archi DOCT

The e-journal for the dissemination of doctoral research in architecture.

July 2020

www.archidoct.net

ISSN 2309-0103

VARIABILITY

15

Listed in Scopus®

ISSN 2309-0103
www.archidoct.net
Vol. 8 (1) / July 2020

Design space exploration through force-based grammar rule

Ioannis Mirtsopoulos, Corentin Fivet // Structural Xploration Lab (SXL), Ecole polytechnique fédérale de Lausanne (EPFL), Switzerland

Abstract

Design exploration frames the process of understanding design as a challenge and helps taming its complexity. It is a creative, but also paramount, process, that flourishes diversity, emergence and variability. The approaches used during this exploration stage can widen, or narrow, the domain where design variants can be found. Computational tools have shown their great potential to explore design in a fast way. However, the digitalization of the process is not sufficient to ensure the generation of radically new designs and subsequently not guaranteed to explore the full range of the design space. The design of architectural forms, that are structurally relevant, shares the same challenges and risks. Additionally, it introduces equilibrium constraints.

This paper presents a design framework, which fuses force-driven grammar rules for the generation of unprecedent spatial structural systems that go beyond any existing catalogues of mathematically known structural typologies. Operating on a rule-based level rather than on a variable-based level, allows the incremental transformation of the model and the backtracking to previous steps, while static equilibrium is always retained. The acquired transformations are possible to be slightly or highly constrained along with the designer's intended exploration preferences.

Keywords

grammar rule, graphic statics, static equilibrium, structural exploration, structural design

ISSN 2309-0103
www.archidoct.net
Vol. 8 (1) / July 2020

Introduction and problem statement

Design is an ill-structured problem [Simon 1973] characterized by open-ended expectations, emerging constraints, non-quantifiable features, the absence of global optimality and contradicting solution paths. As design is a "wicked problem" that "one cannot first understand, then solve" [Rittel and Webber 1973], it requires gaining knowledge about it. Designers tame this complexity through the creative processes of design exploration. Design exploration frames the systematic, iterative generation of design candidates. Ultimately, during this process they aim to gain and/or extend this knowledge. Traditionally, this knowledge only sources from the designer's own experience, fact that leads to the generation of resembling designs, which only represent a tiny fraction of the prospective design candidates. This tendency for premature design fixation is typically followed by lack of creativity.

Recently, the generation of architectural forms became assisted by parametric workflows. Parametric logic allows the variation of a finite set of numerical values within predefined domains, usually set by constraints [Mueller et al 2013]. Though this approach offers the possibility to alter design candidates in an automated way, it is not sufficient to automate the generation of radically new design candidates. Thus, the design freedom and exploration are limited by the available input parameters and the way their parameterization leads to the solution. Moreover, current design workflows seldom provide instant structural feedback. On the other hand, mainstream approaches to design spatial architectural forms that are structurally relevant consist, either, in adapting known geometries, or in searching the optimum solution of well-defined problems. The domain of structural forms in-between is yet to be explored.

Consequently, prospective ways for architects and engineers to improve the structural design process may consider the:

- Computational approaches that tackle the emerging constraints and ease the generation of alternative pathways; rather than computerized approaches that focus on drafting.
- Investment on rule-based rather than on variable-based design; parameters freed from predefined domains that structure the design logic itself.
- Integrated workflows of structural evaluation within the creative process; avoided structural feedback as a discrete and disconnected subsequent step, and structurally informed generations.

Following these principles, this paper presents a force-driven grammar rule, for the generative, interactive and conceptual design of planar structures. Its successive application within an algorithmic framework operates as a form-finding engine, capable of generating numerous design candidates in static equilibrium within a given design domain. Overall, this computational method shows premises to: (a) provide instant feedback on developed axial forces, (b) explore alternative conceptual structural designs and (c) unveil new typologies of structural systems.

Current state of research in the field of force-driven conceptual design

Computational methods, which allow designers to generate and explore the design space more quickly, while handling the design challenges incrementally, interactively, and in a creative way, are needed. This has become clear to researchers that are consistently contributing towards this direction. Current generative solvers are of two main kinds: iterative methods, when convergence is key (e.g. for form-finding of a mathematically approved design solution), and heuristics, when exploration is key. In both cases, the ultimate objective is achieved through numerical modification to the variables, rather than through generative rules that re-establish the topology of a structure and its variables' relationships (see chapter "Rule-based design | Shape grammars"). The former ones provide a global optimum. However, plenty of these approaches are not applicable when the number of design variables (e.g. the number of bar elements) increases, due to unmanageable computation time. This does not mean that they are unsuitable for structural design, but rather play a limited and very precise role in the design process [Fivet 2013]. On the other hand, recent research projects focus on generating a diverse set of near-optimum solutions [Martini 2011] [von Buelow 2012] [Mueller 2014], rather than aiming at finding the single optimum solution. These approaches allow designers to balance mathematical optimality with non-mathematically expressed and non-predictable criteria.

The stage of conceptual design includes two crucial operations: (a) creativity evocation through the design exploration, followed by (b) impactful decision making. Tools that assist good decision making at this stage are of great help to designers [Harding 2012, 2017]. Overall, structural design exploration is assisted by novel methodologies that merge conceptual structural design approaches, i.e. graphic statics, with evolutionary algorithms, i.e. genetic algorithms, and rule-based design, i.e. grammar rules/ shape grammars, which effectively explore the design space. Through the use of design grammars and interactive fitness functions, [Byrne et al., 2011] have shown that Grammatical Evolution is capable of creating surprising and innovative designs. Ultimately, creativity in design "is not simply concerned with the introduction of something new into a design, although that appears to be a necessary condition for any process that claims to be labeled as creative. Rather, the introduction of "something" new should lead to a result that is unexpected (as well as being valuable)" [Gero 1996].

Graphical form-finding | Graphic statics

Static equilibrium is a basic requirement that all structures must satisfy. A handy way to handle networks in static equilibrium are graphic static methods. Graphic statics combine two reciprocal figures: a form diagram and a force diagram. The force diagram is a vector representation of the force magnitudes and directions within the network drawn in the form diagram. Static equilibrium of a sub-set of bars and nodes in the form diagram is shown by a closed force polygon in the force diagram. Methods of graphic statics were created in the 19th century, to analyze structures. Nowadays, combined with contemporary graphical and computational capabilities of computers, graphic statics gain new relevance for early-stage structural design. Rather than assuming that modern graphic statics are just a computerized version of classical graphic statics, [Ohlbrock 2020] [Lee 2016] developed methodologies that implement graphic statics to generate new structures. Both methodologies are mainly applicable to conceptual and early-stage structural design and ensure static equilibrium without considering the actual material. This means that post-processing is required to size the members etc.

Rule-based design | Shape grammars

[Stiny, Gips, 1972] introduced the term of shape grammars for design, inspired by Noam Chomsky's theories on generative grammars in language. "[Chomsky's] idea was that a grammar had a limited number of rules that could generate an unlimited number of different things, and the resulting language was the set of things the rules produced". The concept of rule-based design is equivalent to shape grammars and opposed to that of variable-based design. Briefly, variable-based design con-

ISSN 2309-0103
www.archidoct.net
Vol. 8 (1) / July 2020

structs a model from scratch to completion in one go, whereas rule-based design transforms the model in incremental steps (Fig.1).

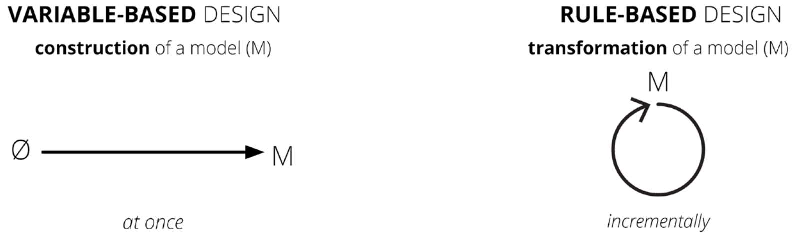

FIgure 1.
Variable-based design vs rule-based design

Variable-based design corresponds to assigning different values to a number of (geometrical) parameters [Strobbe et al. 2015], while rule-based design deals with alternation and rearrangement of design components. Consequently, the former approach is framed around variable relationships that have been predefined and alterations are only allowed at a numerical level, i.e. variables' numerical values. The latter builds purely on topological relationships, easing alterations because the designer has the freedom to reset the existing relationships (numerical or topological). Hence, variable-based design considers designs that are only constructed within human-defined, potentially arbitrary numerical domains, whereas rule-based design considers every possible design, within the design domain. Thus, rule-based design favors diversity, variability, exploration, as well as emergence [Mitchell 1993] [Stiny 1994], without excluding exploitation, if rules are accordingly defined. That said, it does not mean that variable-based design is not beneficial during the design process. It allows for parameterizing the construction of models, that reduces the aim and effort required for changes and reuse [Aish 2005] and eases the numerical tuning of a design (i.e. during optimization).

Shea and Kagan [Shea 1997] [Shea 1999] applied shape grammars to the synthesis of triangulated trusses. Simulated annealing was used to obtain the policy of transformations leading to the optimal shape. [Mueller 2014] applied structural grammars both randomly and manually to generate diverse sets of structural systems. Grammar rules in this case are context specific. [Chakrabarti 2011] reviewed the application of graph grammars (an abstract generalization of shape grammars) for design synthesis.

Method

Mitchell [Mitchell 1991] refers to shape grammars as functional grammars, when the generated design satisfies two conditions: (a) it is realizable using available materials and fabrication processes and (b) it meets specified functional requirements. The present work extends this idea: (a) its theoretical base builds on vector-based graphic statics, which is used to define the syntax of a universal grammar rule, and ensures the static equilibrium of the structure, and (b) a design candidate that satisfies equilibrium is likely to be realizable, provided that the required fabrication processes are available. Analytically, the proposed methodology aims at the transition from a disconnected network in interim equilibrium, to an assembled (complete) one in global static equilibrium. Throughout this transition, the designer has control over parameters that allow him/her to steer the design towards directions that satisfy emerging constraints and, hence, actively explore alternative design candidates that meet functional, or aesthetics-related, requirements. Considering qualitative aspects during the generation of networks is a great asset, because even though the generated forms are always in static equilibrium, not all of them represent meaningful structural forms.

As explained beforehand, the proposed method starts with a network (model object, m) comprised of forces in interim static equilibrium, and, eventually, discrete linear elements (bars). Bar networks consist of axially loaded members in compression or tension, are connected by nodes and are in static equilibrium. The interim equilibrium is ensured by a number of interim forces (pool of forces) that act on the network nodes only (applied loads or support reactions). The interim network is contained within a bounded territory, named as design space or design domain (Dd) that is also part of the model object. This domain defines the geometric space, where the network evolves when the designer applies the grammar rule. The complete network, as a result of the recursive rule application, is in global static equilibrium and is also contained within the design domain.

1. Force-driven rule syntax

The proposed grammar rule is force-driven and, thus, it is constructed to retain static equilibrium when applied. Its inception and originality derive from graphic statics (Fig.2). Static equilibrium is already ensured at a rule-level, which means that no additional adjustments need to be made to the existing model to force static equilibrium. This feature, allows the user to easily backtrack on previous steps/iterations of the model, as part of the design exploration process.

The rule is applied on a selection of interim forces, which are found in the pool and are coupled in various configurations to form force candidates (see chapter "Force candidate objects"). In every iteration, the user is invited to control the type of transformation that the rule application will have on the interim network (see chapter "Entropy rate"). The chosen type of transformation is usually only satisfied by a limited number of (feasible) force candidates. Again, the user has the possibility to actively select (see chapter "Ranking policy") the chosen candidate, among the feasible ones.

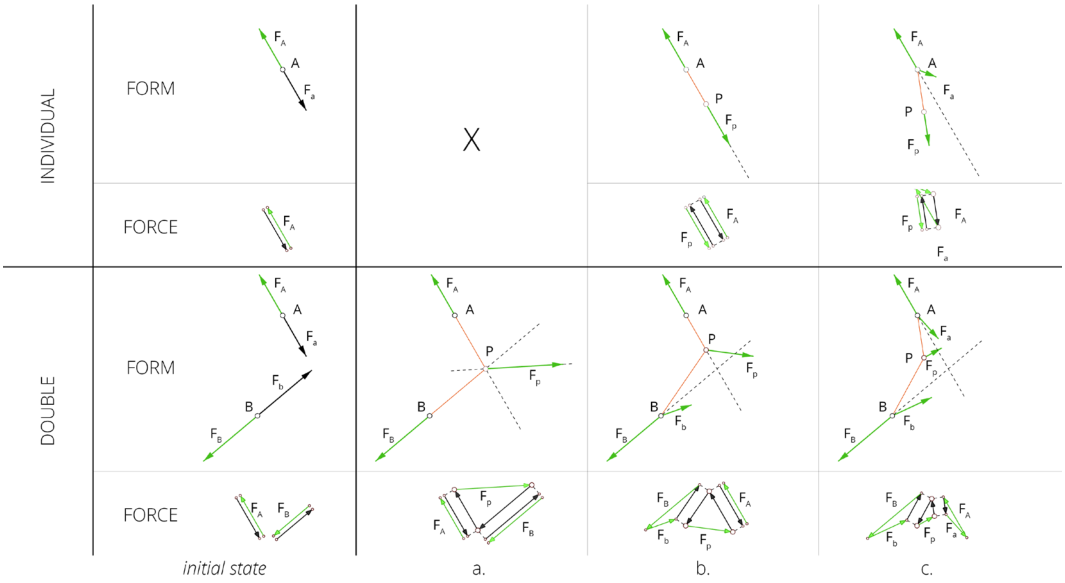

Figure 2.
Grammar rule syntax (a. convergence, b. stagnation, c. divergence)

ISSN 2309-0103
www.archidoct.net
Vol. 8 (1) / July 2020

After, the rule is applied onto the chosen candidate, aiming to transform the network, according to the selected entropy rate. A network undergoes three different geometric transformations as a result of different entropy rates: converge, stagnate, or diverge, i.e. decrease, maintain or increase the total number of remaining interim forces. The interim forced are replaced by bars, adjacent to a new point (ptnew), and when necessary, new interim forces are introduced. These introductions are imposed by the necessity to keep an incomplete network in interim equilibrium. The process continues until no interim forces remain.

2. Force candidate objects

The force candidate objects represent all possible combinations of interim forces where the grammar rule is applied onto. In two dimensions, there are two types of candidate objects: individuals and doubles (Fig.3). Each type includes one or two interim forces, including all possible sequential configurations.

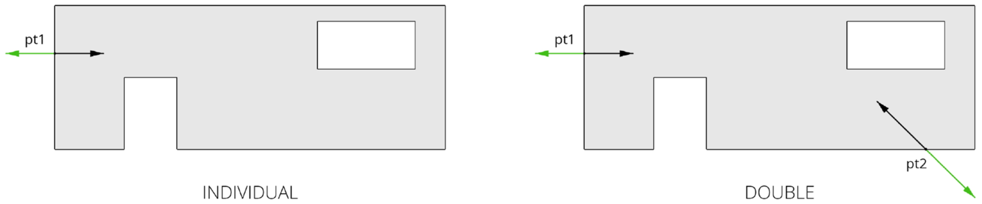

FIgure 3.
Types of force candidates; reflect the number of involved forces

Individuals refer to unpaired interim forces. Hence, the number of individual candidates is equal to the number of interim forces in the network. This candidate type does not allow the network to converge when the rule is applied. All other transformations are feasible though. Doubles refer to all possible couples of forces. The sequence of the forces forming a double candidate results in different allowable transformations under identical conditions. For that, all possible sequential configurations have to be considered. Doubles allow all types of transformations. The total number of prospective force candidates is calculated with the following expression:

$$no_{candidates} = \binom{n}{1} + \binom{n}{2} = n + n(n-1) = n^2$$

where n, the number of temporary forces.

For example, a network with three interim forces (i.e. A, B, C) has a total of 9 prospective force candidates, as shown at the table below. However, their feasibility to undertake the chosen transformation is not known yet.

individuals:	A, B, C
doubles:	AB, AC, BA, CA, BC, CB

3. Designer-controlled parameters

The designer can actively control the way the network transforms via two variables: entropy rate and ranking policy. Analytically:

Entropy rate

The entropy rate controls the rule's application impact onto the system. The rule application includes the introduction of a new point too. The preferred type of transformation, constrains the domain where this new point is located at, either to a single point, or on a segment, or within a planar region. This strictly defined domain, called entropy rate domain (Der), refers to the new point's prospective location and is force candidate-specific, i.e. it directly depends on the anchor point and the direction of the forces formulating a force candidate object (see chapter "Entropy rate domain").

Ranking policy

The force candidate objects, which allow the rule application, namely, the feasible candidates, are sorted out according to designer-defined preferences that can be performance-, geometry- or aesthetics-related. These preferences are called ranking policies. The feasible candidates are ranked according to them. Examples of policies could be:

- Choose the two forces whose distance between anchor points is minimal/maximal;
- Choose the two forces that are the oldest (or newest) ones in the pool;
- Choose the two forces with maximal (or minimal) magnitude;
- Choose the two forces whose design space of new node has the largest (or smallest) area;
- Choose the two forces whose design space of new node has the narrowest (or widest) area;
- Choose the two forces whose orientations are the most parallel (or perpendicular);
- Choose the two forces that are applied the closest (or furthest) from the boundary of the feasible region;
- etc.

4. Feasibility domain

Simply coupling interim forces to construct force candidate objects is not sufficient to apply the rule onto. A candidate is feasible, only if after the rule application the introduced point and the bars are contained within the design domain and the model retains its static equilibrium, namely if the feasibility domain (Df) is not empty. The way the Df is constructed, as well as the supplementary and supportive domains that need to be constructed per candidate, are described below:

I Entropy Rate Domain (D_{er})

The entropy rate domain describes the geometric domain where the introduced point (ptnew) must be located at, in order to achieve the chosen transformation after the rule application. Its size depends on the entropy rate. For *convergence*, the entropy rate domain explicitly consists of a single point in space ($D_{er, conv}$ = {x, y, z} and {x, y, z} $\in D_d$). As such, there is a unique solution for every feasible *double* (force candidate) that allows the network to converge (Fig. 4a). *Stagnation* is ensured for all points introduced along the segment that is defined by the force direction of the force candidate (Fig. 4b and 4c). For *divergence*, the entropy rate domain is irrelevant to the candidate type,

ISSN 2309-0103
www.archidoct.net
Vol. 8 (1) / July 2020

and always equals the *design domain* ($D_{er, div} = D_d$) (Fig. 4d and 4e). In order the grammar rule to be applicable to a force candidate object, the following two conditions must be satisfied, among others; the *entropy rate domain:* (1) must not be empty ($D_{er} \neq \emptyset$) and (2) must be a subset of the design domain ($D_{er} \subseteq D_d$). If conditions (1) and (2) are satisfied, it is known in advance that a new point can be generated inside the design domain. However, this is not a sufficient condition to define whether the chosen transformation is feasible or not.

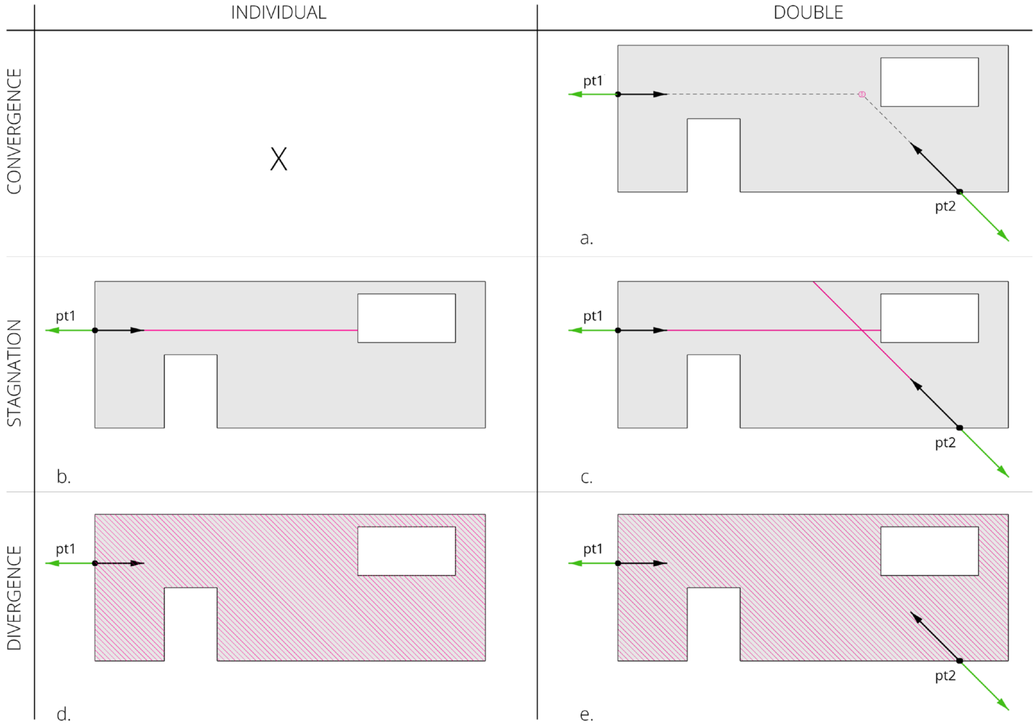

FIgure 4.
Entropy rate domains (magenta), green arrows are applied loads or reactions, black arrows are interim forces.

II Constructability Domain (Dc)

As the rule aims to replace the interim forces by new bars, the new point needs to be reachable ("visible") by the anchor points where the interim forces are applied to, i.e. one anchor point for *individuals*, two anchor points for *doubles*. Geometrically speaking, the segment connecting the new point with the anchor points must be uninterrupted by voids or non-convexities of the design domain. This new restriction defines a new domain, called *constructability domain* (D_c) and defines the region where the creation of two continuous segments, i.e. from one anchor point to the new point and from that point to the other anchor point (Fig. 5), is feasible. The *constructability domain:* (1) must not be empty ($D_c \neq \emptyset$) and (2) must be a subset of the *design domain* ($D_c \subseteq D_d$). If conditions (1) and (2) are satisfied, it is known in advance that the two bars can be built inside the *design domain.*

ISSN 2309-0103
www.archidoct.net
Vol. 8 (1) / July 2020

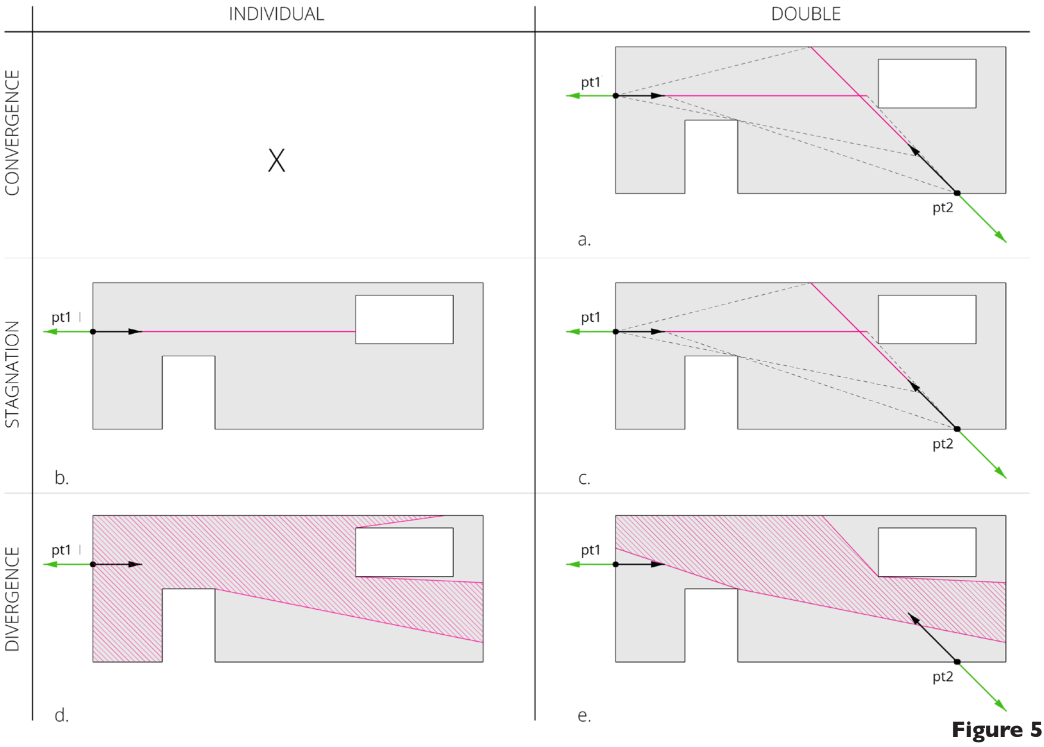

Figure 5.
Constructability domains (purple), green arrows are applied loads or reactions, black arrows are interim forces.

III Feasibility Domain (Df)

As part of the transformation, and in order for the model to retain its static equilibrium, a new point (pt_{new}) and one (Fig. 6b and 6d), or two (Fig. 6a, 6c and 6e), new bars (b_1 and b_2) are introduced. The bars replace some of the existing temporary forces. In order to ensure that the chosen geometric transformation is feasible, pt_{new} must belong to D_{er} ($pt_{new} = \{x, y, z\} \in D_{er}$). On the other hand, in order to ensure that the bars b_1 and b_2, adjacent to pt_{new}, can be built, they must belong to D_c ($\{b_1, b_2\} \in D_c$). These two requirements are met if, and only if, the intersection of D_{er} and D_c belong to D_d ($D_{er} \cap D_c \in D_d$). This intersection defines the candidate's *feasibility domain* (D_f) for a specific *entropy rate*. This means that the chosen geometric transformation of the model is only feasible if D_c is not empty ($D_c \neq \emptyset$). As D_{er} and D_c are candidate-specific, D_f is candidate-specific too. It is unknown whether a candidate can successfully lead to the chosen geometric transformation, before computing D_f. At the same time, more than one candidate might have non-empty D_f. In that case, they all qualify as feasible candidates and a force candidate will be selected according to the chosen ranking policy.

5 Model Update

The rule application results in transforming geometrically the model. The evident changes include: (1) the introduction of a new node (pt_{new}); (2) the iterative replacement of the interim forces by new bars in compression or tension, inherently retaining static equilibrium; (3) the introduction of new interim forces when necessary. Before the rule application, forces (both as individuals and as pairs) are sorted according to the ranking policies. Next, they construct a force candidate and its

ISSN 2309-0103
www.archidoct.net
Vol. 8 (1) / July 2020

feasibility domain is computed. If the domain is empty, the next pair of forces, or individual force, constructs a new force candidate. The procedure continues (a) either until either the candidate's feasibility domain is not empty or (b) until all sorted forces have constructed a force candidate. In the former case, the rule application continues. The model is safely transformed retaining its static equilibrium and the involved temporary forces are eliminated from the pool of forces. This procedure continues until the pool of forces is empty. At that moment, the model is complete. In case all forces have constructed an unfeasible force candidate, the entropy rate has to get decreased. If the chosen entropy rate is convergence, the entropy rate updates to stagnation. If stagnation is chosen, the entropy rate updates to divergence. If divergence still does not yield a feasible candidate, the process is terminated. The decrease happens because a candidate has more chances to diverge than to converge. That can be explained by looking at the size of the respective D_{er} domains.

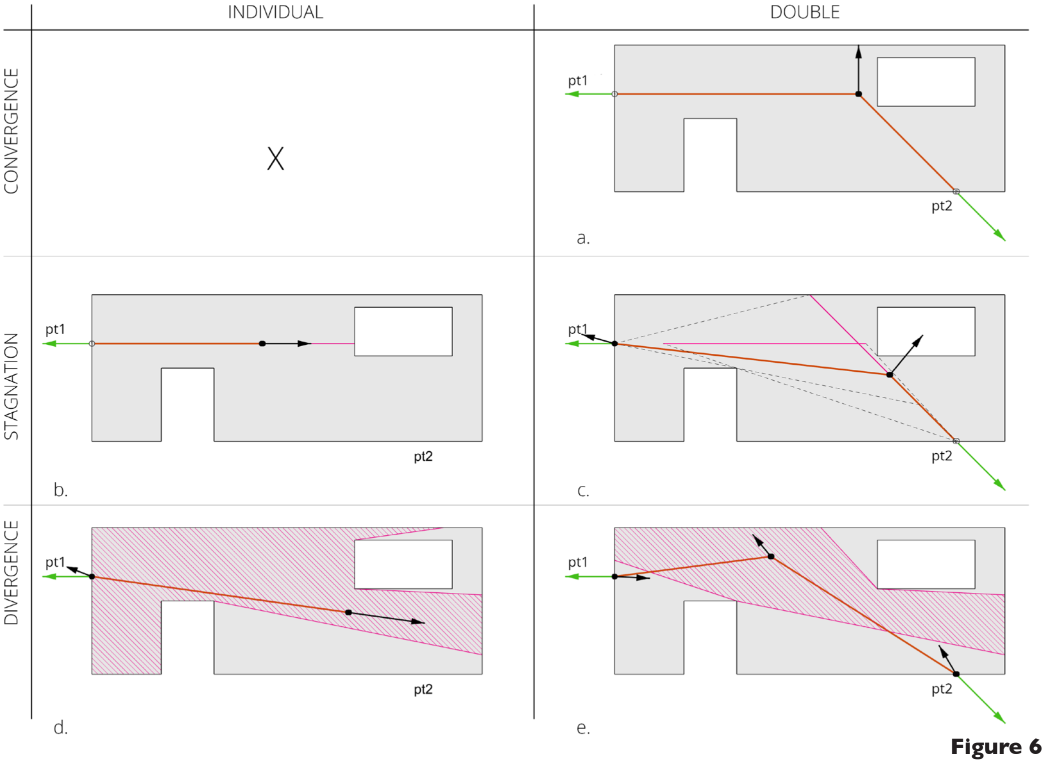

Figure 6.
Rule application results. Feasibility domains (purple), green arrows are applied loads or reactions, black arrows are interim forces, red bars are ties.

Application studies

The rule application on an interim model, consists in incremental transformations that complete the model and bring it in a state of global static equilibrium. This incremental procedure is shown below (Fig. 6). These transformations are highly affected by chosen *entropy rates*, chosen *ranking policies* and the nature of the design domain (e.g. presence of non-convexities or voids). Affected aspects include the total number of steps required to complete an interim network, the sequence of entropy rates, which might be altered if no force candidate is feasible for the chosen entropy rate, the topology

ISSN 2309-0103
www.archidoct.net
Vol. 8 (1) / July 2020

and the geometry of the model, as well as the final aesthetics and the design freedom overall.

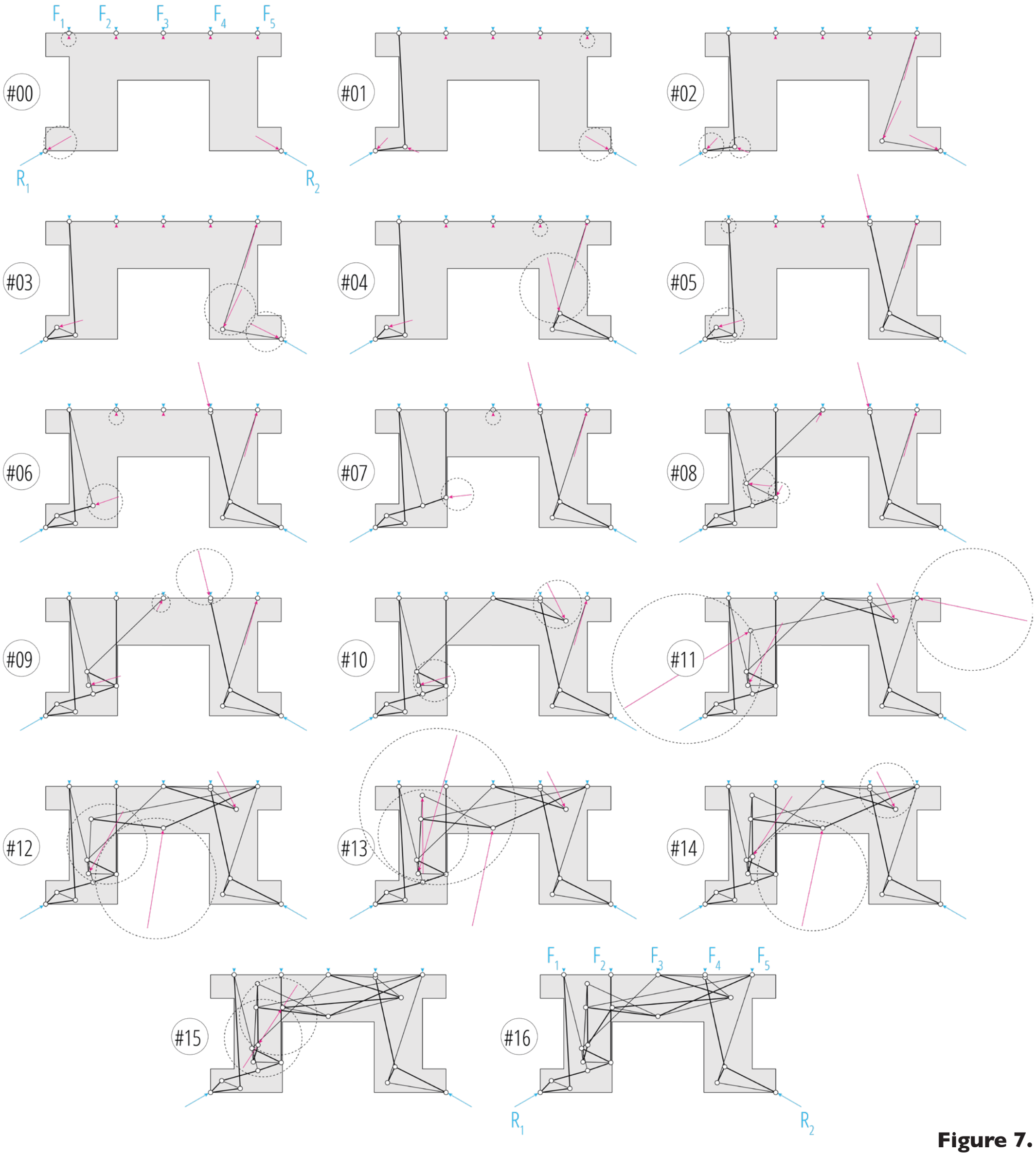

Figure 7.

The rule is incrementally applied and converges after 16 steps. Forces selected to form a force candidate object are circled. The process terminates when the pool of interim forces is empty.

1. Nature of design domain

The shape of the design domain has remarkable impact on the incremental rule application, depending on its convexity and the presence of voids. In general, added complexity increases the number of transformations required until an incomplete network converges to a complete network in global static equilibrium. The small size of the convergence entropy rate domain disqualifies prospective

ISSN 2309-0103
www.archidoct.net
Vol. 8 (1) / July 2020

force candidates to be chosen for the rule application. Often, no force candidate can undergo convergence and the entropy rate has to get decreased. Below the impact of convexity and cavities is demonstrated.

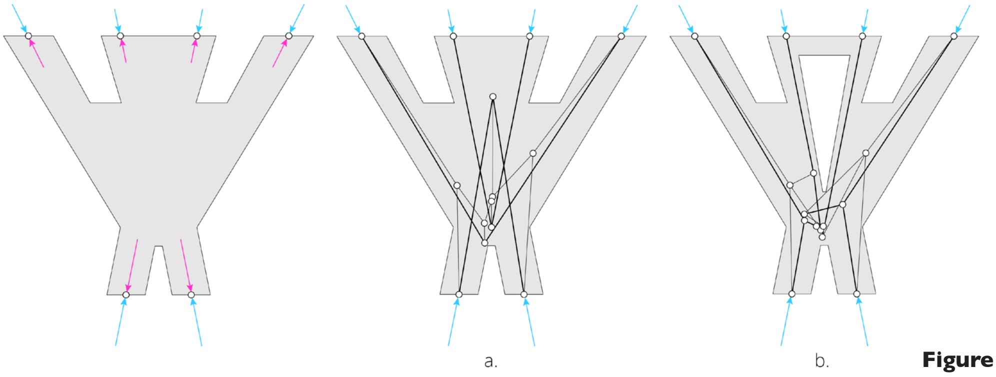

Figure 8.
Impact of design domain on rule application: (a) non-convex boundary without void: 9 steps, (b) non-convex boundary with convex void: 13 steps

2 Entropy rate sequence

This case study considers the exploration of a load-path between two equal, opposite forces (Fig. 9). Different number of initial diverging rules is considered on each row, from left to right, all subsequent applied rules are converging.

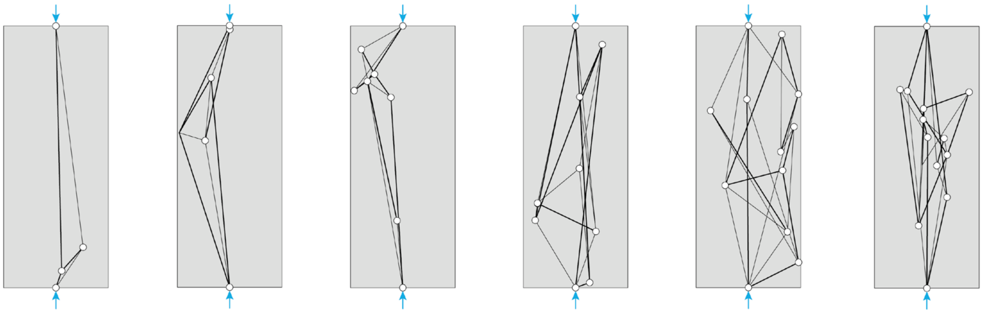

Figure 9.
From left to right the number of steps the system diverges for increases by 1. As a consequence, the number of steps required to converge increases. These additional steps introduce more bars and the structure becomes denser.

3 Ranking policy and selected force candidate

In this case, the situation of two symmetrical supports and a uniformly distributed force on a plane is examined (Fig. 10). Two different ranking policies to select the forces candidates are studied: random selection (Fig. 10a); selection of the two forces whose intersection is closest, which naturally leads to an arch (Fig. 10b). In both cases convergence is chosen.

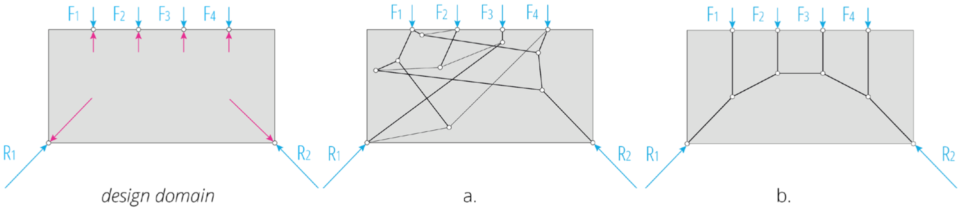

Figure 10.

In the middle (a), the force candidates are selected randomly. This influences the candidate's ability to convergence. On the right (b), choosing the ranking policy intelligently on every step, leads to the generation of a rational structure.

Conclusions

The proposed methodology has proven its capability of generating versatile reticulated structures in static equilibrium. Currently the location of the new node is randomly chosen within the feasibility domain, whose generation is the main concern, and thus final structures look neither rational nor functional. In cases that optimization for specific objectives is not part of the design process, the creation of a feasibility domain allows the designer to understand the variability range. His/her decision making will influence the size of the feasibility domain at the following step and, in combination with the intended entropy rate, shrink or expand the variability range.

Contrary to precedents, static equilibrium at every intermediate step is not achieved through triangulation but through the structural awareness that is embedded into the grammar rule syntax. This way, the design space can be explored thoroughly. Regardless of the number of completed steps and the size of the existing structure, i.e. number nodes and bars, in every iteration, the selected force candidate acts as a sub-system in interim equilibrium. Consequently, the algorithmic complexity does not increase along with the number of steps. Specifically, it is polynomial, as the application of the grammar rule only requires solving a simple matrix. After the rule application, this sub-system gets integrated into the existing structure.

Very few methods allow the designer to choose the number of steps which the process will converge, stagnate or diverge for. The user has direct control over the convergence (or divergence) on demand. Additionally, the fact that interim static equilibrium is ensured at all intermediate steps of the process, allows backtracking on previous design variants and favors exploration of the design space.

Last but not least, the model transformations (convergence, stagnation, divergence) are part of the same, universal rule, i.e. the same matrix is used to solve the rule application. Contrary to previous approaches, this rule is independent of specific structural typologies and is unaware of the designer's intentions. This disconnection allows designers to escape from catalogs of structural systems and frames a new, broader, domain, where new structural forms are to be discovered.

ISSN 2309-0103
www.archidoct.net
Vol. 8 (1) / July 2020

References

Aish, R. & Woodbury, R. 2005, "Multi-level interaction in parametric design", in A. Butz, B. Fisher, A. Krüger, P. Olivier (ed), Smart Graphics 2005, Lecture Notes in Computer Science, vol 3638. Springer, Berlin, Heidelberg

Byrne, J., Fenton, M., Hemberg, E., McDermott, J., O'Neil, M., Shotton, E., Nally, C. 2011, "Combining structural analysis and multi-objective criteria for evolutionary architectural design", in Di Chio et al. (ed.) Applications of Evolutionary Computation, Springer, pp. 204-213

Chakrabarti, A., Shea, K., Stone, R., Cagan, J., Campbell, M., Hernandez, N., Wood, K., L. 2011, "Computer-Based Design Synthesis Research: An Overview", Journal of Computing and Information Science in Engineering, June 2011, vol. 11.

Fivet, C. 2013, "Constrained Based Graphic Statics", PhD Thesis, UCLouvain, Belgium.

Gero, J, S 1996, "Computers and creative design", in M. Tan (ed.), The Global Design Studio: National University of Singapore, pp.11-19.

Harding, J. Joyce, S. Shepherd, P. Williams, C. 2012, "Thinking topologically at early stage parametric design" in L. Hesselgren, S. Sharma, J. Wallner, N. Baldassini, P. Bompas, J. Raynaud (ed.), Advances in Architectural Goemetry, 2012, Springer, Vienna.

Harding, JE. Shepherd, P 2017, "Meta-Parametric Design", Design Studies, vol.52, pp.73-95.

Martini, K. 2011, "Harmony search method for multimodal size, shape and topology optimization of structural frameworks", Journal of Structural Engineering, vol.137, no. 11, pp. 1332-1339.

Mitchell, WJ. 1991, "Functional Grammars: An introduction", Proceedings of ACADIA 1991, Los Angeles, California, p.167-176.

Mitchell, WJ. 1993, "A computational view of design creativity", in JS. Gero and ML. Maher (ed.), Modelling Creativity and Knowledge-Based Creative Design, Hillsdale, NJ: Lawrence Erlbaum, pp. 25-42.

Mueller, C. Ochsendorf, J. 2013, "An integrated computational approach for creative conceptual structural design." Proceedings of the International Association for Shell and Spatial Structures, Wroclaw, Poland.

Mueller, C. 2014, "Computational exploration of the structural design space", PhD Thesis, Massachusetts Institute of Technology, Cambridge, USA.

Lee, J. Mueller, C. Fivet, C. 2016, "Automatic generation of diverse equilibrium structures through shape grammars and graphic statics", International Journal of Space Structures, vol.

31, no.2-3, pp. 146-163.

Ohlbrock, OP., D'Acunto, P. 2020, A computer-aided approach to equilibrium design based on graphic statics and combinatorial variations, Computer Aided Design, vol. 121

Rittel, HW, Webber MM 1973, "Dilemmas in a general theory of planning", Policy sciences, vol. 4, no. 2, pp 155-169.

Shea, K. Cagan, J. Fenves, S. 1997, "A shape annealing approach to optimal truss design with dynamic grouping of members." Journal of Mechanical Design, vol. 199 no. 3, pp. 388-394.

Shea, K. Cagan, J. 1999, "Languages and semantics of grammatical discrete structures." Artificial Intelligence for engineering design, analysis and manufacturing, vol. 13, pp. 241-251.

Simon, HA. 1973, "The structure of ill-structured problems", Artificial Intelligence, vol. 4, no. 3-4, pp. 181-201.

Stiny, G. & Gips, J. 1972, "Shape Grammars and the Generative Specification of Painting and Sculpture", Information Processing, vol. 71, pp. 1460-1465, Amsterdam.

Stiny, G. 1994, "Shape rules: closure, continuity and emergence", "Environment and Planning B: Planning and Design", vol. 21, no.7, pp. 49-78.

Strobbe, T. Pauweis, P. Verstraeten, R. De Meyer, R. Van Campenhout, J. 2015 "Toward a visual approach in the exploration of shape grammars", Artificial Intelligence for engineering design, analysis and manufacturing, vol. 29, no. 4, pp. 503-521.

von Buelow, P. 2012, "ParaGen: Performative Exploration of Generative Systems", Journal of the International Association for Shell and Spatial Structures, vol. 53, no. 4, pp.271.