Proceedings of the IASS Annual Symposium 2019 – Structural Membranes 2019
Form and Force
7 – 10 October 2019, Barcelona, Spain
C. Lázaro, K.-U. Bletzinger, E. Oñate (eds.)

Rationalization of layout optimization result by updating discretization of the design domain

Qingpeng LI*, Paul SHEPHERD[a], Matthew GILBERT[b], Linwei HE[b]

* Department of Architecture & Civil Engineering, University of Bath, Bath BA2 7AY, UK
hitssrclqp@gmail.com

[a] Department of Architecture & Civil Engineering, University of Bath
[b] Department of Civil and Structural Engineering, University of Sheffield

Abstract

Truss layout optimization provides a means of identifying the global optimal arrangement of truss bars capable of transmitting a given load or loads to defined support points within a defined design domain. However, the solutions obtained are generally complex and lead to structures with far too many members to be practical, especially when fine discretization of the design domain is employed. In this paper, a heuristic approach to the practical rationalization of optimal layouts is proposed, based on the observation that fan-like segments of bicycle-wheel structures often appear in multiple places in an optimal layout, with a central 'hub' joint of high valence linking through 'spokes' to a curved 'rim'. In this proposed method, the global optimum layout of a given problem is first obtained, and joints that are not used in the optimal truss are removed from the design domain. Subsequently, 'hub' joints of high valence are identified, as are the 'rim' joints they are connected to. A new discretization of the design domain is then produced, which reduces the density of joints in the curved 'rim'. A second layout optimization is then conducted to generate a more rational and buildable truss, with only a small increase in structural weight compared to the global optimal. Using two case-studies, the effectiveness and the efficiency of the proposed approach is validated.

Keywords: Rationalization, layout optimization, optimal layout, discretization, design domain

1. Introduction

Truss layout optimization provides a computationally efficient means of identifying the global optimal arrangement of truss bars capable of transmitting a given load or loads to defined support points within a design domain. Among others, 'ground structure' based layout optimization methods are used widely. The 'ground structure' layout optimization procedure was first proposed by Dorn et al. [1]. More recently, Gilbert and Tyas [2] made it more efficient by proposing an adaptive 'member adding' algorithm, which means that much larger scale layout optimization problems can be solved. Based on the proposed method, a Grasshopper plugin was subsequently developed to make the procedure accessible to those in practice [3].

The solutions obtained are provably globally optimal, but are generally complex and lead to structures with far too many members to be practical, especially when a fine discretization of the design domain is employed. Others have made good progress in employing integer programming methods to address some of these issues [4][5], but these techniques are computationally expensive and therefore not capable of tackling very large problems. Therefore, an efficient post-processing rationalization method, which can consider practical constraints and is suitable for computationally larger problems, is needed urgently.

From the authors' experience of applying these techniques to many case-studies, they have observed that fan-like segments of bicycle-wheel structures often appear in multiple places in an optimal layout, with a central 'hub' joint of high valence linking through 'spokes' to a curved 'rim'. Based on this interesting observation, a heuristic approach to the practical rationalization of optimal layouts is proposed.

2. Methodology

In this section, the layout optimization method of truss structures and the resulting Grasshopper plugin are briefly introduced. Subsequently, the rationalization method for optimal layouts, which updates the discretization of the design domain is proposed.

2.1. Layout Optimization Method and the Grasshopper Plug-in

The standard layout optimization process involves four steps, as shown in Figure 1. Firstly the design domain, load and support conditions are specified, Figure 1(a). Secondly, the design domain is discretized using nodes, Figure 1(b). Thirdly, these nodes are interconnected with all potential members to create a 'ground structure', Figure 1(c). This 'ground structure' normally contains a vast number of potential structural forms, even with medium number of nodes (e.g., 100~1000 nodes).

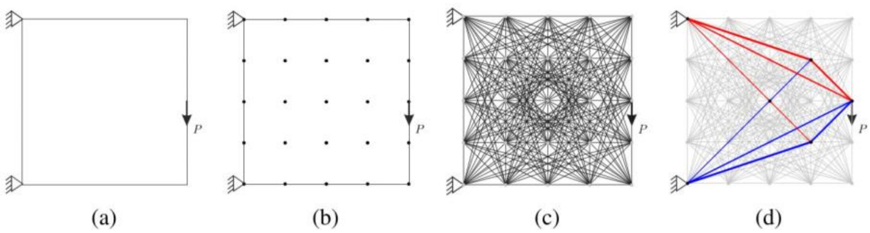


Figure 1: Steps in layout optimization: (a) specify design domain, loads and supports; (b) discretize domain using nodes; (c) interconnect nodes with potential truss members; (d) use optimization to identify the optimal structural layout.

The most efficient structural layout is then identified (Figure 1(d)) by solving the optimization problem below:

$$\begin{aligned} &\underset{\mathbf{a},\mathbf{q}}{\text{Minimize}} \quad V = \mathbf{l}^{\mathrm{T}}\mathbf{a} \\ &\text{Subject to} \\ &\qquad \left.\begin{aligned} &\mathbf{B}\mathbf{q}^{(k)} = \mathbf{f}^{(k)} \\ &\mathbf{q}^{(k)} \geq -\sigma^{-}\mathbf{a} \\ &\mathbf{q}^{(k)} \leq \sigma^{+}\mathbf{a} \end{aligned}\right\} \text{for } k = 1, 2, ..., p \\ &\qquad \mathbf{a} \geq \mathbf{0}, \end{aligned} \tag{1}$$

where, $\boldsymbol{V}$ is the structural volume, $\mathbf{a} = [a_1, a_2, ..., a_m]^{\mathrm{T}}$ is a vector containing member cross-sectional areas, with m denoting the number of members. $\mathbf{l} = [l_1, l_2, ..., l_m]^{\mathrm{T}}$ is a vector of member lengths. $\boldsymbol{B}$ is a $2n \times m$ equilibrium matrix comprising direction cosines, with n denoting the number of nodes. k is the load case identifier and p is the number of load cases. $\mathbf{q}^{(k)}$ and $\mathbf{f}^{(k)}$ are vectors containing the internal member forces and the external forces in the k^{th} load case, respectively. σ^{+} and σ^{-} are limiting tensile and compressive stresses respectively.

Using this 'ground structure' approach, the number of members increases with the square of the number of nodes, which leads to very large-scale optimization problems. Fortunately, problem (1) is a linear programming (LP) problem that can be solved very efficiently via modern LP solvers utilizing an interior-point method, e.g., MOSEK. Moreover, an adaptive solution scheme can be adopted [2] to decompose the problem into a number of sub-problems that can be tackled relatively easily.

Based on the proposed approach by Gilbert and Tyas [2], a conceptual design optimization tool [5] was developed for Rhinoceros-Grasshopper [6]. Figure 2 shows an example of the components in use, where they can be grouped into the following types:

1) Geometry Definition. The geometry of the design domain is defined using standard Grasshopper components so as to provide a parametric workflow. Users are free to define a design space in terms of lines, polygons, NURBS surfaces and complex BReps. The geometry is then meshed to faces and vertices as input for the design domain. A number of bespoke components have been provided to aid this process for layout optimization.

2) Design Domain. A number of components are then used to assign material properties such as tensile and compressive strength, Young's modulus, Passion's ratio, and material density. The support and load conditions need to be defined.

3) Layout Optimization. This component performs the layout optimization and provides diagnostic information on the time and number of iterations needed to solve. The volume of material required by the structure is also reported.

4) Visualization of solutions. A component is provided to visually present the resulting members, joint positions, internal forces and cross-sectional areas.

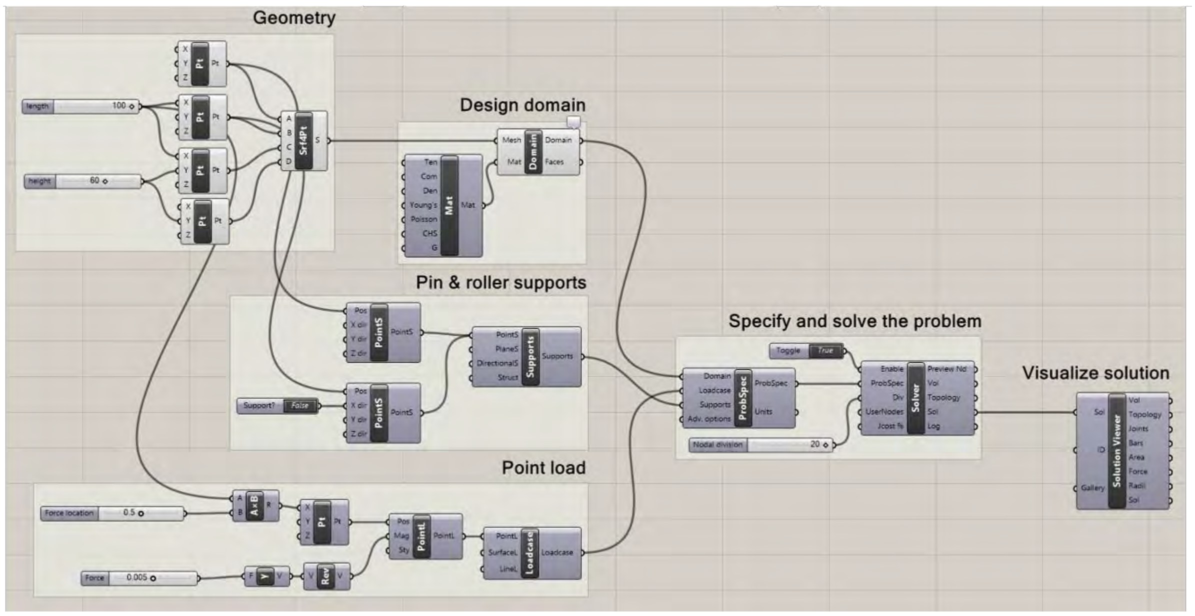
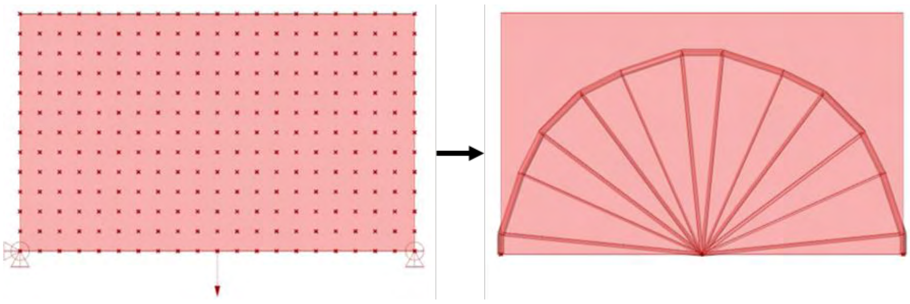


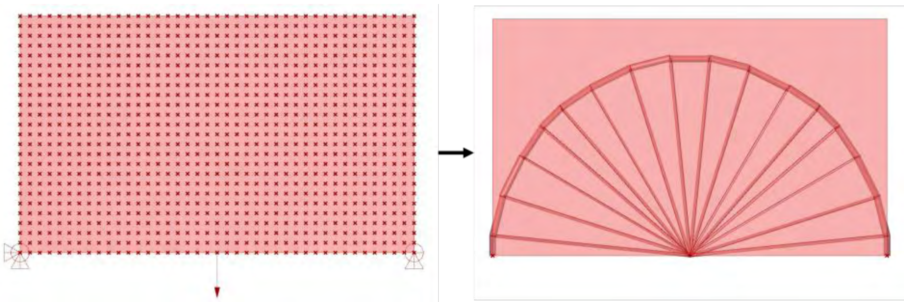
Figure 2: The Grasshopper layout optimization plugin.

2.2. The Rationalization Method

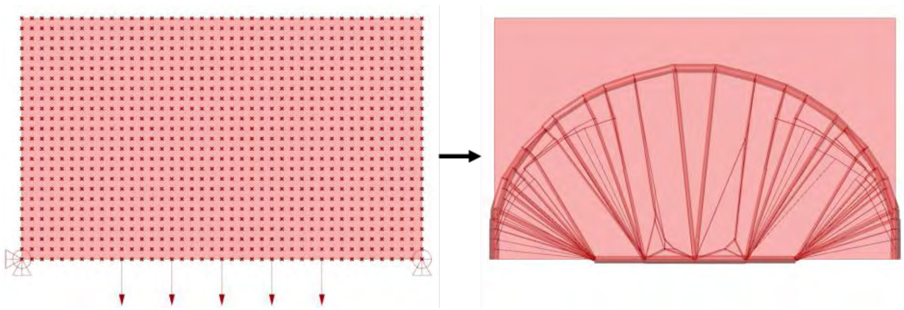
Figure 3(a) shows the optimal layout of a simply supported truss under one concentrated load. It can be observed that there appear fan-like segments of bicycle-wheel structures in the optimal layout, with a central 'hub' joint of high valence linking through 'spokes' to a curved 'rim'. Moreover, the resulting truss structure would be too complex to build, since too many members are linked to the same central joint, which would not be practical to manufacture. When a finer discretization of the design domain is employed, as in Figure 3(b), the number of 'spoke' members increases further, and when the concentrated load is distributed over multiple joints, as in Figure 3(c), the manufacturing complexity of the optimal solution become even more prohibitive.



(a) The simply supported truss under one concentrated load with a coarse mesh.



(b) The simply supported truss under one concentrated load with a finer mesh.



(c) A simply supported truss under five equally-spaced concentrated loads.

Figure 3: Numerical examples.

Base on the observations above, a rationalization approach have been developed. As the optimal layout of the truss can represent the load path of the applied load within the given design domain (ignoring multiple load cases for now), the basic assumption is that a rational and buildable truss exists close to the unbuildable optimal layout. By considering simplifications of the optimal layout, a practical truss using a sub-set of joints from the optimal layout is found. Thus the goal of the rationalization approach becomes a sensible selection of joints from the optimal layout. Considering the main contribution to complexity of the optimal layout results from the fan-like segments of bicycle-wheel structures, then a rationalization approach that reduces the density of the joints used in the curved 'rim' seems a sensible first step.

A numerical example, which is of a simply supported truss under three concentrated loads, is chosen here to describe the rationalization approach:

1) The optimal layout of the truss is obtained by conducting layout optimization within the given design domain using the Grasshopper plugin, shown in Figure 4. For later benchmarking, the structural volume of this optimal layout is 1239500 units^3.

2) Joints which are not used in the optimal layout are removed from the design domain.

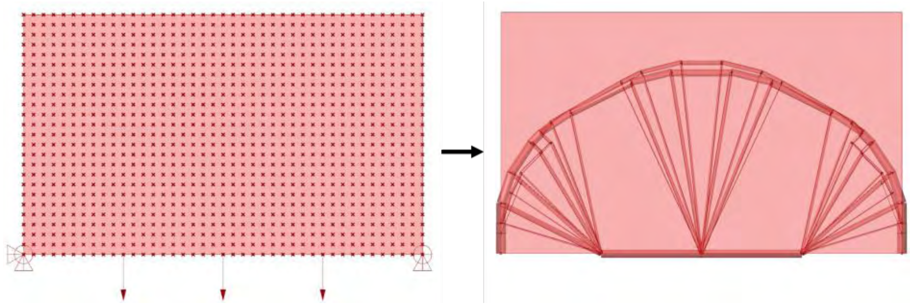


Figure 4: 1st layout optimization.

3) Potential ‘hubs’, which denote joints with high valence, are identified by counting the number of bars connected to each joint. Figure 5 shows three ‘hub’ joints identified in the optimal layout, which are joints where the concentrated loads are applied.

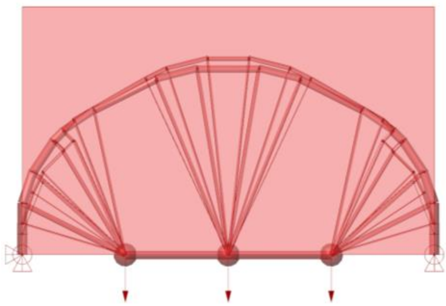


Figure 5: Three identified ‘hub’ joints.

4) Joints and relevant bars which are connected to each potential ‘hub’ are ordered by the angle they wind around the ‘hub’, for example Figure 6 shows in green the fifth connected joint/bar of the first ‘hub’ joint. Note that this angle ordering works well for 2D problems, using the best-fit plane as a coordinate system, but a different approach is needed to order the members in 3D.

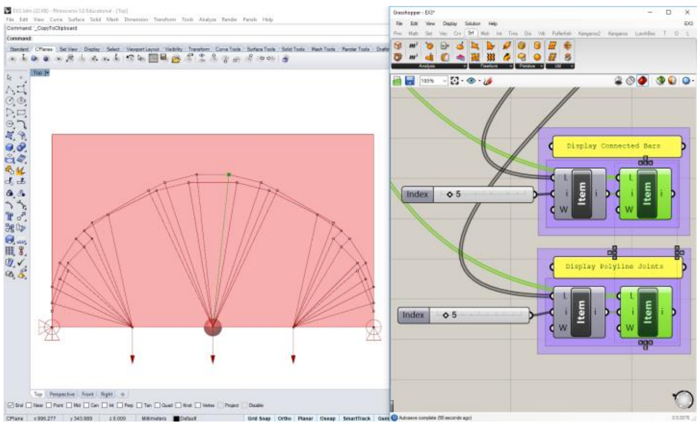


Figure 6: Fifth connected joint and bar of the first ‘hub’ joint.

5) A sub-set of connected joints and bars is created for each ‘hub’ joint, in which lengths of the bars within each group only differ from the length of the previous by ±10%. In this way, a smooth ‘rim’ is identified, ignoring ‘spokes’ which are of a very different length to their neighbours, indicating they might be part of a different ‘rim’ or of another sub-structure altogether.

6) For each 'hub' joint, a polyline is constructed, linking its 'rim' joints together. The polyline is smoothed using standard Grasshopper 'smooth' and 'rebuild' components. If the connected joints form more than one candidate polyline, the geometrically shortest polyline is chosen, representing an arc-like segment of the bicycle wheel. For example, Figure 7 shows the shortest polyline of the central 'hub', where the two connected joints at the ends of the two horizontal members have not been included.

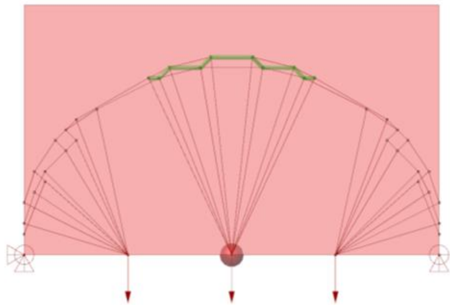


Figure 7: The shortest polyline of the first 'hub' joint.

7) The curved 'rim' polylines are then sub-divided into equal-length segments, resulting in several new joints. This provides a new (more rationalized) discretization of the design domain, as shown in green in Figure 8.

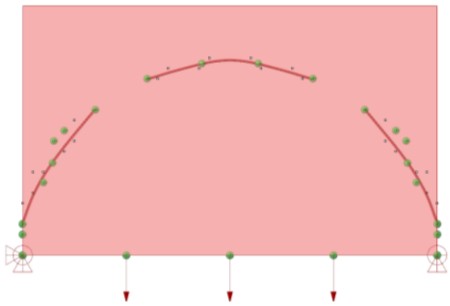


Figure 8: New mesh of the design domain.

8) A second layout optimization is then performed using the new design domain discretization. Figure 9 shows a more rational and buildable truss structure is obtained. Its structural volume is 1258900 units^3, only 1.57% larger than the volume of the original optimal layout shown in Figure 4.

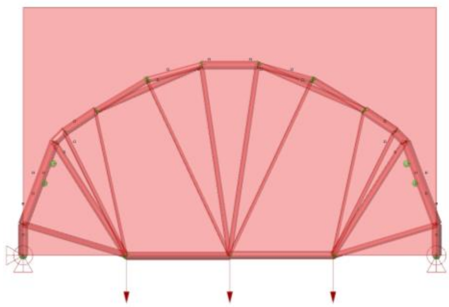


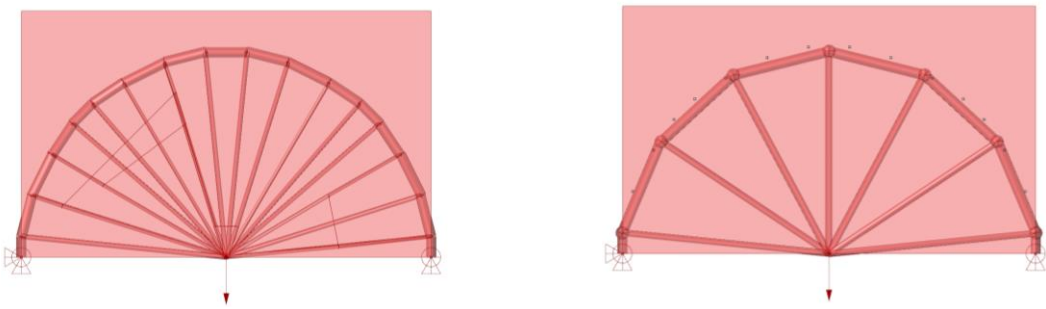
Figure 9: The simpler optimal layout.

Using these eight steps, more rational and buildable trusses can be quickly obtained, with an acceptably small increase in structural weight compared to the global optimal. In this example, it can also be seen that a circular-arc 'rim' appears around the central 'hub', whilst non-circular, curved 'rims' appear around the other two potential 'hubs'.

3. Examples

In this section, two further case-studies are illustrated to validate the effectiveness of the proposed rationalization approach.

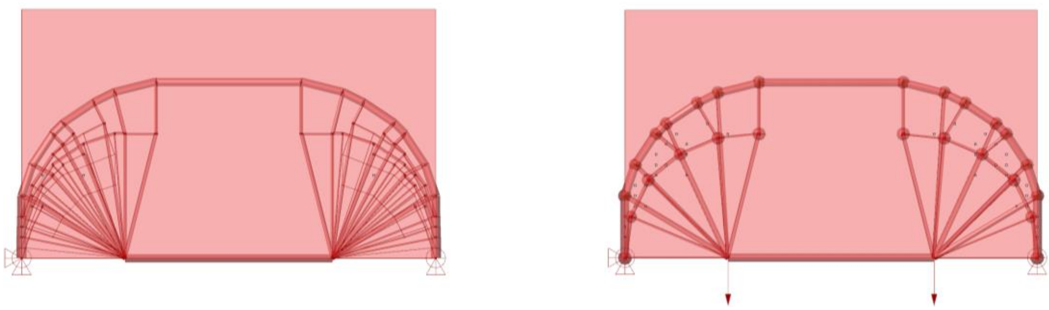
The first example is a simply supported truss under one concentrated load. Shown in Figure 10, it can be seen that one 'hub' joint is identified, and after rationalization, the circular-arc 'rim' is maintained around it. By using the proposed rationalization approach, the structural volume of the rationalized optimal layout is 1602900 units³, which is just 1.57% larger than its original optimal volume of 1578200 units³).



(a) The original optimal layout. (b) The rationalized optimal layout.

Figure 10: The simpler optimal layout.

The second example is a simply supported truss under two concentrated loads. Shown in Figure 11, it can be seen that two potential centres are identified, and circular-arc 'rims' appear around them. By using the proposed rationalization approach, the structural volume of the rationalized optimal layout is increased from 1169600 units³ to 1174000, an increase of only 0.38%.



(a) The original optimal layout. (b) The rationalized optimal layout.

Figure 11: The simpler optimal layout.

4. Conclusions

By updating and simplifying the discretization of the design domain, a heuristic approach to the practical rationalization of optimal layouts is proposed. This approach is a post-processing procedure, which can consider practical constraints and is independent of the layout optimization algorithm used. It produces simplified layouts that are more practical from a fabrication point of view with only a small penalty in terms of increased material use. Moreover, by reducing the density of the joints within the design domain, the computational overhead of this post-processing step is small. However, there are still some limitations of the current rationalization approach. For example, so far, only 2D problems have been studied, and only for a single load case. Heuristics for post-processing of more general truss structures therefore remains the subject of ongoing work by the authors.

5. Acknowledgements

This research was supported by the EPSRC, under grants EP/N023269/1 and EP/N023471/1.

References

[1] W.S. Dorn, R.E. Gomory, H.J. Greenberg. "Automatic design of optimal structures", *Journal de M`ecanique*, vol. 3, pp.25-52, 1964.

[2] M. Gilbert and A. Tyas, "Layout optimization of large-scale pin-jointed frames", *Engineering Computations*, vol. 20, iss. 8, pp.1044-1064, 2003.

[3] L. He, M. Gilbert, P. Shepherd, J. Ye, A. Koronaki, H.E. Fairclough, B. Davison, A. Tyas, J. Gondzio, A.G. Weldeyesus. "A new conceptual design optimization tool for frame structures", in *Creativity in Structural Design: Proceedings of the IASS Annual Symposium 2018*, MIT, Boston, USA, July 16-20, 2018, C. Mueller and S. Adriaenssens Ed.

[4] H. Fairclough, M. Gilbert, C. Thirion and A. Tyas, "Balancing complexity and structural efficiency in the design of optimal trusses", in *Creativity in Structural Design: Proceedings of the IASS Annual Symposium 2018*, MIT, Boston, USA, July 16-20, 2018, C. Mueller and S. Adriaenssens Ed.

[5] L. He and M. Gilbert, "Rationalization of trusses generated via layout optimization", *Structural and Multidisciplinary Optimization*, vol. 52, iss. 4, pp. 677–694, 2015.

[6] McNeel R. Grasshopper generative modeling for Rhino. Computer software (2011b), http://www.grasshopper3d.com. 2010.