Proceedings of the IASS 2024 Symposium
Redefining the Art of Structural Design
August 26-30, 2024, Zurich, Switzerland
Philippe Block, Giulia Boller, Catherine DeWolf,
Jacqueline Pauli, Walter Kaufmann (eds.)

A Graph-Based Grammar for Structural Design using Deep Reinforcement Learning

Lazlo BLEKER[*,a], Kam-Ming Mark TAM[b], Pierluigi D'ACUNTO[a,c]

[*,a] Technical University of Munich, School of Engineering and Design, Professorship of Structural Design
Arcisstraße 21 Munich 80333, Germany
lazlo.bleker@tum.de

[b] University of Hong Kong, Department of Architecture
[c] Technical University of Munich, Institute for Advanced Study

Abstract

This paper introduces a graph-based grammar method that combines the Combinatorial Equilibrium Modeling (CEM) form-finding approach and Deep Reinforcement Learning (RL). We formalize the design process of the CEM as a Markov Decision Process (MDP) that can act as an environment for an RL agent. This discrete step-wise design process allows both geometrical and topological manipulation of structural designs through a structural grammar consisting of a specific set of design actions. All design actions achieve design state transitions that preserve not only equilibrium but also the validity of the input structural topology. This guarantees that all design states are fully interactable parametric models, enabling intuitive manipulation of intermediate structural solutions by human designers. Parameters are represented by the input features of the CEM that relate to physical quantities of the structure such as desired edge/member lengths and forces. The combination of the top-down constraints of the CEM and the bottom-up design actions of the MDP results in a cross-typological design space that can be collaboratively explored by a human designer and an RL agent.

Keywords: structural grammar, form-finding, reinforcement learning, graph neural networks.

1. Introduction

A significant portion of the embodied carbon of buildings and infrastructures can be traced back to their load-bearing structures, underlining the critical need for minimizing their material use [1]. Especially for reticulated/discrete structures, layout design in the conceptual design phase presents the most significant opportunity for designers to influence the material efficiency of these systems [2].

In this context, Machine Learning (ML) presents a significant opportunity for designers and engineers to harness the increasingly large computation power at their disposal to assist in the exploration of efficient structural designs. While for structural analysis there exists a vast body of successful ML applications [3], applications of ML to conceptual structural design are not particularly well covered in the literature despite the proven potential brought about by leveraging ML algorithms [4].

One of the reasons for the discrepancy in applications of ML to structural analysis and conceptual structural design is the complex nature of design problems in general, sometimes referred to as *wicked* [5], which requires dealing with both qualitative and quantitative objectives and reformulating the problem while it is still being solved [6]. To address these challenges, current ML applications to conceptual structural design often incorporate a human-in-the-loop component in which structural designs are generated, evaluated by a human designer, and then re-generated accordingly [7, 8].

We propose a novel human-machine collaborative design environment for conceptual structural design combining structural grammars with Deep Reinforcement Learning (RL). The iterative nature of structural grammars and RL allows human and machine to directly collaborate on the design of the same structure (Figure 1). Human and machine both operate at the level of individual design decisions overcoming the limitation of needing a completed design before this collaboration can first take place.

We introduce a structural grammar based on the Combinatorial Equilibrium Modeling (CEM) [9, 10], a form-finding method based on Vector-based 3D Graphic Statics [11, 12] and graph theory. We formulate this structural grammar as a Markov Decision Process (MDP) in which states are represented as graphs. In this way, the structural grammar can be interacted with not only by designers and engineers, but also by RL agents based on Graph Neural Networks (GNN).

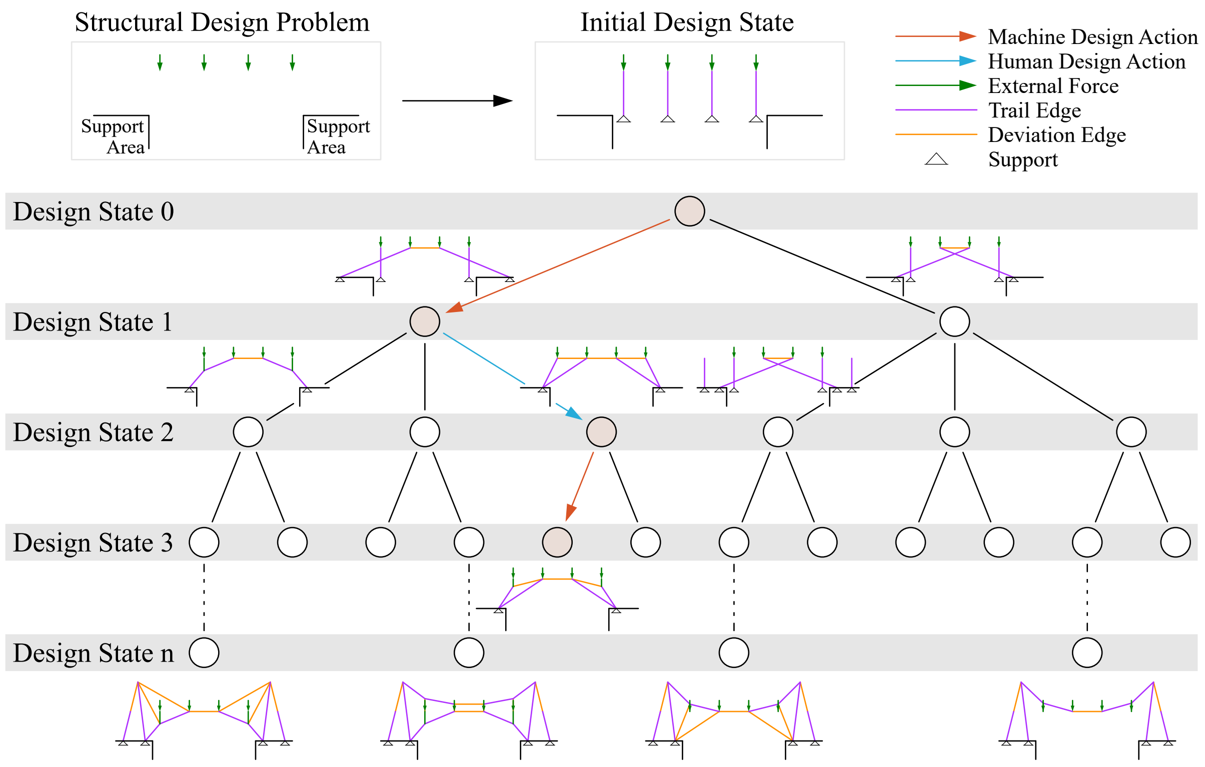

Figure 1: Human-machine collaborative conceptual structural design through a graph-based structural grammar. Human designers and machine RL agents can both execute design actions that modify the current design state.

2. Research Context

2.1. Grammar-based Structural Design

A structural grammar consists of a set of rules that build the structure from the bottom up. Individual rules from the grammar that add to or modify the structure are applied in series resulting in a single complete structure. Structural grammars are a type of shape grammar of which the rules are explicitly informed by engineering knowledge (e.g. equilibrium). Mueller [13] introduced one of the first cross-typological structural grammars, i.e. a structural grammar capable of generating structures of various typologies using the same ruleset, and applied an interactive evolutionary algorithm to a structural grammar for 2D bridge designs. The interactive evolutionary framework optimizes the order in which these rules are applied for structural performance while at the same time allowing a human designer to guide this process in other directions by selecting preferred candidate solutions.

A more general structural grammar that is not limited to 2D bridges has been introduced by Lee et al. [14] as well as Mirtsopoulos and Fivet [15]. However, unlike the context-specific structural grammars, more general structural grammars can also more easily generate designs that look unstructured. This makes the exploration of the design space challenging, and the efficacy of an interactive evolutionary framework combined with non-context-specific structural grammars has yet to be demonstrated.

2.2. Reinforcement Learning in Structural Design

Reinforcement learning is a subfield of ML concerning algorithms that learn to optimally navigate within an environment, formalized as an MDP [16]. An MDP is a system consisting of a set of states and actions, in which actions make an agent move from one state to another. On top of this, these state transitions are associated with rewards that quantify the associated obtained result, and agents are trained to maximize the long-term reward as they progress through the environment. Unlike other ML methods, RL algorithms generate solutions in an inherently step-wise fashion, continually moving from state to state. This naturally creates room for human-machine interaction by giving human designers ways to interact with intermediate states/solution steps.

Some applications of Deep RL to conceptual structural design already exist: Tam et al. solves various inverse design tasks involving reticulated shell structures using Graph Neural Networks (GNN) [17, 18], and Hayashi et al. used Deep RL to learn simplified binary truss optimization problems involving incremental member subtraction [19, 20] and to plan structural assembly [21]. An application of Deep RL to cross-typological structural design has not yet been made in part due to the lack of a structural grammar compatible with Deep RL algorithms.

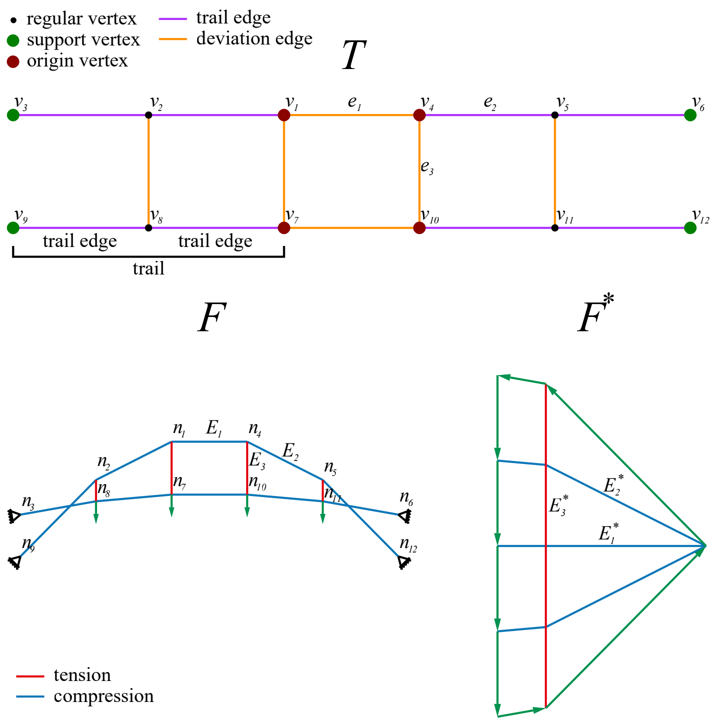

Figure 2: Overview of a topology diagram graph of the CEM (T) and corresponding form (F) and force diagram (F^*). Every vertex in T (v_i) has a corresponding node in F (n_i) [22].

3. Graph-based Structural Grammar

Formulating a structural grammar as an MDP involves formalizing intermediate structural design steps as states of the MDP. To ensure the ability for Deep RL algorithms to interact with the structural grammar, MDP states need to be encoded in such a way that all states of the MDP can be interpreted by the same neural network. This is challenging due to the fact that when designing structures with a structural grammar, the design generally grows in size and complexity as actions are being applied. Standard neural networks present a limitation in this sense, always requiring a constant-sized input to process. We overcome this limitation by encoding the states of our MDP as graphs and consequently allowing these states to be processed by Graph Neural Networks (GNN). In contrast to conventional neural networks, a single GNN model can process graphs of varying size and topology.

Encoding discrete structures as graphs can naturally be done by directly mirroring the connectivity of the structure in which edges of the graph encoding represent members of the structure. Graphs however do not naturally possess any geometrical qualities and there are various potential strategies for storing the geometry of the structure it represents in the graph. A naive method for encoding the geometrical information of the structure is by embedding the spatial coordinates of each node in the corresponding vertex of the graph. Such an encoding often contains redundant information and is not invariant to translation operations that do not affect the structure it encodes.

The structural grammar presented in this work instead encodes structures as input graphs of the CEM form-finding method [9, 10]. The input graph of the CEM, referred to as a topology diagram, consists of edges belonging to one of two classes: trail edges and deviation edges (depicted as purple and orange lines respectively in Figure 2. Trail edges embed information related to the member lengths (λ) while deviation edges contain information regarding their internal forces (μ). These trail lengths and deviation forces are design parameters of the CEM and varying them allows the generation of various equilibrium structures. It has already been shown that GNNs are an effective way of interpreting CEM topology diagrams [23, 22]. Basing the state encoding of the structural grammar on the topology diagram allows a data-efficient encoding of which the corresponding equilibrium structure can continuously be retrieved through the form-finding algorithm of the CEM.

3.1. Ruleset

The introduced structural grammar has a ruleset of four distinct actions (Figure 3):

Rule 1: Update Metric Value operates at the edge level and updates the metric value associated with either a trail edge or a deviation edge. In the case of a trail edge the associate trail length λ is increased by a parameter m. When the rule is applied to a deviation edge the same parameter m is added to the associated deviation force μ instead. Rule 1 only affects the geometry of the structure by changing the metric input parameters of the CEM. This rule can also be applied to underlying deviation edges for which $\mu = 0$, effectively adding new deviation edges to the topology diagram.

Rule 2: Split Trail Edge also operates at the edge level by splitting a given edge in two connected edges. To ensure retained validity of the topology diagram it can only be applied to trail edges and the trail length of the original edge λ is applied to both new trail edges. Applying Rule 2 adds an additional node to which subsequent rules can be applied.

Rule 3: Create Trail allows the creation of an additional trail consisting of one trail edge of length λ. The newly created trail is initially unconnected to the rest of the structure but can be connected through subsequent application of Rule 1.

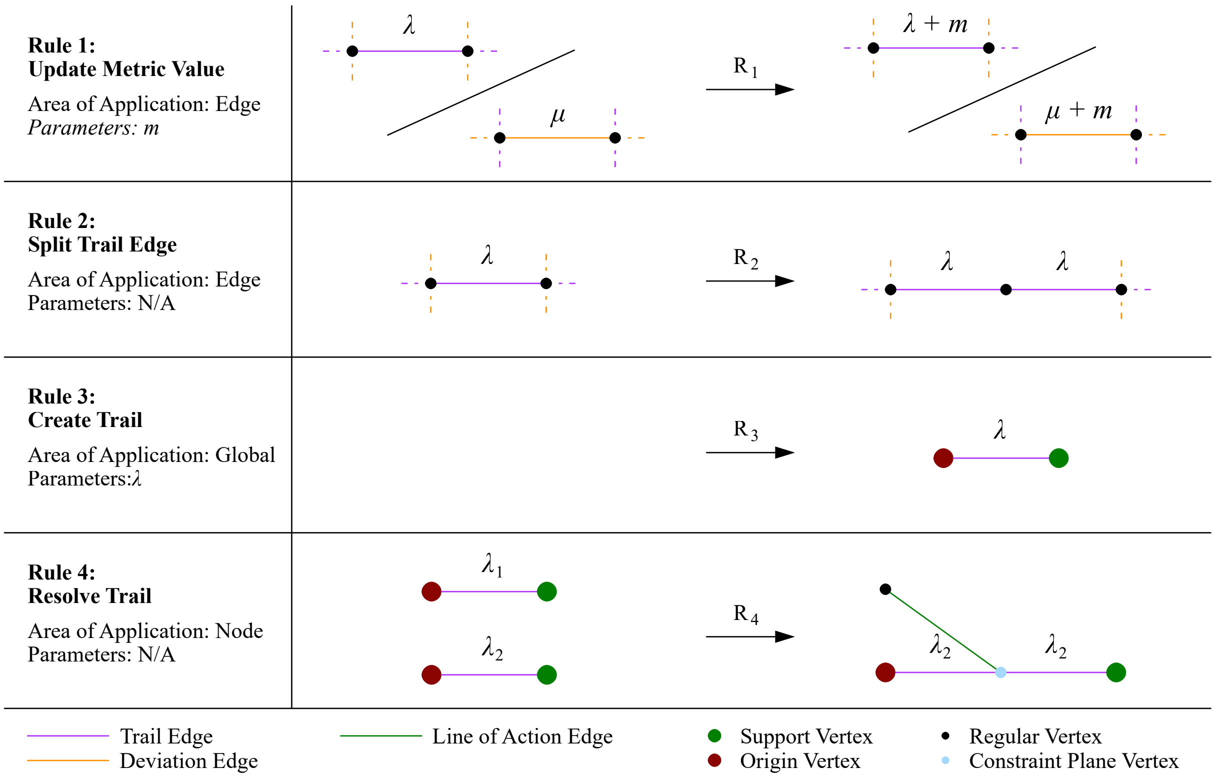

Figure 3: Ruleset of the graph-based structural grammar consisting of 1: Update Metric Value, 2: Split Trail Edge, 3: Create Trail, and 4: Resolve Trail. Updated metric values are trail lengths (λ) and deviation forces (μ).

Rule 4: Resolve Trail is an action that targets the origin node at the end of a trail and connects it directly to the closest intersecting trail in the equilibrium structure. In this way, a trail is resolved, and any load on the resolved trail is guaranteed to stay on its line of action through the application of a constraint plane to the node of the trail it is connected to.

3.2. Rewards

Training an RL agent to use the defined ruleset requires giving it an objective by supplying it with rewards: numerical scores associated with specific state transitions. The simplest possible mechanism for determining the reward (R) of the proposed MDP consists of two parts, reflected by the two terms in Equation 1. The first term is the total load path [24] given by the sum over all edges of the product of their member length (L_i) and their internal force magnitude (N_i). Due to the nature of the CEM, intermediate states can contain structures in which supports are not intersecting with permissible support locations of the design problem. The second term of the reward equation represents a penalty for such invalid supports. To accurately penalize larger reaction forces and worse support locations it is given by the product of the reaction force magnitude ($F_{support,i}$) and the distance to the closest permissible support location (d_i) squared. Both terms are negative to comply with the maximization convention of RL and are weighted by a hyperparameter α ($0 < \alpha < 1$). The computed reward can be applied just to the final state transition of an episode, rewarding the agent for the final design. Alternatively, to help steer the learning process, intermediate state transitions can be associated with non-zero rewards equal to the delta of Equation 1 of the state before- and after the transition. For specific applications in which additional quantitative objectives are desired, terms can be added to the reward equation and adjusted accordingly.

$$R = -\alpha \sum_{i=0}^{n_{\text{edges}}} L_i |N_i| - (1-\alpha) \sum_{i=0}^{n_{\text{supports}}} |F_{support,i}| d_i^2 \tag{1}$$

3.3. Case study examples

Figure 4 shows an example of a solution generation process using the proposed structural grammar. The problem is defined by four loads and two horizontal black lines depicting the permissible support area. At every step, a rule is applied to the topology diagram after which the equilibrium structure is generated with the CEM form-finding algorithm and visualized for the user. In this variant trails connected to support nodes are always extended to intersect with the permissible support area or, in case there is no intersection, extended to minimize the distance between the support and the permissible support area. Minimizing this distance ensures a minimal penalty to the reward obtained for each state transition from Equation 1.

Figure 5 shows a possible continuation of design actions taking as a starting point the arch generated through the rule application sequence from Figure 4. Subsequent application of rules allows the transformation of the original arch structure to a hybrid cable-stayed structure, demonstrating the ability of the structural grammar to generate structures of various typologies.

A final potential continuation from the last design state of this state sequence is shown in Figure 6 in which additional compression struts are added connected to the arch that push the backstay cables inwards.

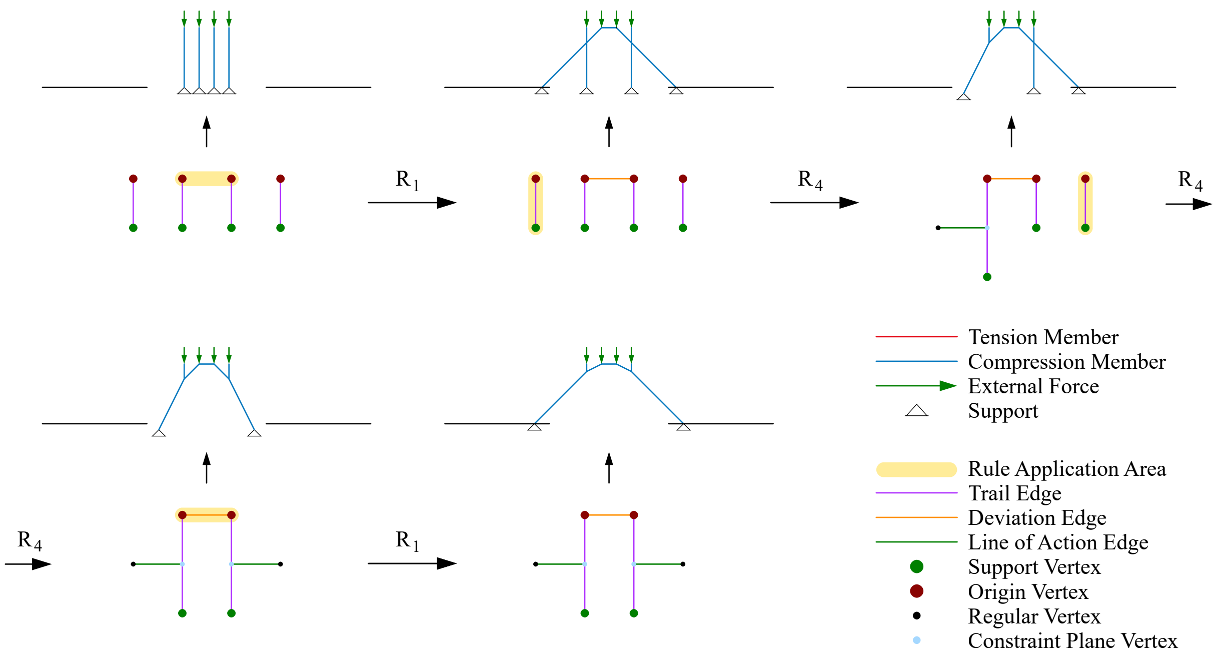

Figure 4: Sequence of structural grammar rule application for generating an arch structure.

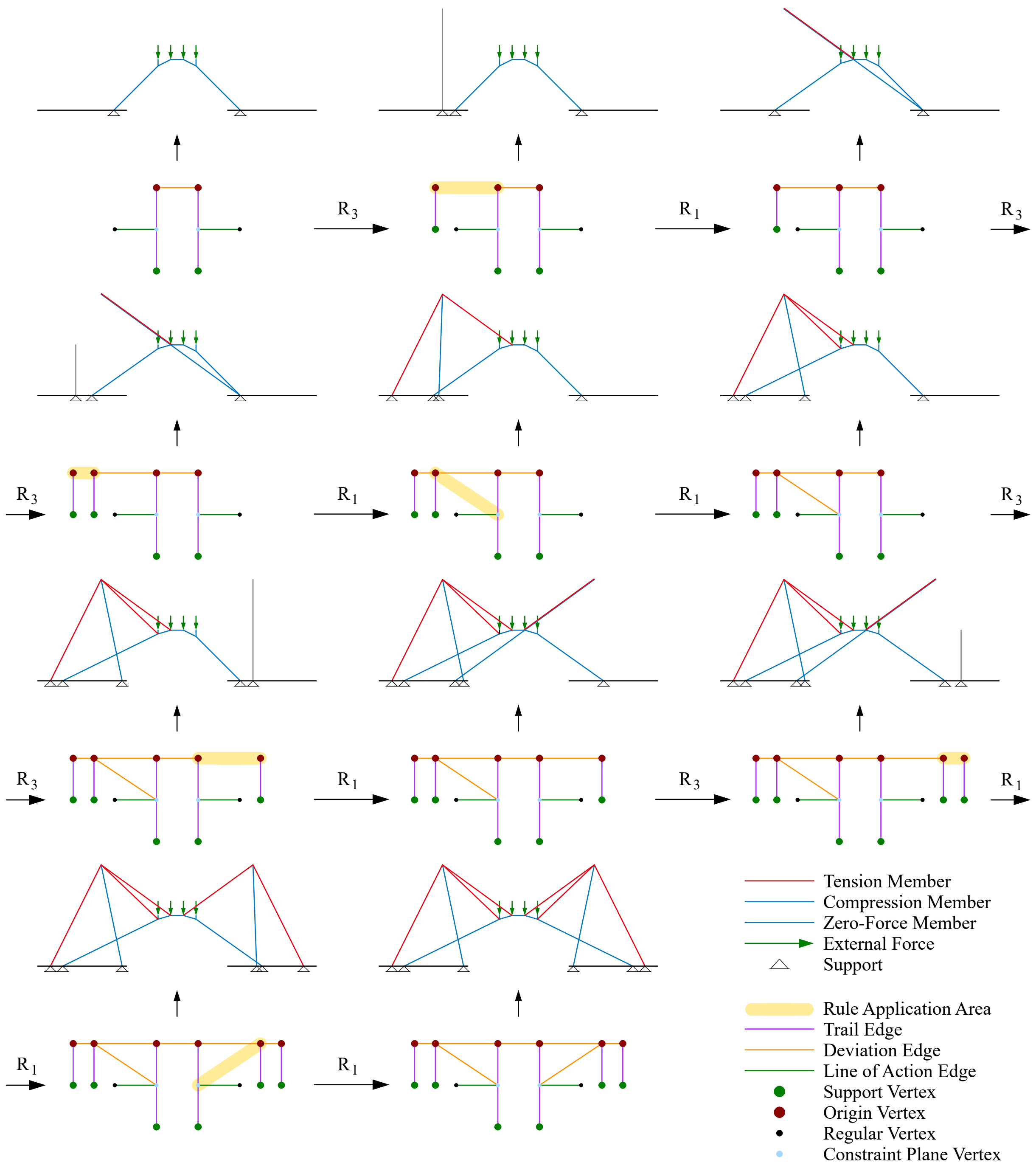

Figure 5: Sequence of structural grammar rule application for transforming an arch structure into a hybrid cable-stayed structure.

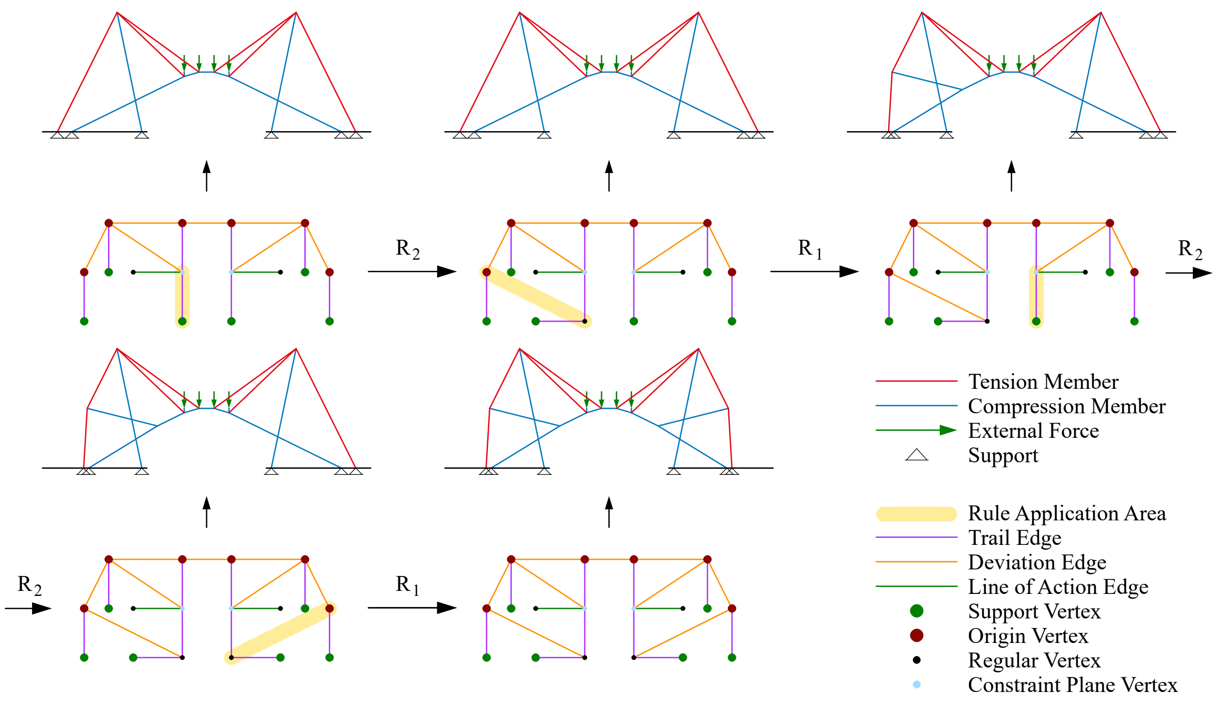

Figure 6: Sequence of structural grammar rule application for modifying a hybrid cable-stayed structure.

4. Conclusion

This paper presented a cross-typological structural grammar based on CEM for conceptual structural design. By encoding design states as graphs we can formalize the structural grammar as an MDP through which a GNN-based Deep RL agent can interact. We highlighted the advantages of human-machine collaborative design in incremental design frameworks facilitating a more interactive exploration of the design space. The case studies presented illustrate the practical application of this methodology, demonstrating its effectiveness in generating topologically diverse equilibrium structures. Future work will focus on further training an RL agent on the MDP of the structural grammar and integrating it within a human-machine collaborative structural design framework.

References

[1] S. Kaethner and J. Burridge, "Embodied CO2 of structural frames," *Structural Engineer*, vol. 90, no. 5, pp. 33–40, 2012.

[2] B. C. Paulson, "Designing to reduce construction costs," *Journal of the Construction Division*, vol. 102, no. 4, pp. 587–592, 1976.

[3] H. Sun, H. V. Burton, and H. Huang, "Machine learning applications for building structural design and performance assessment: State-of-the-art review," *Journal of Building Engineering*, vol. 33, p. 101 816, 2021.

[4] C. Málaga-Chuquitaype, "Machine learning in structural design: An opinionated review," *Frontiers in Built Environment*, vol. 8, 2022.

[5] H. W. J. Rittel and M. M. Webber, "Dilemmas in a general theory of planning," *Policy Sciences*, vol. 4, no. 2, pp. 155–169, 1973.

[6] B. Lawson, *How Designers Think – The Design Process Demystified*. University Press, Cambridge, 2006.

[7] K. Saldaña Ochoa, P. O. Ohlbrock, P. D'Acunto, and V. Moosavi, "Beyond typologies, beyond optimization: Exploring novel structural forms at the interface of human and machine intelligence," *International Journal of Architectural Computing*, vol. 19, no. 3, pp. 466–490, 2021.

[8] Z. Guo, K. Saldana Ochoa, and P. D'Acunto, "Enhancing structural form-finding through a text-based ai engine coupled with computational graphic statics," in *Innovation, Sustainability and Legacy*, Beijing, China: International Association for Shell and Spatial Structures, 2022.

[9] P. O. Ohlbrock and P. D'Acunto, "A computer-aided approach to equilibrium design based on graphic statics and combinatorial variations," *Computer-Aided Design*, vol. 121, p. 102 802, 2020.

[10] R. Pastrana, P. O. Ohlbrock, T. Oberbichler, P. D'Acunto, and S. Parascho, "Constrained form-finding of tension–compression structures using automatic differentiation," *Computer-Aided Design*, vol. 155, p. 103 435, 2023.

[11] P. D'Acunto, J.-P. Jasienski, P. O. Ohlbrock, C. Fivet, J. Schwartz, and D. Zastavni, "Vector-based 3D graphic statics: A framework for the design of spatial structures based on the relation between form and forces," *International Journal of Solids and Structures*, vol. 167, pp. 58–70, 2019.

[12] J.-P. Jasienski, Y. Shen, P. O. Ohlbrock, D. Zastavni, and P. D'Acunto, "A computational implementation of vector-based 3d graphic statics (vgs) for interactive and real-time structural design," *Computer-Aided Design*, vol. 171, p. 103 695, 2024.

[13] C. T. Mueller, "Computational exploration of the structural design space," Ph.D. dissertation, MIT, 2014.

[14] J. Lee, C. Mueller, and C. Fivet, "Automatic generation of diverse equilibrium structures through shape grammars and graphic statics," *International Journal of Space Structures*, vol. 31, pp. 146–163, 2-4 2016.

[15] I. Mirtsopoulos and C. Fivet, "Design space exploration through force-based grammar rule," *archiDOCT*, vol. 8, pp. 50–64, 1 2020.

[16] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[17] K.-M. M. Tam, D. Kudenko, M. Khosla, T. Van Mele, and P. Block, "Performance-informed pattern modification of reticulated equilibrium shell structures using rules-based graphic statics, cw networks and reinforcement learning," in *Innovation, Sustainability and Legacy*, Beijing, China: International Association for Shell and Spatial Structures, 2022.

[18] K.-M. M. Tam, R. M. Avelino, T. Van Mele, and P. Block, "Well-conditioned ai-assisted submatrix selection for numerically stable constrained form-finding of reticulated shells using geometric deep q-learning," *Meccanica*, 2024.

[19] K. Hayashi and M. Ohsaki, "Reinforcement learning and graph embedding for binary truss topology optimization under stress and displacement constraints," *Frontiers in Built Environment*, vol. 6, 2020.

[20] K. Hayashi and M. Ohsaki, "Graph-based reinforcement learning for discrete cross-section optimization of planar steel frames," *Advanced Engineering Informatics*, vol. 51, p. 101 512, 2022.

[21] K. Hayashi, M. Ohsaki, and M. Kotera, "Assembly sequence optimization of spatial trusses using graph embedding and reinforcement learning," *Journal of the International Association for Shell and Spatial Structures*, vol. 63, no. 4, pp. 232–240, 2022.

[22] L. Bleker, K.-M. M. Tam, and P. D'Acunto, "Logic-informed graph neural networks for structural form-finding," *Advanced Engineering Informatics*, vol. 61, p. 102 510, 2024.

[23] L. Bleker, R. Pastrana, P. O. Ohlbrock, and P. D'Acunto, "Structural form-finding enhanced by graph neural networks," in *Towards Radical Regeneration*, C. Gengnagel, O. Baverel, G. Betti, M. Popescu, M. R. Thomsen, and J. Wurm, Eds., Berlin, Germany: Springer International Publishing, 2022, pp. 24–35.

[24] P. O. Ohlbrock, P. D'Acunto, J.-P. Jasienski, and C. Fivet, "Vector-based 3d graphic statics (part iii): Designing with combinatorial equilibrium modelling," in *Spatial Structures in the 21st Century*, Tokyo, Japan: International Association for Shell and Spatial Structures, 2016.