Proceedings of the IASS Annual Symposium 2020/21 and
the 7th International Conference on Spatial Structures
Inspiring the Next Generation
23 – 27 August 2021, Guilford, UK
S.A. Behnejad, G.A.R. Parke and O.A. Samavati (eds.)

Grammar-based generation of bar networks in static equilibrium with bounded bar lengths

Ioannis MIRTSOPOULOS*[a], Corentin FIVET[a]

*[a] SXL, Structural Xploration Lab, EPFL Fribourg, Switzerland
* ioannis.mirtsopoulos@epfl.ch

Abstract

This paper presents a generative approach for the conceptual design of structures, which aims at exploring alternative bar networks in static equilibrium by means of a grammar rule. Operating on incomplete networks of bars in interim static equilibrium, the incremental rule application imposes the iterative introduction of: a new node, some bar elements and a few interim forces, based on the desired type of transformation. The node introduction is only possible within a feasible domain that allows both the rule application and the bar elements' constructability. In this paper, the grammar rule not only considers static equilibrium, but also designer-defined bar length constraints. These supplementary constraints are transformed into geometric domains and respectively inform the feasible domain. The length constraints neither shrink the exploratory power of rule-based design nor the size of the design space, but rather allow for structures materialization at a later stage. The generation of planar networks is used as a case study of the methodology.

Keywords: conceptual design, grammar rules, rule-based design, spatial structures, generative design form-finding

1. Introduction

Design is an "ill-structured" (Simon [1]) and "wicked problem" that "one cannot first understand, then solve" (Rittel *et al.* [2]). Hence, designers may only acquire missing knowledge along the way, by means of an iterative, creative process known as *design exploration*. At the same time, as indicated by (Paulson [3]), late design decision making is highly costly for the project and thus forces the exploration process to happen during the early stages of the project time. Also, design solutions tend to lack diversity and creativity when limited exploration drives designers to premature design fixation (Purcell *et al.* [4]). The invasion of parametric workflows in the field of architects and engineers has revolutionized the generation of architectural forms. Even though they ease the generation of new design variants, they limit the exploration to a subset of the available design space, as they usually focus on numerical alterations within strictly predefined domains, i.e. lower and upper bounds. On top of that, explorations of structural forms often disregard static equilibrium considerations despite those being a necessary feature of any solution. In lack of appropriate computational methods, the design of architectural and structurally relevant forms typically consists in the adaptation of well-known and catalogued forms, or in searches for a seemingly optimum solution of a carefully circumscribed problem. Numerous structural forms not resulting from these routines are yet to be explored.

The above described constraints, limitations and lacks, frame the objectives of the developed design framework by the authors (Mirtsopoulos *et al.* [5][6]); a computational method for the search of original 3-dimensional bar networks in static equilibrium with given loads. Specifically, this framework:

- Aims to be used for the generation of *infinite design variants*, assisting designers to escape cases of design fixation.
- Operates *rule-based* incremental transformations, instead of computing numerical variables to construct forms (*variable-based* design). In the latter case, the relationship among the construction variables is static and subsequently they reproduce the same, identical result.
- Embeds *structural awareness* into the syntax of the exploited rule transformations and no post processing is required to bring the network in static equilibrium.
- Applies equilibrium-informed transformative rules incrementally, in a semi or fully automated way and allows *backtracking* on previous states, when the desired qualitative criteria are not met (e.g. aesthetics), a feature that amplifies the method's exploration potential.
- Generates *networks of bars in compression and tension* that are indicative, but not deemed optimized, force flows to be used as *inspiration* for the design of reticulated structures.

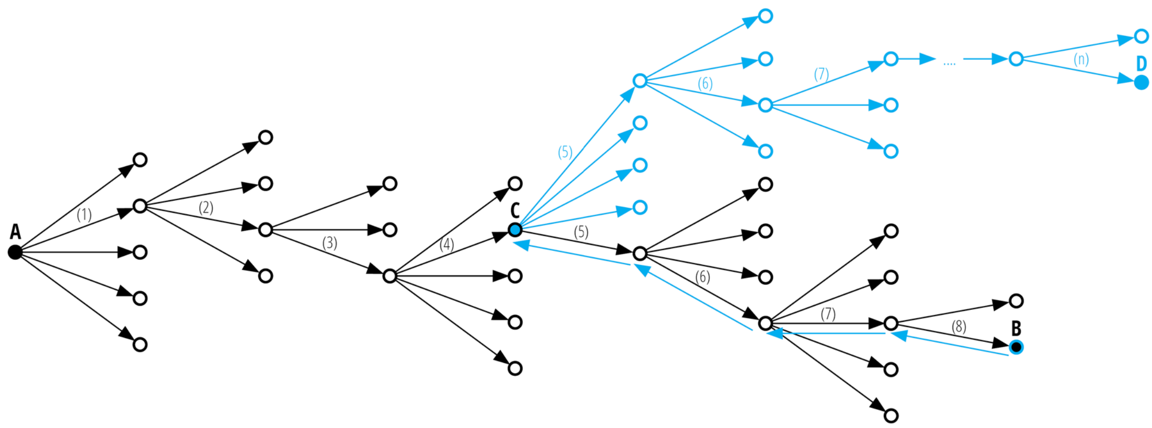

Figure 1: Design exploration; out of a vast range of options, designers explore only a tiny fraction of the design space before choosing a unique design. Starting from state A, the design exploration lasted until state B, when the designer decided to backtrack to state C, where from resumed towards a different branch until the completion of the design at state D.

In this paper, extended functionalities of the proposed design methodology are demonstrated. More precisely, the designer obtains control of the minimum and maximum length of the used struts and ties. This addition, slightly constrains the size of the design space but incorporates realistic conditions and limitations (i.e. material mechanical capacity).

2. Precedent research in the field of explorative structural design

For the scope of this research, *explorative structural design* goes beyond mathematically determined design solutions that satisfy specific objectives and always result in the same output; likely to be the globally optimum solution. Instead, the term is interpreted by the authors as a *structurally-informed form* that does not necessarily resemble an existing structural typology, and that is meant to unveil new ones. The use of heuristics successfully satisfies the latter requirement, as search algorithms are perfect for ill-structured problems (Simon [1]), act blindly and are characterized by the absence of global optimality (Goldberg [7]). As such, the following research projects are further investigated and are used as inspiration.

Search algorithms have been applied to structures whose topology is predefined and the evolutionary exploration mechanism which controls the nodal coordinates imposes the nodes relocation until specific criteria are satisfied (von Buelow [8]; von Buelow [9]; Martini [10]; Mueller [11]). Among the features that amplify the approaches' strength, the storage of design solutions to a database, as well as the interactive parents' selection during the crossover operation, are underlined. Ohlbrock [12] has developed a bottom-up structural design approach that generates structures in static equilibrium with the help of a topological graph which is a manipulative design input. Desired topological configurations are also optimized for specific objectives.

In lack of interactive search computational tools that could explore the design space without committing to explicit goals, Harding [13] developed *Biomorpher* which allows to interactively explore, optimize – with the help of genetic algorithms - and display multiple design variants simultaneously within Grasshopper, the parametric environment of Rhinoceros McNeel.

Alternative approaches for the exploration of the vast design space come from the field of shape grammars. Their inception as design means dates back in the 70s when Stiny and Gips [14] introduced the term inspired by Noam Chomsky's theories. Since then, multiple projects have been developed in the fields of urban and architectural design, but not only. The variable configuration and rearrangement of closed shapes or volumes has eased the exploration of urban patterns [15], room configurations in architectural drawing plans [16][17], building volumes [18] and triangulated truss structures [19][20].

Inspired by the concept of replicating rules for the synthesis of larger entities, like a limited number of letters synthetizes words, sentences and eventually formulate a communication language, Mueller [21] applied grammar rules to generate diverse structural systems, i.e. bridges. More recent implementations have coupled vector-based graphic statics with grammar rules [22] and have ultimately proven their suitability to generate structurally-aware forms during the conceptual early design stage.

3. Method

In the scope of this research project, conceptual structural design stands for the process leading to schematic early-stage options of bar networks in static equilibrium, or, in other words of synthetic representations of a structure's static equilibrium. Chosen options are expected to materialize into effective structural forms in subsequent design steps, through comprehensive structural analyses. The design space exploration (DSE) navigates an unbounded territory of design candidates, all satisfying set constraints. The hypothesis is that a DSE unveils unprecedented structural typologies, fights design fixation and provokes creativity.

The proposed design approach adopts the concept of rule-based design to explore the design space. It exploits a universal, parametric grammar rule, which builds on vector-based graphic statics. Hence the rule application embeds structural awareness by ensuring the network's static equilibrium. The iterative process transits from a disconnected network of forces in interim static equilibrium, to a connected network of bars and forces in static equilibrium. Each grammar rule application imposes a transformation that consists in: the introduction of a new node (P); the replacement of interim forces by bars in compression or tension; and, if necessary, the introduction of new interim forces to retain static equilibrium. By definition, the process filters out design variants that are not in static equilibrium. Additional constraints can be set through the applications of the rule. This paper considers a constraint that limits bar lengths within a bounded domain.

In the developed design pipeline, the DSE is initiated with a set of forces (applied loads and support reactions) applied onto a bounded design space (D_{space}). Together with their nodes (anchor points) of application, they form a disconnected network of forces. At this stage, the disconnected network is imposed into interim equilibrium with the introduction of equal and opposite interim forces applied at the same anchor points. The goal of the rule application is the incremental elimination of the interim forces through their replacement with bars. The process stops once the pool of interim forces is empty.

3.1. Rule features

A transformative rule is defined by three features: the *entropy rate*, the *force candidates* and the *location of the new node P*.

3.1.1. Entropy rate

The entropy of a network is here a measure of the number of interim forces in the network. The application of a rule to the network increases, decreases, or maintains its entropy, hence it defines a rate at which the entropy of the network evolves. Overall, the transition from a disconnected network of interim forces to a connected network of bars leads to a reduction of entropy. A reduction of entropy therefore correlates to the *convergence* of the DSE. Respectively, an increase of entropy correlates to the *divergence* of the DSE. The preservation of entropy leads to the *stagnation* of the DSE. Temporary divergence or stagnation occur either due to explicit design intentions by the designer or due to an impossibility of ensuring convergence when applying the transformative rule. In the latter case, the aimed transformation is deemed unfeasible and is replaced by a less-constraining transformation of decreasing entropy rate, as per the following sequence: *convergence* → *stagnation* → *divergence*.

Unsuccessful rule applications of high entropy rate (i.e. convergence), increase the number of introduced bar elements as a total and the number of required transformative rule applications until the network gets connected. Overall, the desired entropy rate has great impact as a design feature and explicitly transforms the network's topology on demand.

3.1.2. Force candidates

Every application of the rule operates on forces selected from the pool of all interim forces currently in place. Selected forces are called *force candidates* and their set is a *monome,* a *binome*, or a *trinome*. The network can never decrease its entropy if the rule is applied to an *monome*. All entropy rates are possible, under conditions, when the rule is applied to a *binome* or a *trinome* [see 3.1.3. Location of the new node (P)]. The selection of these force candidates follows the designer's preferences and is materialized via *ranking policies* [see 3.3. Rule application].

3.1.3. Location of the new node (P)

Choosing the entropy rate and selecting force candidates is not sufficient to fully characterize a rule transformation. Each rule application imposes the introduction of a new node and its location is dependent on the above-mentioned features. Not all locations guarantee the desired change of entropy rate or retain the network's static equilibrium. This requirement constrains the placement of P within strictly defined geometric domains, called *entropy rate domains* (D_{er}), which range from a unique location (point), when convergence is attempted, to a segment, a closed polygon, or a volume, when divergence is attempted [Figure 3]. Regardless of the force candidate, P is adjacent to one, two or three bars and completes the generation of a connected sub-network that is aggregated to the main network. The necessity to connect the involved nodes with continuous bars imposes a new constraint related to the location of P. Precisely, all the introduced bars must be fully circumscribed within the design space. This requirement may prevent the rule application in cases of design spaces with non-convexities and/or voids. In those cases, additional geometric domains are considered called constructability domains ($D_{construct}$) [Figure 3]. The intersection of these two domains defines the *feasibility domain* (D_f). The rule is safely applied, hence static equilibrium is retained, for any location of P within the feasibility domain. The location of the new node follows the designer's preferences and is materialized via *input policies* [see 3.3. Rule application].

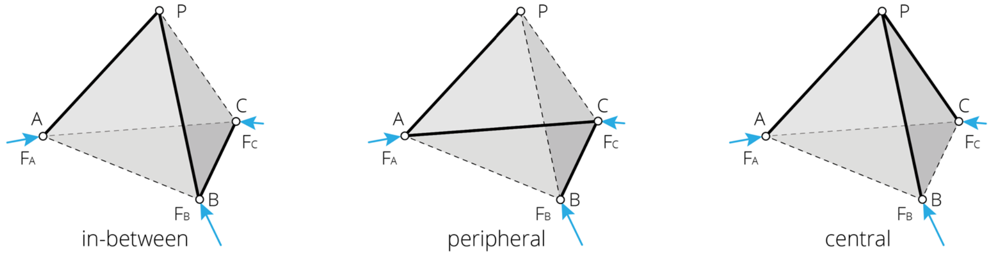

Figure 2: Different connectivity patterns in the case of a trinome.

The force candidate's anchor nodes and *P* may have different topological relationship, i.e. connectivity. Three feasible connectivity patterns exist: *in-between, peripheral* and *central*, named after the topological position of *P* with regards to the anchor nodes. The former pattern is observed when *P* is connected simultaneously with 2 nodes strictly [Figure 2 (left)]. This connectivity occurs when *P* is located "in-between" other nodes, in a topological sequence. As such, this pattern only concerns *binomes* or *trinomes*. The latter pattern is observed when *P* is connected simultaneously with 3 nodes strictly [Figure 2 (right)] and occurs for *trinomes* only. *Peripheral* occurs when *P* is connected with 1 node only [Figure 2 (middle)], which occurs for all three sets of force candidates, i.e. *monomes*, *binomes*, *trinomes*. For *monomes*, all three connectivity concepts represent the same bar connectivity. Similarly, for *binomes*, both the *in-between* and the *central* pattern represent the same bar connectivity.

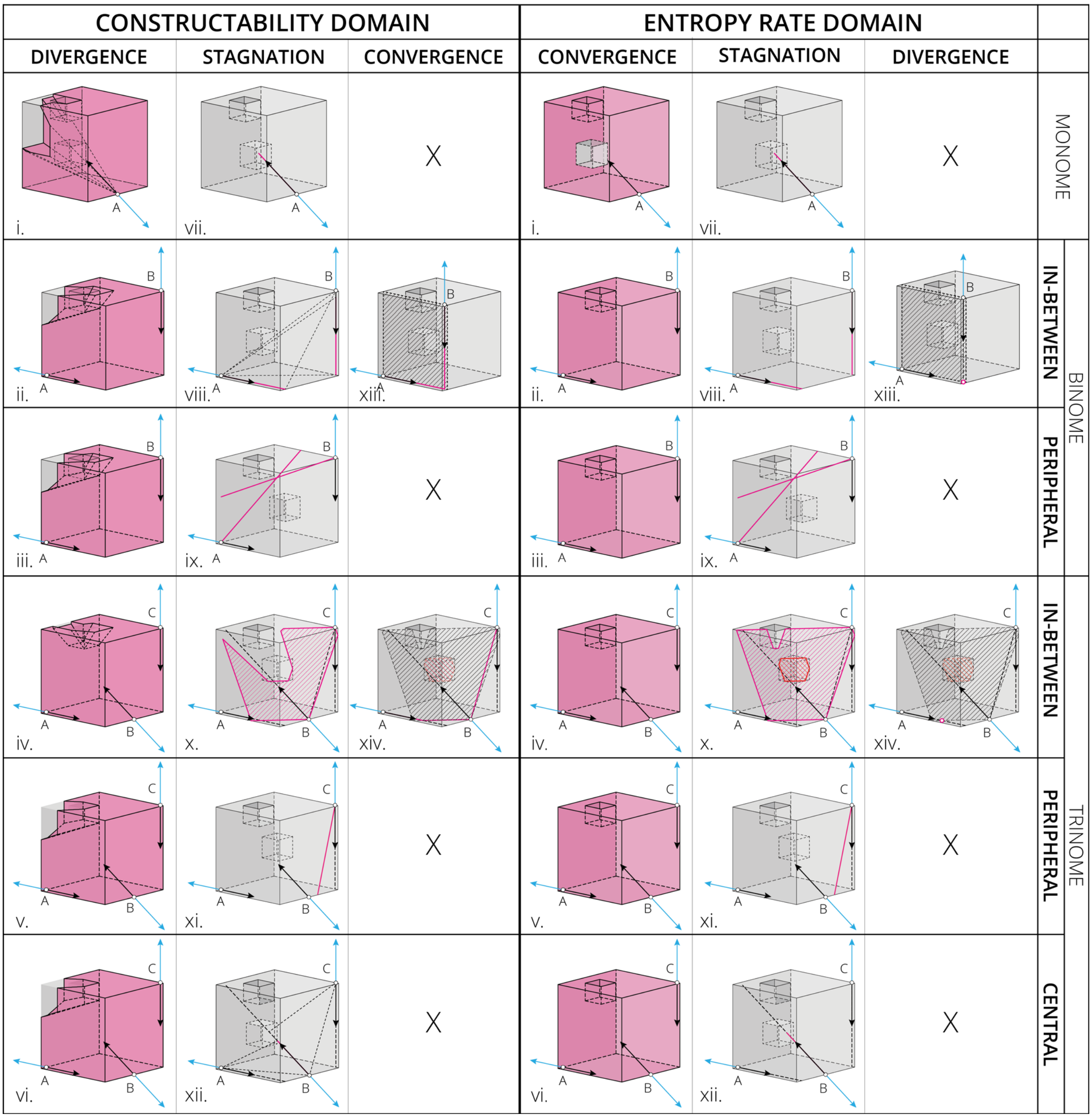

Figure 3: Constructability and entropy rate (geometric) domains of the node P for various number of force candidates (black arrows) and various entropy rates. Each cyan arrow is resultant of inner bar forces or externally applied forces in static equilibrium with each force candidate. The grey cube is the primitive design space. The placement of P within these domains (simultaneously; in constructability and entropy rate domain) guarantees the static equilibrium retainment.

These connectivity patterns explicitly influence the entropy rate domains as well as the constructability constraints. Namely, for the same change of entropy rate and the same force candidates the feasibility domain is different if the connectivity pattern differs too [Figure 3].

3.2. Constrained domain

As part of the design decisions, or due to material properties, the bars' length might need to be constrained. The developed design approach allows the designer to define the minimum and maximum allowed lengths of the introduced bars. More precisely, the distance *d* between *P* and the existing anchor nodes of the sub-network should be constrained within a predefined domain (d $\in$ [minL, maxL]). In this case, the location of *P* is further constrained by the necessity to be included into the constrained domain ($D_{constrained}$), which considers all those *P* locations for which the introduced bars, as imposed by the desired connectivity pattern, are fully circumscribed within the design space. Hence, the feasibility domain is defined differently: $D_f = D_{er} \cap D_{construct} \cap D_{constrained}$. Geometrically speaking, the constrained domain consists in the intersection of two rings (when D_{space} is planar) or hollowed spheres (when D_{space} is spatial), with inner and outer radiuses equal to *minL* and *maxL* respectively [Figure 4].

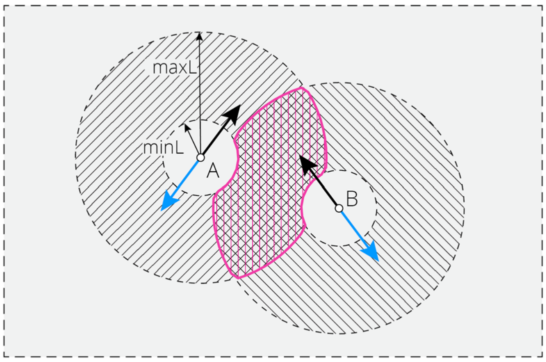

Figure 4: Constrained domain for planar design space (region in magenta).

3.3. Rule application

The performed rule transformations result in variant *networks of bars*, circumscribed into the *design space* (D_{space}) whose geometric boundaries and voids, if any, are explicitly defined. A network of bars, incomplete or connected, represented within the geometric boundaries of the design space is called *model* (M). The model at its inception usually only consists in the design space and the interim forces (applied loads and support reactions).

After manually setting up the static features of the design exploration, i.e. the applied forces and the design space boundaries, the designer chooses the rule features that transform the model accordingly. During the transformative process, these features are subject to change per the designer's decisions. Different sequences of rule features result in new design variants and favor the design space exploration in accordance with the designer's decisions. More precisely, the *entropy rate* expresses the designer's intention and is an explicit, ternary decision. For the other two features (*selection of force candidates* and the *location of node P*), the number of available options, called *policies*, is large but they are still defined by the designer.

The selection of force candidates is synthesized by *ranking policies*, i.e. chosen numerical criteria that are used to order the set of all feasible selections of force candidate(s), e.g. anchor points proximity, or force magnitude. The highest performing selection is the one the rule is applied onto. If the resulting D_f is empty ($D_f = \emptyset$), due to conflicting constraints that do not allow the network to retain its static equilibrium, the rule application is attempted onto the next force candidate, and so on.

Similarly, the location of *P* is determined through *input policies*, that impose either a *random* or an *explicit*, defined manually by the designer, location. For the random definition of a location, the policy describes the geometric domain(s) or sub-domain(s) where *P* is placed. For the explicit definition of a location, the designer manually selects the coordinates of *P*. When input policies allow randomness, additional constraints are feasible to be applied, e.g. bar length constraints.

After the rule features are set, the rule is applied onto the model. The force candidates are replaced by bars, in compression and tension, connected to the new node *P*, and new interim forces are introduced as imposed by the entropy rate and the necessity to retain static equilibrium. The entire process continues until the pool of interim forces is empty [Figure 5]. Since the network is in static equilibrium at the initial state and each transformation maintains static equilibrium, the eventual elimination of all interim forces is always possible.

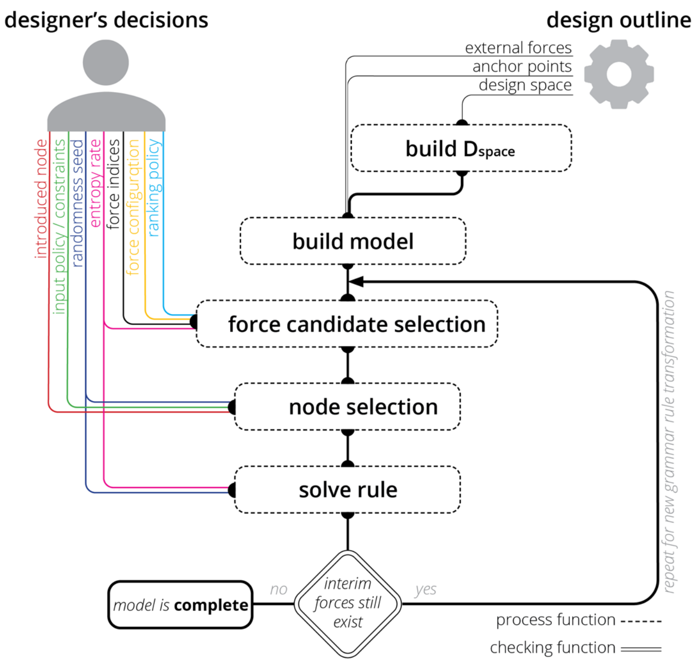

Figure 5: Algorithmic workflow for rule transformation; "designer's decisions" refer to rule-syntax related features, subject to change during the generative process; "design outline" refers to rule-unrelated features that remain static during the generative process

At each rule application, the chosen *force candidates* as well as the *location of P* have great impact on the transformative process overall. The incremental rule application and the aggregation of sub-networks to the *model* (M) at the end of each transformation, occurs blindly, without the designer being able to foresee how the decisions made on each step impact the DSE later during the process, e.g. if convergence of the design space is feasible on demand at the next transformation. This does not imply that due to specific rule features at the beginning of the process, the transformative process is ultimately unable to eliminate the pool of interim forces, but rather that there is an increase of the needed number of transformations before it is achieved. In other words, like in chess playing, where the human mind can only foresee and process ahead a limited number of movements that bring the player closer to the victory, the designer does not have the capacity to choose the sequence of policies that generate optimum, or near to optimum, non-intersecting networks of bars for example. While the design space converges, the range of choices the designer has gets limited until the moment that the occasion of undesired transformations is inevitable. In order to avoid these cases, the designer has to manually backtrack to the *model*'s earlier states and follow alternative design branches, i.e. different sequences of policies.

4. Application studies

The above described method is executed onto four case studies. No optimization is attempted, but rather the application focuses on other aspects: how rule-based leads to the generation of unexpected networks that still retain static equilibrium, how different design decisions (sequences of policies) result in different topologies and number of nodes/bars, how bounded bar lengths constrain the exploratory process, influence the feasibility of certain transformations and/or help to decrease the intersections between bars.

4.1. Incremental rule application for length-constrained networks

In this first study, a cantilevering network is generated [Figure 6] to showcase the incremental steps with applied constraints on the bar lengths. The constrained domain's computation is force candidate-dependent and represents the geometric domain (highlighted with hatch) where *P* is feasible to be placed, with respect to the desired, by the designer, entropy rate and the set length constraints. The entropy rate is set to divergence for the first transformation and for the remaining transformations convergence is attempted. For most of the steps, convergence is not possible due to the bar length constraints. When that is the case, the aimed transformation is deemed unfeasible and is replaced by a less-constraining transformation of decreasing entropy rate, as indicated in 3.1.1. Entropy rate.

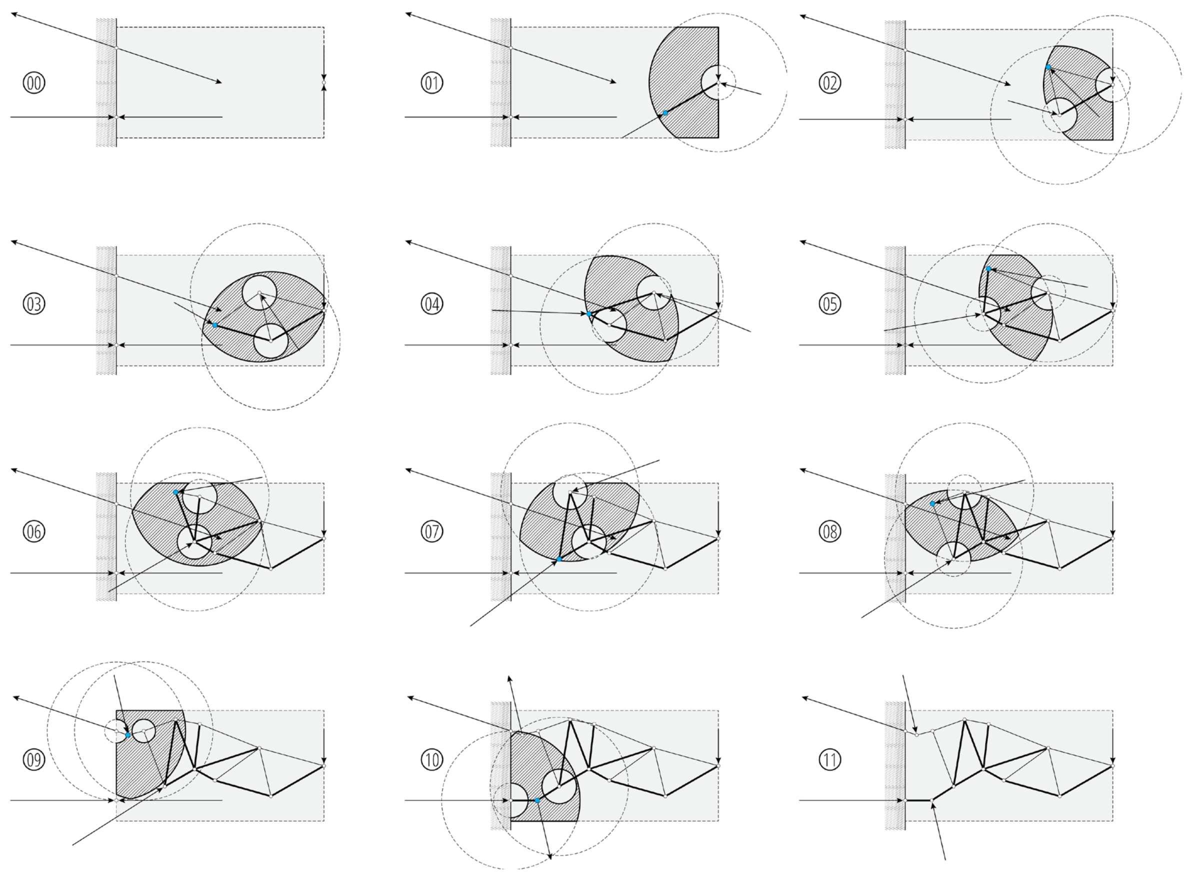

Figure 6: Incremental rule application for the generation of a cantilevering network of bars in compression and tension. The hatched regions represent the constrained domains, as defined by the bars' minimum and maximum lengths (represented with dashed circles). The newly introduced nodes are highlighted in cyan.

The *ranking policy* ranks the selections of force candidates based on their proximity to the right edge of the design space. If convergence is feasible for any of the interim forces found in the respective pool, the ranking policy is ignored and the rule is applied onto the converging force candidates. For all other cases, the ranking policy selects the force candidates. The *location of P* is generated randomly within the constrained domain, as bar length constraints apply. At the last step, the pair of equal and opposite interim forces is replaced by a bar in tension (tie); no introduction of a new node is necessary in this case.

4.1. Cantilevering networks of bars in compression and tension

In an attempt to highlight the importance of the location of *P* not only to the geometry but also to the topology of the networks, the same study was repeated 4 times. The connected network, as a result of the incremental rule application described above, is shown at Figure 7i. For comparison reasons, basic metrics are provided for each variant. For the second variant no bar length constraints are applied [Figure 7ii], whereas the rest of the design variants consider constrained bar lengths of [1.5, 10.0] units [Figure 7i] and [2.5, 10] units [Figure 7iii & Figure 7iv].

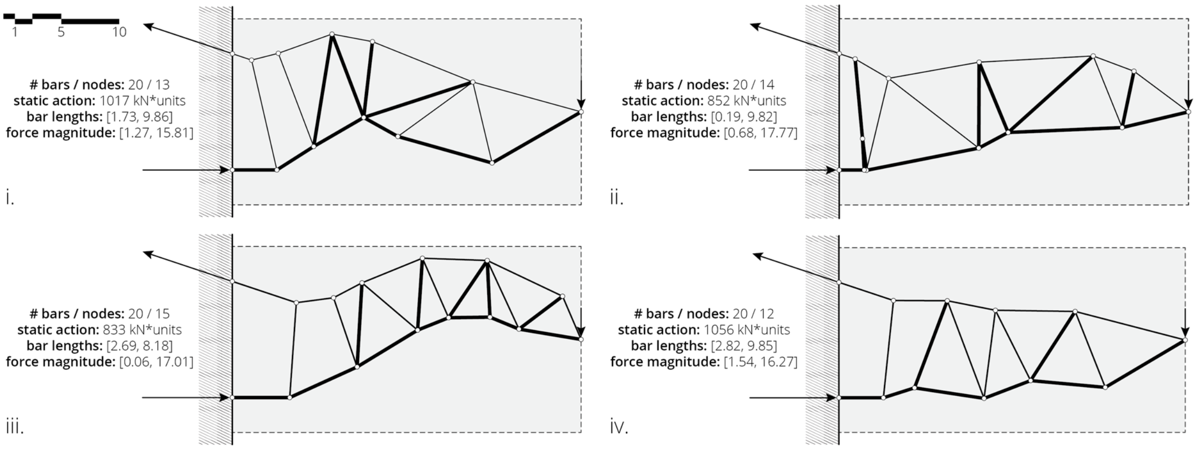

Figure 7: Cantilevering networks of bars.

4.3. Arch-like networks of bars in compression and tension

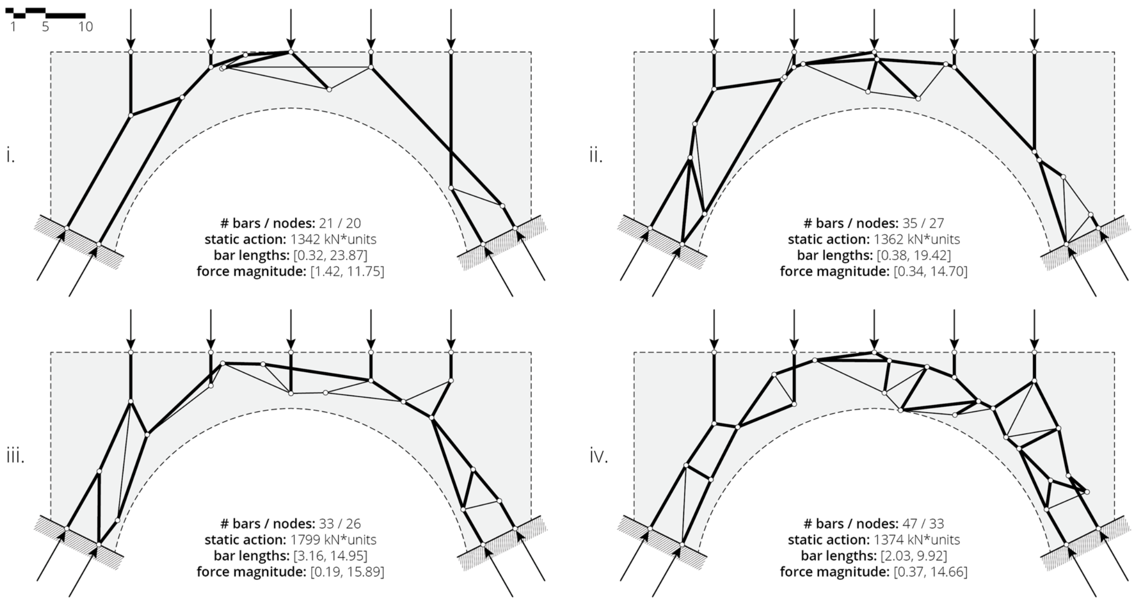

Figure 8: A collection of variant arch-like networks of bars as a result of different design parameters. For comparison reasons, useful metrics are provided for every design variant.

The second application study considers the exploration of a load-path within an arch-shaped design space. The considered applied loads are equally distributed on top of what could be a deck. The location of *P* is constrained by bar length constraints in Figure 8iii - Figure 8iv but not in Figure 8i - Figure 8ii. Precisely, the third and fourth variants consider constrained bar lengths of [3, 15] and [2, 10] units respectively. Comparing these design variants, it is evident that length-constraints not only prevent failure (i.e. buckling), but also increase the designer's control over the process and allow the generation of networks with no, or few, intersections between bars and reasonable bar lengths.

4.4. Tree-like networks of bars in compression and tension

The last case study generates alternative flows of forces for tree-like structures, supported at a single point. The use of constrained domains allows the generation of networks with less intersecting bars [Figure 9]. For the last three design variants, the length-constraints had to be deactivated when applying the last transformation in order to reach the state of global equilibrium without additional divergence, i.e. without the introduction of a greater number of bars. The use of different ranking policies has allowed distinctive variance and topology among the proposed designs.

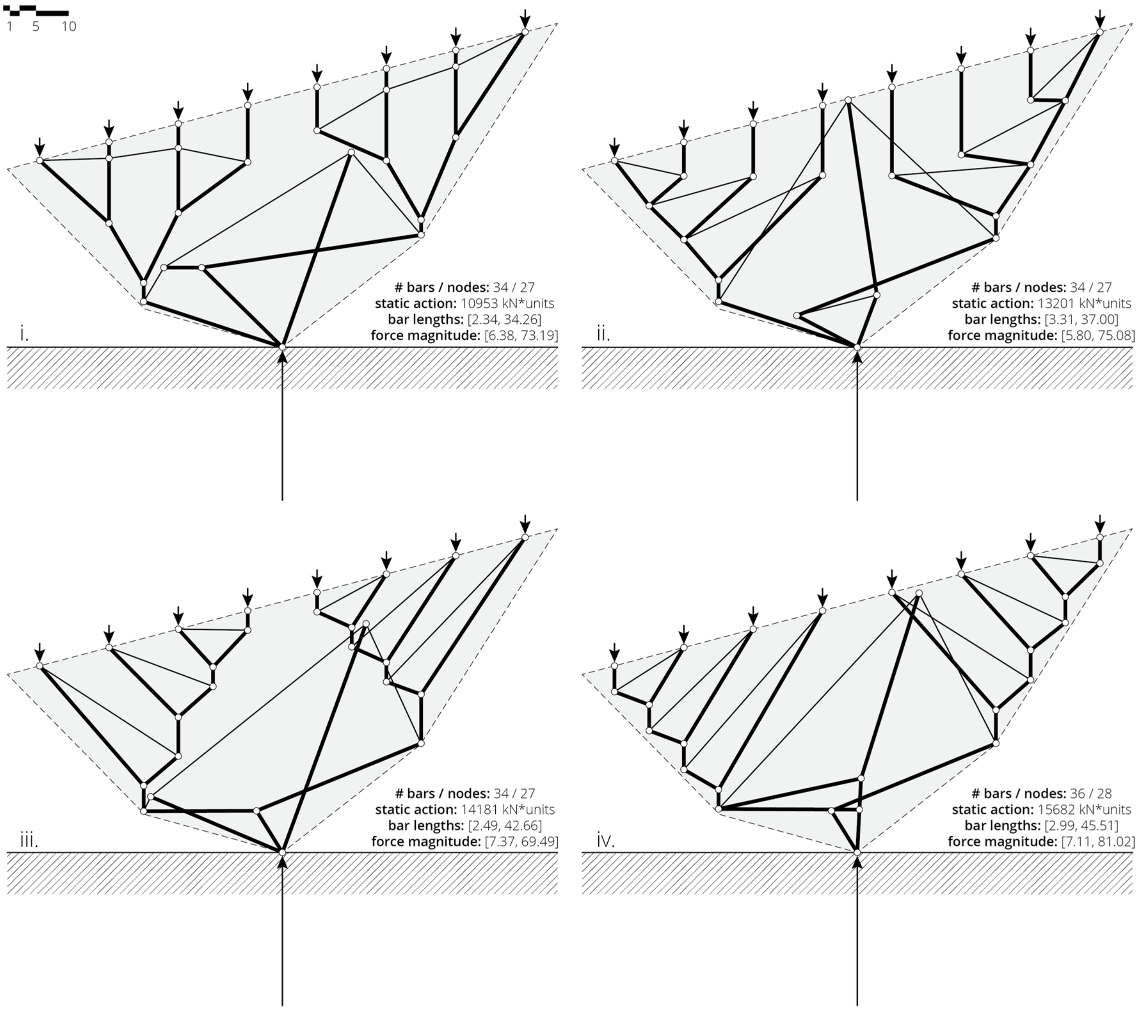

Figure 9: Tree-like networks of bars.

5. Conclusion

The proposed methodology has proven its capability of generating diverse reticulated networks of bars in static equilibrium with given external forces. The geometry, the topology and especially the number of bars of the final network is defined incrementally via a discrete set of values used to define the features of the grammar rule. The addition of bar length constraints to the rule, reduces the field of possibilities at each transformation but does not constrain the exploratory power of the rule-based approach as explained in this paper. Instead, it highlights how constraints along with the designer's decisions stir the generation of networks towards non-chaotic solutions, that respect material properties and that have less, or no, intersecting bars.

References

[1] Simon H.A., The structure of ill structured problems, Artificial Intelligence (1973) 181-201.

[2] Rittel H.W., Webber M.M., Dilemmas in a general theory of planning, Policy sciences 4 (2) (1973) 155-169.

[3] Paulson B.C. Jr, Designing to reduce construction costs, Journal of the construction division 192 (4) (1976) 587-592.

[4] Purcell A.T., Gero J.S., Design and Other Types of Fixation or Is Fixation Always Incompatible with Innovation?, Design Studies 17 (4) (1996) 363-383.

[5] Mirtsopoulos I., Fivet C. Grammar-based generation of trusses within non-convex domains, in: Proceedings of the International fib symposium on conceptual design of structures, 2019, p. 193-200.

[6] Mirtsopoulos I., Fivet C. Design space exploration through force-based grammar rule, archiDOCT, 2020, vol. 8, num 1, p. 50-64.

[7] Goldberg D.E., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, (1988), 13th edition.

[8] von Buelow P., Advantages of evolutionary computation used for exploration in the creative design process, Journal of Integrated Design and Process Science 11 (3) (2007) 5-18.

[9] von Buelow P., Paragen: Performative exploration of generative systems, Journal of the International Association for Shell and Spatial Structures 53 (4) (2012) 271-284.

[10] Martini K., Harmony search method for multimodal size, shape and topology optimization of structural frameworks, Journal of Structural Engineering 137 (11) (2011) 1332-1339.

[11] Mueller C., Ochsendorf J., An integrated computational approach for creative conceptual structural design, Proceedings of IASS Annual Symposium, vol. 2013, International Association for Shell and Spatial Structures IASS 2013, pp.1-6.

[12] Ohlbrock P.O., D' Acunto P., A computer-aided approach to equilibrium design based on graphic statics and combinatorial variations, Computer-Aided Design 121 (2020).

[13] Harding J., Brandt-Olsen C., Biomorpher: Interactive evolution for parametric design, International Journal of Architectural Computing 16 (2) (2018) 144-163.

[14] Stiny G., Gips J. Shape grammars and the generative specification of painting and sculpture, The Best Computer Papers of 1971, vol. 71, 1971, pp.1460-1465.

[15] Beirao J.N., Duarte J.P. Stouffs R., Creating specific grammas with generic grammars: Towards flexible urban design, Nexus Network Journal 13 (1) (2011) 73-111.

[16] Stiny G., Mitchell W.J., The palladian grammar, Environment and Planning B: Planning and Design 5 (1) (1978) 5-18.

[17] Chakrabarti A., Shea K., Stone R., Cagan J., Campbell M., Hernandez N.V., Wood K.L, Computer-based design synthesis research: an overview, Journal of Computing and Information Science in Engineering 11 (2) (2011).

[18] Geyer P., Multidisciplinary grammas supporting design optimization of buildings, Researching engineering Design 18 (4) (2008) 197-216.

[19] Shea K., Cagan J., Fenves S.J., A shape annealing approach to optimal truss design with dynamic grouping of members, Journal of Mechanical Design, Transactions of the ASME 119 (3) (1997) 388-384.

[20] Shea K., Cagan J., Languages and semantics of grammatical discrete structures, Ai Edam 13 (4) (1999) 241-251.

[21] Mueller C.T., Computational exploration of the structural design space, PhD Thesis, Massachusetts Institute of Technology (2014).

[22] Lee J., Mueller C., Fivet C., Automatic generation of diverse equilibrium structures through shape grammars and graphic statics, International Journal of Space Structures 31 (2-4) (2016) 147-164.