure was derived, enable clearer understanding of the structure and informs better design decisions more quickly. The rule history, which records all the parameters that were used to generate the current iteration, is an important feature that enables reproducibility



(a) (b)

Figure 8. (a) Altering the order in which a set of forces are assembled to construct the equilibrium force polygon, drastically changes the resulting form (from top, down): A-B-C-D-E, B-A-C-D-E, A-C-E-D-B; and (b) location at which rule is applied changes the form diagram (from top, down): split rule applied to A only, split rule applied to C only, split rule applied to A and D.



(a) (b) (c)

Figure 9. The workflow of the proposed methodology.

of the same iteration during later stages in design, when more information about the boundary conditions and the project in general, may be available.

Structural feasibility criteria

In order to assess the general structural feasibility of the designs generated using the presented method, its stability needs to be checked. By using the *extended Maxwell rule*, various states of a structure's equilibrium with different degrees of static and kinematic (in)determinacy can be evaluated[32;33]:

$$k - m = b - 2n + r, \qquad (1)$$

with k the number of independent states of self-stress, m the number of inextensible mechanisms, b the number of bars, n the number of nodes, and r the number of kinematic restraints at the supports. Using graphic statics, equation (1) can be simplified and used as a convenient way to check for the stability of designs. As in *Algebraic Graph Statics*[34], the form diagram in graphic statics can be interpreted as a graph (Figure 13). Since all external forces (k in equation (1) as

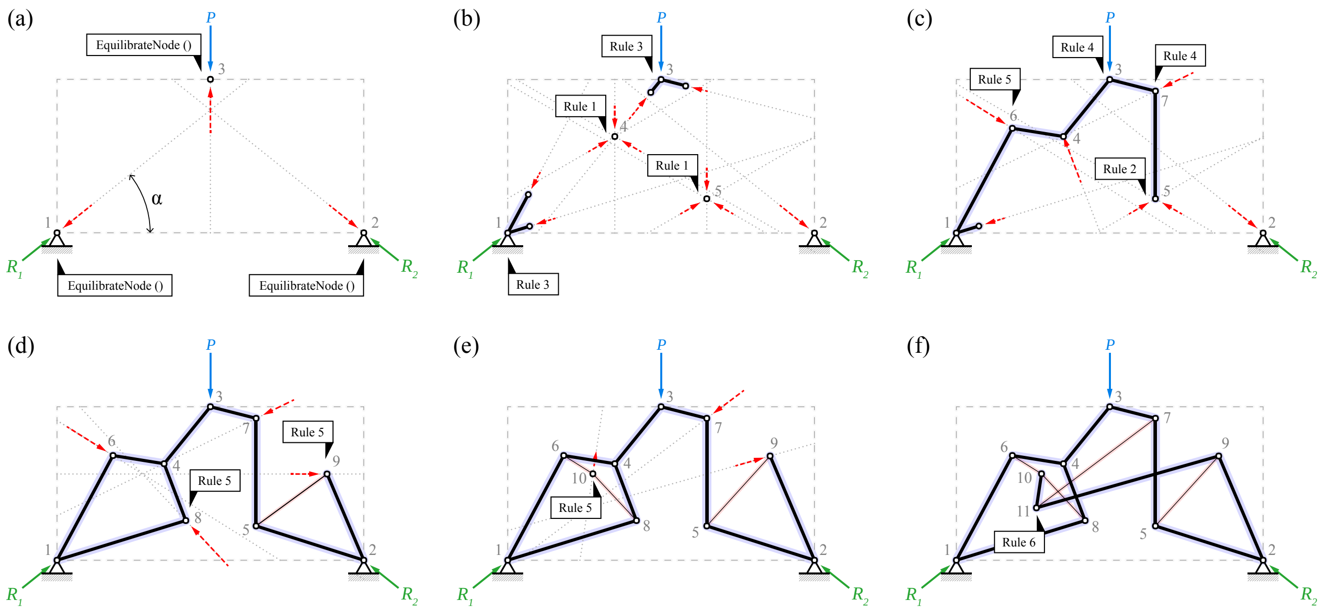


Figure 12. An example of an automatic random generation sequence.

applied forces, r in equation (1) as support restraints) are represented in this graph interpretation of the form diagram, the equation (1) can be simplified to[34]:

$$k - m = e - 2v_i, \tag{2}$$

with e the number of edges or members and v_i the number of internal vertices in the form graph. Values of m and k can be calculated by using the equilibrium matrix of the structure[35]. In general, if $k = 0$, no states of self-stress exists and the structure is unstable. During the design generation process, penalty scores can be enforced to sort or eliminate such designs.

All designs shown are in static equilibrium with the specified load cases. However, there are *Nodes* and *Members* that would be unstable under different loading conditions. Zero-force *Members*, which are shown as dashed lines in this paper, are added in order to prevent these potential instabilities by acting as a temporary tensile or compressive *Members*. In the designs presented, these instabilities generally occur at *Nodes* where applied loads are present. Zero-force *Members* are automatically added at these *Nodes*, connecting them to either the next closest *Node* with an applied load, or a support *Node*.

Evaluation metric

While unbiased generation and exploration of diverse designs is the key objective of the proposed approach, the designs still need an evaluation metric through which they can be objectively compared from a structural performance point of view. The performance metric used in this paper is based on an approximation of the total volume of structural material, or equivalently the total load path[36]. Assuming constant internal stress at its optimal or final iteration state, the total volume or load path can be calculated as follows in terms of the locations of the nodes in the force diagram, x[37]:

$$\min_x V = \min_x \frac{1}{\sigma} \sum |P_i| \cdot L_i \tag{3}$$

where V represents the total load path or volume of the structure, σ is a constant that represents the allowable stress, P_i is the internal force and L_i is the length of the ith member, respectively. In reciprocal form and force diagrams used in graphic statics, member i with length L_i in the form diagram has a corresponding line in the force diagram of length $\hat{L}_i$ that is proportional to the force P_i in the member. Using form and force diagrams in graphic statics, equation 3 can be rewritten as follows[38]:

$$\min_x V = \min_x \frac{1}{\sigma} \sum \hat{L}_i \cdot L_i \tag{4}$$

Using graphic statics, the total load path of the structure can be computed easily by multiplying the length of the member in the form diagram, and the length of the corresponding force vector which is provided by the force diagrams.

2D results

Figure 14 and Figure 15 show the application of the methodology in six different planar design scenarios. For each scenario, the simplest solution that can be derived using the fewest number of *Members* is shown on top left, which will be used as the benchmark for comparison. The upper row for each scenario shows the top four performing solutions out of 40 iterations. All designs have performances that are approximately within 20% of the benchmark solution, which is given a normalized score of 1.00. The normalized scores for these top four performing solutions are shown in blue.

For each design scenario, four additional designs chosen by the authors are also shown. These designs are presented to showcase significant amount of diversity that can be explored, which may often be desirable even at the sacrifice of a small amount of efficiency. For reference and comparative evaluation, the normalized score for these four additional solutions are also provided in magenta. For clarity, *Members* in compression are highlighted in blue, and tension in red. All designs presented are generated assuming fully stressed design.

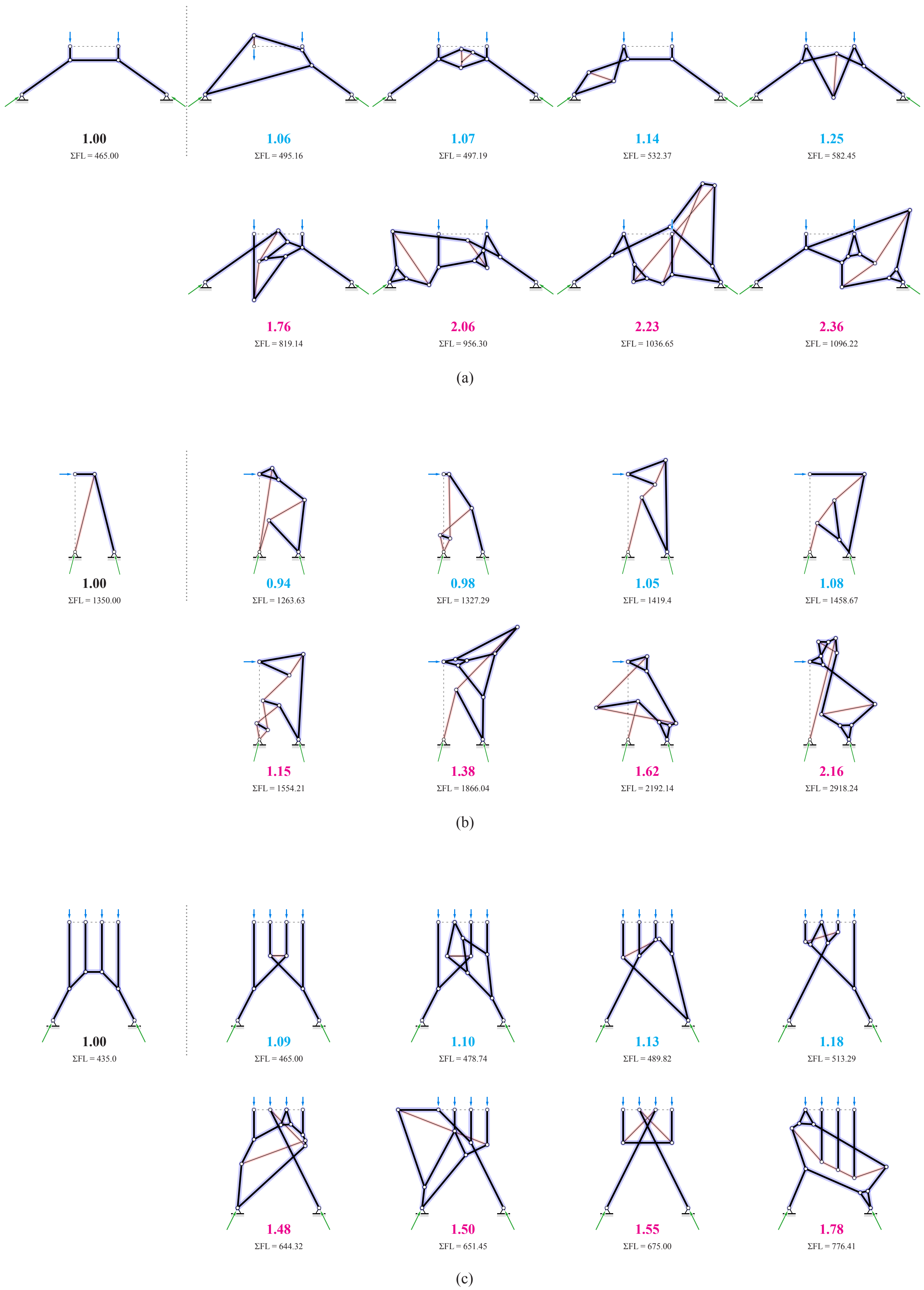


Figure 14. Application of the methodology on various design scenarios: (a) span-like structure; (b) vertical cantilever structure; and (c) vertical, wall-like structure.

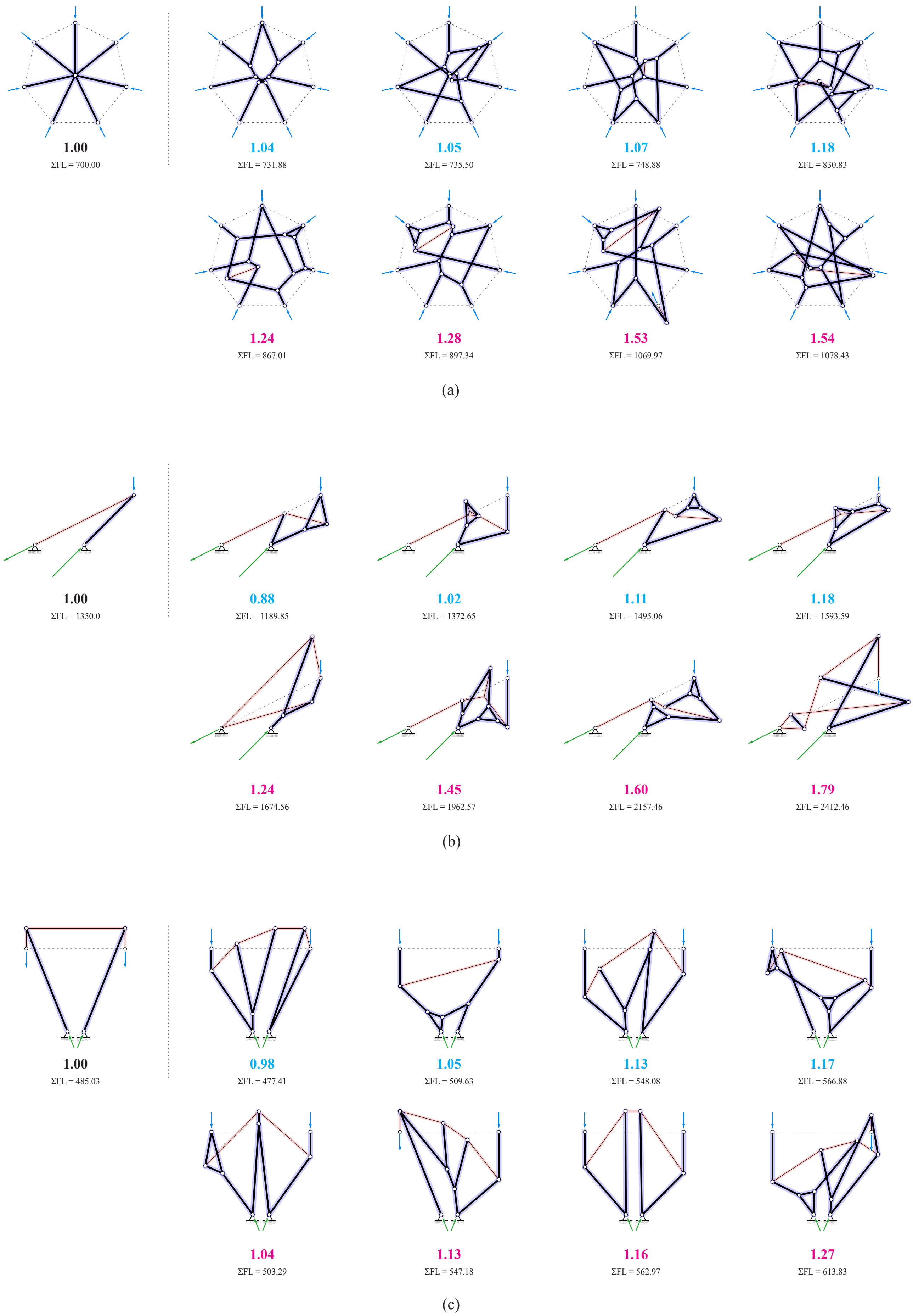


Figure 15. Application of the methodology on various design scenarios: (d) radial, compression structure; (e) horizontal cantilever; and (f) horizontal cantilever in two directions

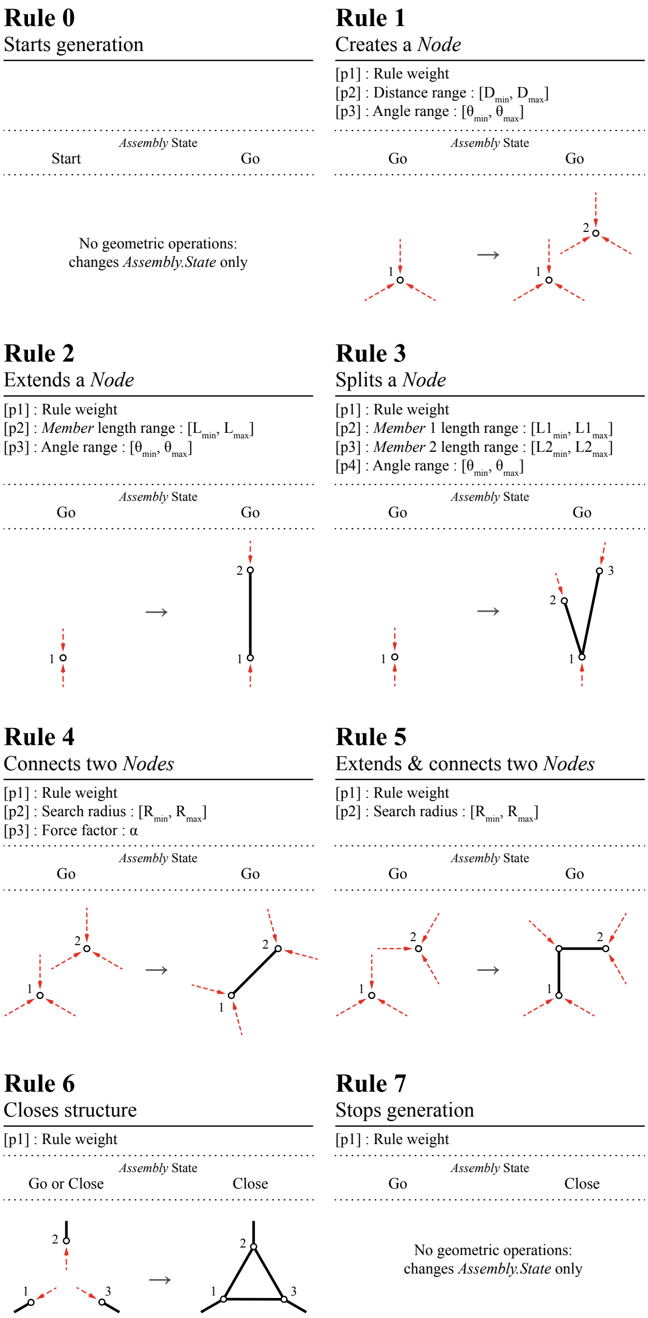


Figure 10. Summary of rules and parameters.

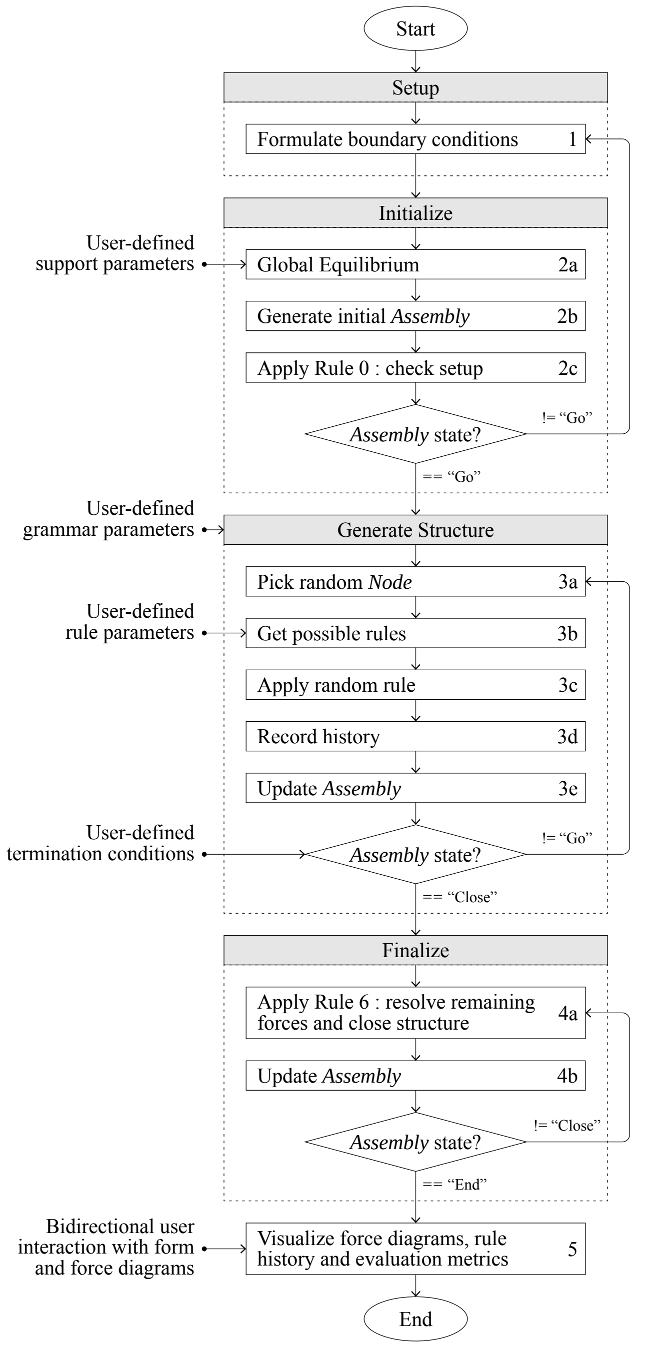


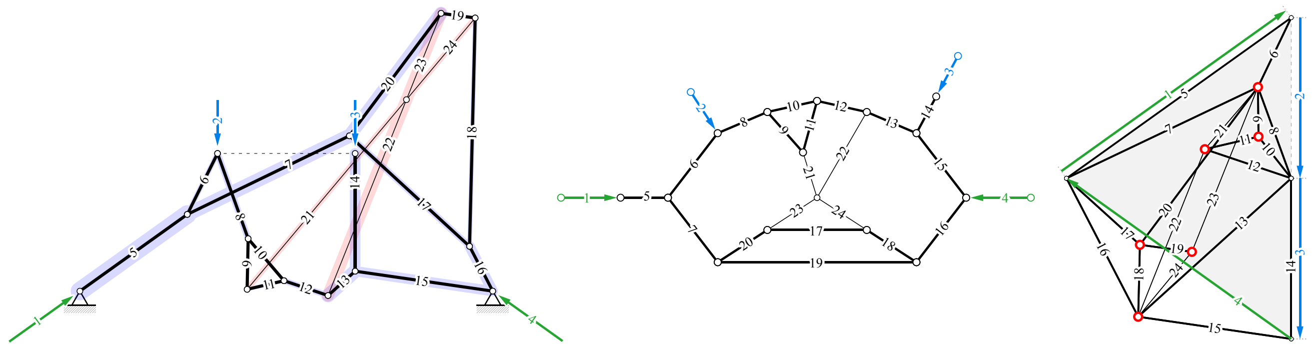
Figure 11. Framework of the automatic random generation algorithm.

Real-time interaction with results

In addition to a wide range of topological diversity, the proposed methodology also provides the user with bidirectional control between the form and force diagrams, and enable direct manipulation of the designs. This interactivity is possible only if the following condition is satisfied. Any form diagram Γ can be interpreted as an abstracted mathematical graph, which is a network of vertices and edges. In traditional graphic statics, a complete force diagram for a form diagram can be constructed only if the form diagram can be drawn as a planar graph[34]. A graph is planar, if it can be redrawn in the plane without any crossing edges[39].

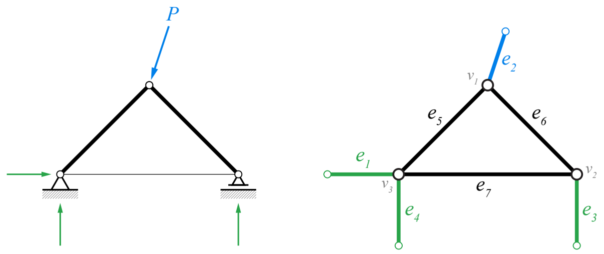
The example design in Figure 16 has crossing edges in its initial form diagram Γ. However, Γ can be redrawn as a planar graph $P(\Gamma)$, while maintaining the connectivity of Γ. By using the internal *Member* force magnitudes and directions provided by Γ, and the new clockwise ordering of *Members* around each *Nodes* in $P(\Gamma)$, a complete force diagram Γ^* can be constructed. The red vertices in Γ^* can be moved around to modify the topology of the structure in real time as the user desires, without breaking the structure's equilibrium. For a thorough technical overview of constructing complete force diagrams from form diagrams with crossing edges, please refer to Van Mele and Block[34].

Construction of the complete force diagram is essential for both interaction and evaluation, because it enables the user to freely modify the form or the force diagram to explore design refinements while constantly enforcing equilibrium. In addition, the complete force diagram offers



Γ P(Γ) Γ*

Figure 16. The form diagram Γ, of a solution to the span-like structure problem in Figure 14(a). If Γ is planar, Γ can be redrawn as a planarized graph P(Γ) such that none of the edges are crossing. Using traditional graphic static conventions for constructing reciprocal force diagrams, Γ and P(Γ) can be used to construct the complete force diagram for the entire structure, Γ^*



(a) (b)

Figure 13. (a) A simple structure with one applied load P, one pinned support on the left and one roller support on the right; and (b) graph interpretation of the structure, with seven edges ($e = 7$) and three internal vertices ($v_i = 3$). This particular structure is statically determinate and stable, with $m = 0$ and $k = 1$.

a snap shot view of the internal forces of the structure, and more specifically their relative magnitudes. This visual feedback greatly improves the user's creative intuition, overall understanding of the solutions and the behavioral consequences of the changes being made.

Exploration of rule parameters

Several global and local parameters can be modified to explore design alternatives, as well as trade-offs between various constraints. Parameters described in this section relate specifically to the span-like structure scenario in Figure 14(a). In each of the following three parameter explorations, the performance scores are normalized to the first design shown on the left.

Global parameter 1: reaction angles Because the support reaction vectors are determined before the automatic random generation begins, a variety of possible solutions with varying shapes and performances can be explored by setting a reasonable bound for this reaction angle parameter for pinned supports, as illustrated in Figure 17. Conversely, the reaction angles can be altered after a design has been chosen to improve the performance. This parameter is most closely related to the boundary conditions of the supports, and how much horizontal reaction a support can withstand.

Global parameter 2: generation count Generation count defines the maximum number of rules that can be applied in a generation. Figure 18 shows three similar structures with varying generation counts. The generation count can be increased if more members with reduced magnitudes of internal stresses may be required, or more geometric variation and expression within a design are desired.

Rule parameters Figure 19 shows the effect that rule parameter variations can have on the results. The lower and upper bounds for the angle range of Rule 3 was changed for each generation. While increasing the range of possible angles does not necessarily improve the performance, the larger angles may be necessary for constructability of joints. Similarly, modifying the parameters for other rules will result in drastically diverse designs.

Practical applications

While the proposed methodology allows exploration of diverse design possibilities during conceptual design, the results will need to be interpreted and refined by the architect and the engineer in order to develop the design with more detail and rigor during later stages in design. In order to develop designs that are topologically unique and interesting, and yet somewhat regular for practical and constructability considerations, global constraints such as symmetry can be enforced prior to the design generation. Figure 20 demonstrates how a simple span problem can be decomposed into smaller problems to enforce global symmetry in the designs to be generated. Using global symmetry constraints, Figure 21 illustrates how three designs selected by the authors can be developed into realistic, yet significantly different and unexpected roof structures.

Extending rules into 3D

The presented methodology can also be used to generate free-form spatial structures in 3D, with slight modifications to Rule 5. All other rules work in both 2D and 3D. Rule 5 is an important rule that closes the structure by incrementally consolidating temporary *Forces*. In 2D, Rule 5 performs this operations by finding the intersection of the lines of action (LOA) of two temporary *Forces* of two different *Nodes*. This intersection is where a new *Node* is created,

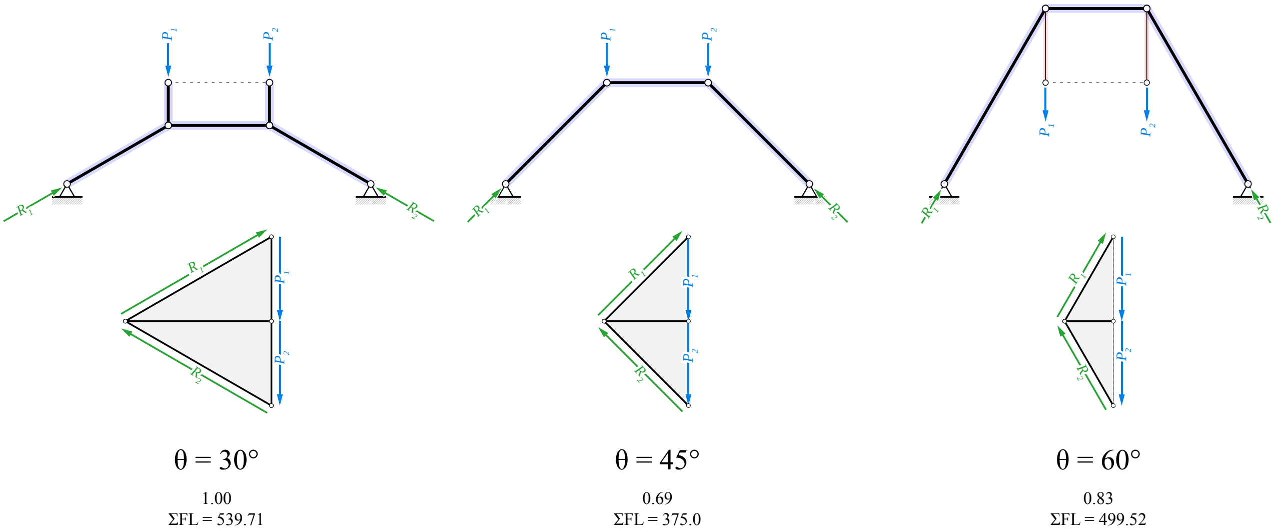


Figure 17. Changing the angle of the reactions not only changes the shape of the structure, but also its performance

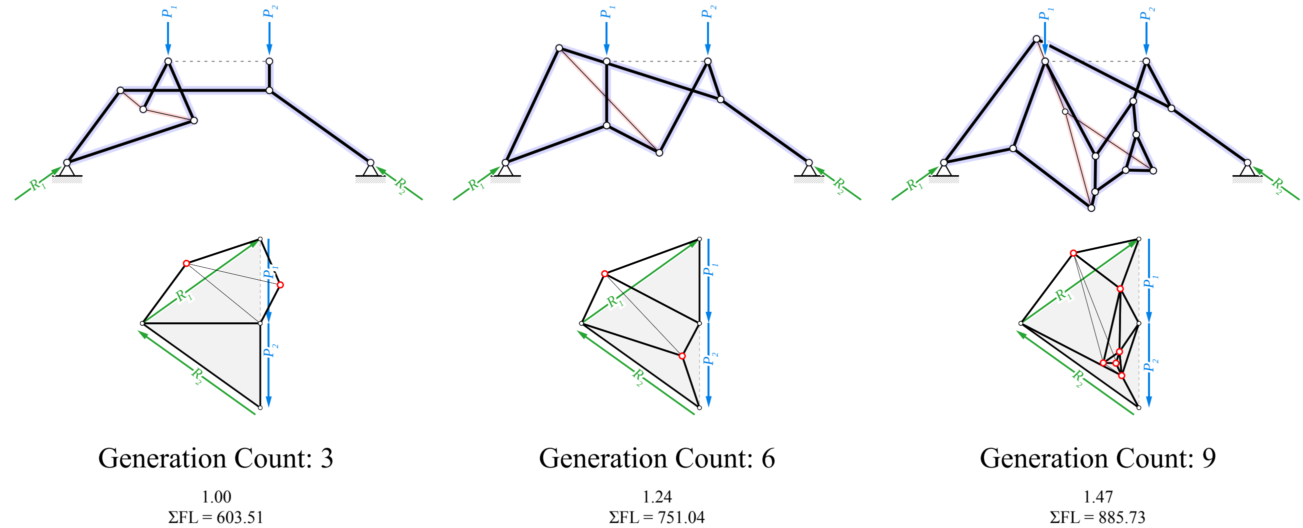


Figure 18. Changing the generation count controls the number of rules to be applied for each generation

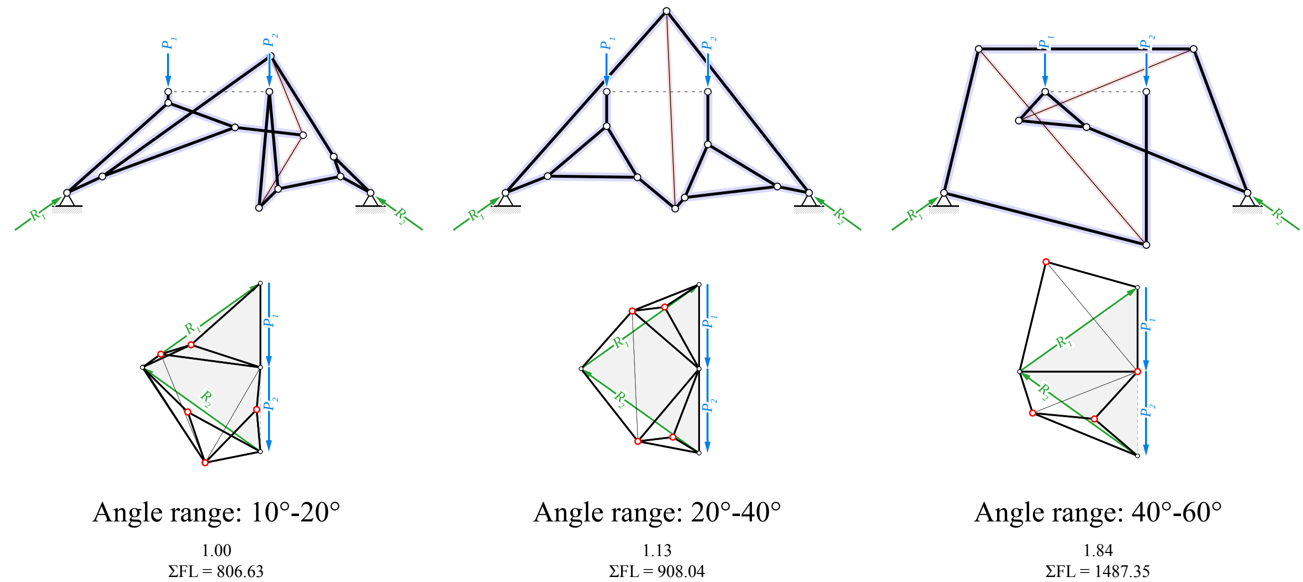


Figure 19. Effect of changing rule parameters on the results.

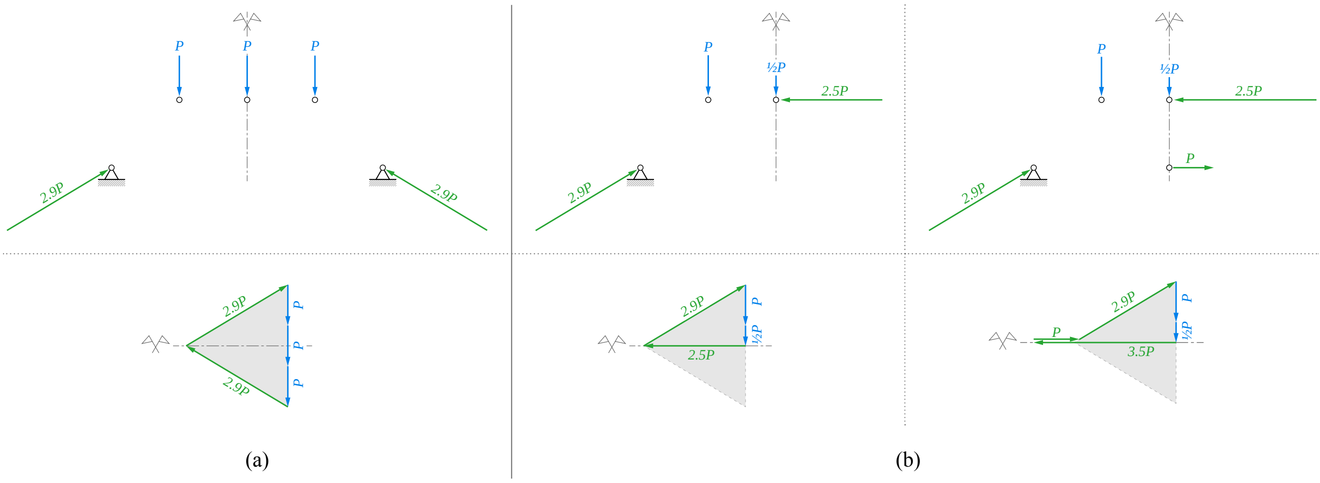


Figure 20. Effect of changing rule parameters on the results.

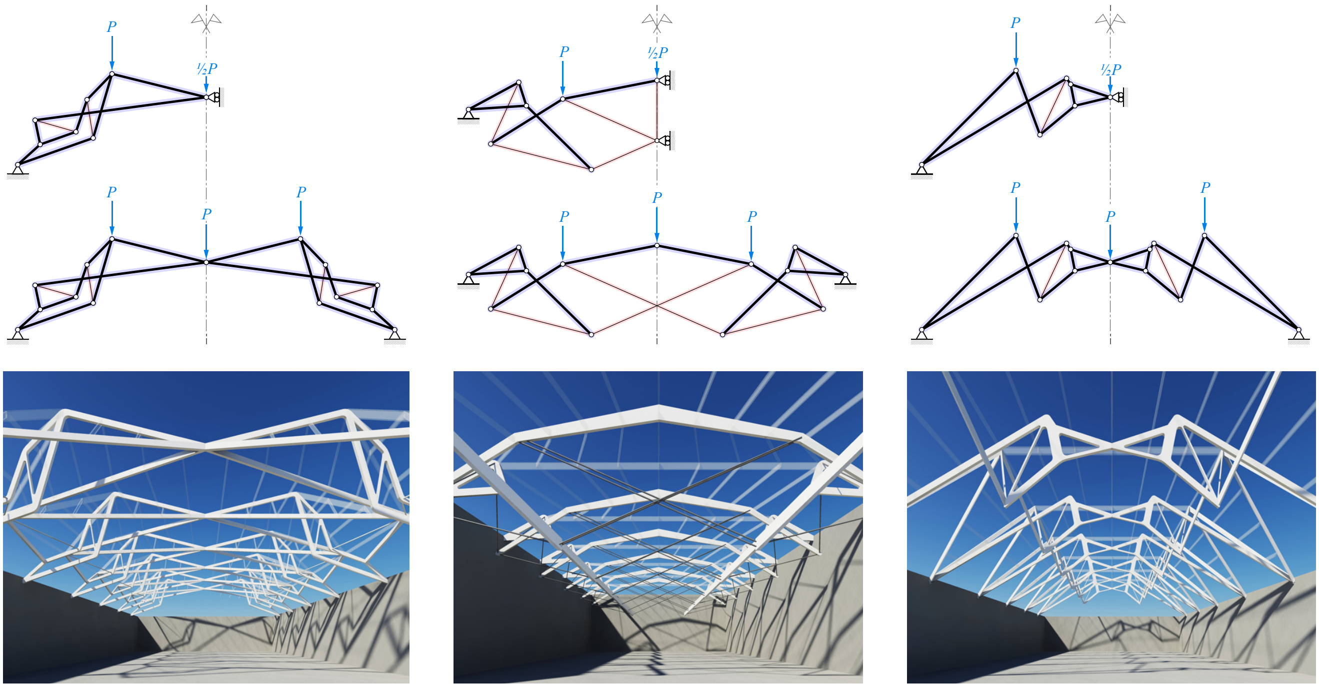


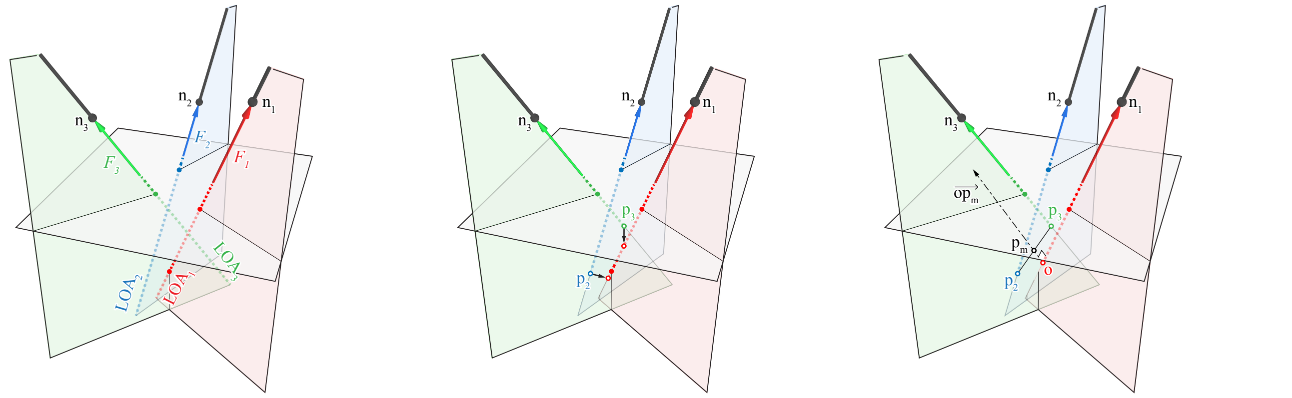
Figure 21. Sample designs with symmetry enforced, resulting in more practically applicable designs [40;41].

and new *Members* are created by connecting this new *Node* with the initial two *Nodes*. As a result, two temporary *Forces* are consolidated into one temporary *Force*. In 2D, the intersection of LOAs can be easily computed since all vectors and geometries are coplanar. However, LOAs in 3D are likely to be non-coplanar, in which case there is no intersection point and hence no equilibrium.

Figure 22 illustrates an extension of Rule 5 in 3D, where three temporary *Forces* can be reduced to two temporary *Forces* by using a temporary plane A that is coplanar with LOA of the selected temporary *Force*. This plane ensures that every set of three *Forces* acting on a single *Node* lie in the same plane, which is a preliminary condition for equilibrium of a *Node*. While there are infinite number of planes that could be used to complete this spatial operation, this strategy ensures that the resulting *Members* are within reasonable lengths. This strategy is applicable to all cases except when the three or more temporary *Forces* are all parallel orientation, which may be solved trivially by equilibrating the three *Forces* with any two *Members* and two *Forces*.

Design example in 3D

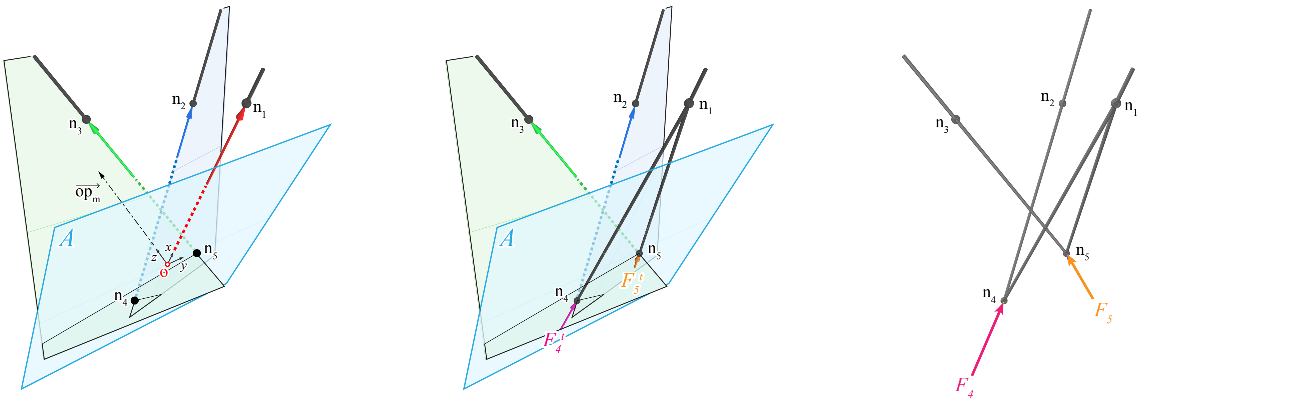
Using the 3D extension of Rule 5, free-form spatial structures in equilibrium can be generated as shown in Figure 23. The spatial configuration of the design is highly complex and intricate, which would be extremely difficult to create manually by either sketching or drawing. Automated creation of three-dimensional equilibrium has tremendous design potential in discovering new possible structural forms for complex spatial problems that cannot easily be projected and visualized onto a 2D plane, as in most traditional graphic statics applications.



1) Three non-coplanar *Members*, and their corresponding temporary *Forces* F_1, F_2 and F_3.

2) Choose random temporary *Force* (F_1 in this example). On LOA$_2$ and LOA$_3$, find the closest points to LOA$_1$ (p_2 and p_3, respectively).

3) Find midpoint (p_m) between p_2 and p_3. The point that is closest to p_m on LOA$_1$, is o.

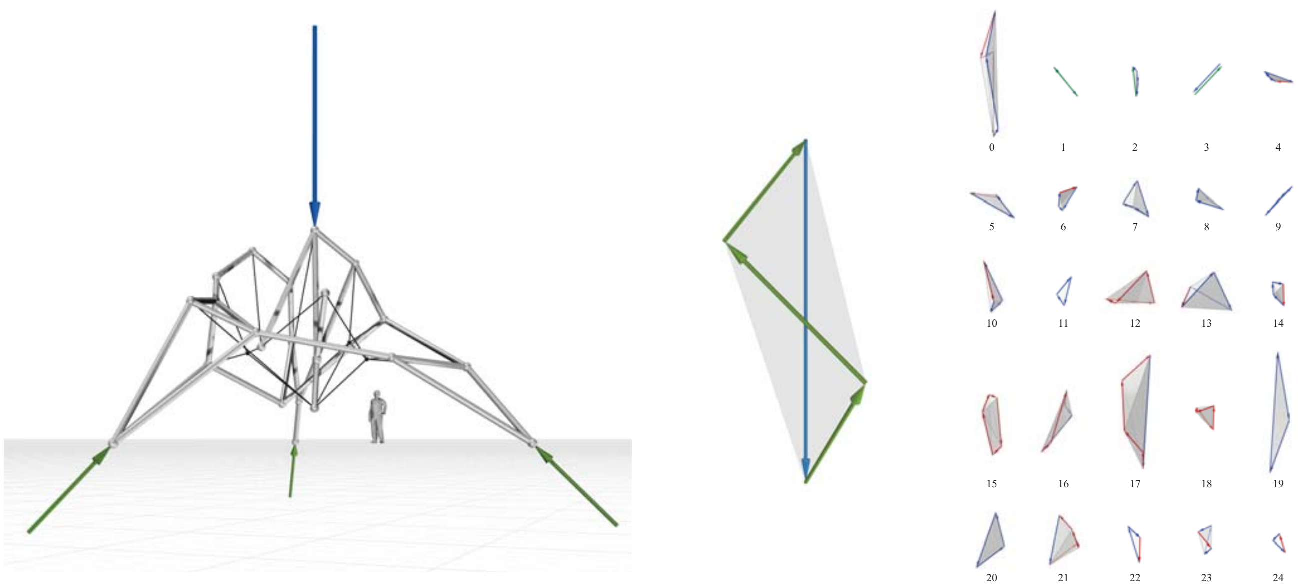


4) Using o as the origin, $\overrightarrow{op_m}$ as the z-axis, and LOA$_1$ as the x-axis, create plane *A*. Intersection of LOA$_2$ and LOA$_3$ with plane *A* creates *Nodes* n_4 and n_5, respectively.

5) By using Rule 4, two new *Members* are created to connect n_1 with n_4 and n_5. *Nodes* n_4 and n_5 are temporarily equilibrated with temporary *Forces* F_4' and F_5'.

6) By connecting n_2 with n_4, and n_3 with n_5, additional two *Members* are created, and only two temporary *Forces* (F_4 and F_5) remain.

Figure 22. Extension of Rule 5 in 3D.



(a) (b) (c)

Figure 23. A 3D example of a free-form spatial structure generated using the methodology: (a) the form; (b) 3D diagram of global equilibrium; and (c) 3D diagram of local equilibrium for each *Node*.

Conclusions

Contributions

This paper introduced a grammar-based design methodology as an alternative to the conventional parametric design paradigm, which is limited in topological diversity and often leads to expected solutions. The following specific contributions were presented.

Structural design with more trial and less error By incorporating forces during the form generation process, the resulting designs are guaranteed to be in equilibrium for the specified loading conditions. Therefore, no further numerical analysis is required to check equilibrium. Reduced coordination time between architects and engineers allows exploration of better and more interesting ideas faster and more efficiently. While most numerical analysis tools provide quick feedback on performance, they do not inform the designer with any guidance for improving future designs. On the other hand, graphic statics instantly generates clear visualization of forces, which results in clearer understanding of the structure's internal forces. As a result, the user's intuition of the relationship between form and forces is improved, and better decisions will be made more quickly as the project progresses.

Unbiased exploration of new design spaces With automated generation by the computer which is guided by the design goals input by the human designer, diverse solutions can be generated that simply would not be conceivable manually by a human designer with a pencil or a mouse. The purpose of this approach is not to replace the human designer, but rather use the computerized automation to enable the designer to spend more time exploring new forms than analyzing one in detail. In addition, the automated generation of multiple topologies at once not only increases the creative capacity of the designer, but also leads to new insights and better understanding of the design problem itself.

Beyond reciprocity: generative graphic statics The reciprocal relationship between form and forces in graphic statics generally means that one has to be created before the other can be drawn. Therefore, traditional computational graphic statics tools only work with pre-set problems and functions mostly as an interactive analysis or visualization tool. By combining graphic statics with shape grammars, the form-finding capabilities of graphic statics can be used to generate equilibrium structures. Most previous work done on shape grammars require a shape to preexist before any rule can be applied. However, the rules presented in this paper are based on *Nodes* and are not dependent on any preceding shapes or geometries. Therefore, the methodology is flexible enough to be applied to a variety of design problems, and is able to generate structures without any prescribed topologies or preferences.

Future work

There are several important directions for future work. Firstly, global parameters could be improved to gain a better control of the overall generation process, including more intelligent ways in which the rules are chosen and where they are applied. Secondly, post-processing of generated structures could be performed by applying additional rules that locally deconstruct the model and then rebuild it. Also, more detailed or material-specific constraints, buckling constraints, minimizing overlapping members and self weight could be incorporated in future extensions of this research. Furthermore, because all designs shown in this paper are also statically equilibrated only for the defined load case, it will be important to develop a procedure to control possible mechanisms and local instabilities, especially for spatial structures. Lastly, while this paper focused on rules based on the form diagram, rules can also be developed for the force diagram[42], which will further enrich the structural design possibilities using graphic statics.

Closing remark

Overall, this new methodology demonstrates the validity in combining and applying shape grammars and graphic statics together to various engineering design problems. The general versatility and customizability of the presented approach, and the speed at which it can generate mass quantities of unconventional and yet statically equilibrated structures, greatly improves possibilities for creative yet performance-focused explorations during early stages of conceptual structural design.

References

1. Rozvany GIN. *Topology Optimization in Structural Mechanics*. Vienna, Austria: Springer, 1997.
2. Bendse MP and Sigmund O. *Topology optimization: theory, methods and applications*. 2nd ed. Berlin: Springer, 2004.
3. Dorn WS, Gomory RE and Greenberg HJ. Automatic design of optimal structures. *Journal de Mecanique* 1964; 3: 25–52.
4. Ben-Tal A and Bendse MP. A new method for optimal truss topology design. *SIAM Journal on Optimization* 1993; 3(2): 322–358. URL http://dblp.uni-trier.de/db/journals/siamjo/siamjo3.html#Ben-TalB93.
5. Achtziger W. *Topology Optimization in Structural Mechanics*, chapter Topology Optimization of Discrete Structures. Vienna: Springer Vienna. ISBN 978-3-7091-2566-3, 1997. pp. 57–100. DOI:10.1007/978-3-7091-2566-3_2. URL http://dx.doi.org/10.1007/978-3-7091-2566-3_2.
6. Christensen P and Klarbring A. *An Introduction to Structural Optimization*. Solid Mechanics and Its Applications, Springer Netherlands, 2008. ISBN 9781402086663. URL https://books.google.ch/books?id=80IeN__MYI8C.
7. Mueller C. *Computational Exploration of the Structural Design Space*. PhD Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2014.
8. Stiny G and Gips J. Shape grammars and generative specification of painting and sculpture. In Freiman CV (ed.) *Proceedings of Information Processing*, volume 71. Amsterdam, The Netherlands, pp. 1460–1465.
9. Mitchell W. Functional grammars: An introduction. In *Proceedings of ACADIA Conference 1991*. Los Angeles, USA, pp. 167–176.
10. Stiny G. Introduction to shape and shape grammars. *Environment and Planning B: Planning and Design* 1980; 7(3): 343–351. URL http://EconPapers.repec.org/RePEc:pio:envirb:v:7:y:1980:i:3:p:343-351.

11. Kirkpatrick S, Gelatt CD and Vecchi MP. Optimization by simulated annealing. *SCIENCE* 1983; 220(4598): 671–680.
12. Cagan J and Mitchell W. Optimally directed shape generation by shape annealing. *Environment and Planning B: Planning and Design* 1993; 20: 5–12.
13. Reddy G and Cagan J. An improved shape annealing algorithm for truss topology generation. *Journal of Mechanical Design* 1995; 117(2A): 315–321.
14. Shea K, Cagan J and Fenves SJ. A shape annealing approach to optimal truss design with dynamic grouping of members. *ASME Journal of Mechanical Design* 1997; 119(3): 388–394.
15. Shea K and Cagan J. The design of novel roof trusses with shape annealing: assessing the ability of a computational method in aiding structural designers with varying design intent. *Design Studies* 1999; 20(1): 3 – 23. DOI:http://dx.doi.org/10.1016/S0142-694X(98)00019-2. URL http://www.sciencedirect.com/science/article/pii/S0142694X98000192.
16. Shea K and Cagan J. Innovative dome design: Applying geodesic patterns with shape annealing. *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing* 1997; 11: 379–394. DOI:10.1017/S0890060400003310. URL http://journals.cambridge.org/article_S0890060400003310.
17. Shea K. *Essays of discrete structures: Purposeful design of grammatical structures by directed stochastic search*. PhD Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1997.
18. Shea K and Cagan J. Languages and semantics of grammatical discrete structures. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 1964; 13: 241–251.
19. Geyer P. Multidisciplinary grammars supporting design optimization of buildings. *Research in Engineering Design* 2008; 18: 197–216.
20. Culmann K. *Die Graphische Statik*. Zrich: Verlag Meyer und Zeller, 1864.
21. Cremona L. *Graphical Statics: Two Treatises on the Graphical Calculus and Reciprocal Figures in Graphical Statics. Translated by Thomas Hudson Beare*. Oxford: Clarendon Press, 1890.
22. Greenwold S and Allen E. Active Statics. website, 2003. URL http://acg.media.mit.edu/people/simong/statics.
23. Group BR. eQUILIBRIUM. website, 2009-. URL http://block.arch.ethz.ch/equilibrium.
24. Fivet C and Zastavni D. Constraint based graphic statics: New paradigm of computer-aided structural equilibrium design. *Journal of the International Association of Shell and Spatial Structures* 2013; 54: 271–280.
25. Rippmann M, Lachauer L and Block P. Interactive vault design. *International Journal of Space Structures* 2012; 27(4): 219–230. DOI:10.1260/0266-3511.27.4.219.
26. Allen E and Zalewski W. *Form and Forces: Designing Efficient and Expressive Structures*. New York: Wiley, 2009.
27. Stiny G. *Shape: Talking About Seeing and Doing*. The MIT Press, 2008. ISBN 0262693674, 9780262693677.
28. Mitchell WJ. *The Logic of Architecture: Design, Computation, and Cognition*. 1st ed. Cambridge, MA, USA: MIT Press, 1990. ISBN 0262631164.
29. Mcneel R. Rhinoceros: Nurbs modeling for windows. website, 2015. URL http://www.rhino3d.com.
30. Rutten D. Grasshopper 3d. website, 2015. URL http://www.grasshopper3d.com.
31. Python. website, 1990. URL http://www.python.org.
32. Maxwell JC. On the calculation of the equilibrium and stiffness of frames. *Philosophical Magazine Series 4* 1864; 27(182): 294–299.
33. Calladine CR. Buckminster Fuller's 'Tensegrity' Structures and Clerk Maxwell's Rules for the Construction of Stiff Frames. *International Journal of Solids and Structures* 1978; 14: 161–172.
34. Van Mele T and Block P. Algebraic graph statics. *Computer-Aided Design* 2014; 53: 104–116. DOI:10.1016/j.cad.2014.04.004.
35. Pellegrino S. Structural computations with the singular value decomposition of the equilibrium matrix. *International Journal of Solids and Structures* 1993; 30(21): 3025–3035.
36. Baker W, Beghini L, Arkadiusz M et al. Maxwells reciprocal diagrams and discrete michell frames. *Structural and Multidisciplinary Optimization* 2013; 48(2): 267–271.
37. Stromberg LL, Beghini A, Baker WF et al. Application of layout and topology optimization using pattern gradation for the conceptual design of buildings. *Struct Multidiscip Optim* 2011; 43(2): 165–180. DOI:10.1007/s00158-010-0563-1. URL http://dx.doi.org/10.1007/s00158-010-0563-1.
38. Beghini LL, Carrion J, Beghini A et al. Structural optimization using graphic statics. *Struct Multidiscip Optim* 2014; 49(3): 351–366. DOI:10.1007/s00158-013-1002-x. URL http://dx.doi.org/10.1007/s00158-013-1002-x.
39. Harary F. Planarity. In *Graph Theory*, chapter 11. Reading, MA: Addison Wesley, 1994. pp. 102–125.
40. Lee J. Grammatical design with graphic statics: Rule-based generation of diverse equilibrium structures, 2015.
41. Lee J, Fivet C and Mueller C. Modelling with forces: Grammar-based graphic statics for diverse architectural structures. In Ramsgaard Thomsen M, Tamke M, Gengnagel C et al. (eds.) *Proceedings of the Design Modelling Symposium 2015*. Copenhagen: Springer International Publishing, pp. 491–504. DOI:10.1007/978-3-319-24208-8_41.
42. Akbarzadeh M, Van Mele T and Block P. Compression-only form finding through finite subdivision of the external force polygon. In *Proceedings of the IASS-SLTE 2014 Symposium*. Brasilia, Brazil.