IASS 2018 BOSTON, JULY 16-20
CREATIVITY IN STRUCTURAL DESIGN

Proceedings of the IASS Symposium 2018
Creativity in Structural Design
July 16-20, 2018, MIT, Boston, USA
Caitlin Mueller, Sigrid Adriaenssens (eds.)

A new conceptual design optimization tool for frame structures

Linwei HE*, Matthew GILBERT, Paul SHEPHERD[a], Jun YE[a], Antiopi KORONAKI[a], Helen E. FAIRCLOUGH, Buick DAVISON, Andy TYAS, Jacek GONDZIO[b], Alemseged G. WELDEYESUS[b]

*Department of Civil and Structural Engineering, University of Sheffield, UK
Mappin Street, Sheffield, S1 3JD, UK
linwei.he@sheffield.ac.uk

[a] Department of Architecture and Civil Engineering, University of Bath, UK
[b] School of Mathematics, University of Edinburgh, UK

Abstract

Architects and structural engineers are increasingly embracing the use of optimization tools for use in the design of building structures. One tool is numerical layout optimization, which provides a powerful and highly efficient means of generating theoretically optimal lightweight structures. However, the layouts so obtained are often impractical due to their complex forms. In this paper, a means of generating families of more practical solutions for use as inspiration at the conceptual design stage is described. With layout optimization simpler solutions can be generated by running the optimizer with penalization parameters, e.g. to represent the cost of joints. Alternatively, a family of frame designs can be automatically generated by minimizing structural complexity indices (e.g. the total number of members, number of intersecting members at joints, etc) in a post-processing step. This is here shown to provide a pragmatic and computationally efficient means of identifying families of practical, near-optimal, truss designs. In the paper the aforementioned techniques are first briefly outlined; a Rhino-Grasshopper plugin tool incorporating the techniques is then used to generate families of 2D and 3D design concepts, which clearly indicate the promise of the methods involved.

Keywords: layout optimization, conceptual design, frame, gradient-based

1. Introduction

Digital design tools have become popular in the design of modern spatial structures. A number of optimization software modules have also been developed to help engineers identify lightweight structures, sometimes using nature-inspired optimization methods (e.g. genetic algorithms). Since no mathematical derivations (e.g. gradient information) are required, these methods can work well in parametric modelling software. However, the lack of underlying mathematics also leads to computationally expensive processes with local optima identified in general [1]. For spatial structures, an alternative approach is numerical layout optimization, which provides a powerful and highly efficient means of generating lightweight truss structures [2,3]. Using this method, a linear programming (LP) problem is formulated, leading to very efficient optimization processes with global optimum solutions identified in general. In addition, geometry optimization can be performed in a post-processing step to generate textbook-like solutions by solving a sequence of nonlinear programming (NLP) problems, where the sizes of members and the positions of joints are simultaneously optimized [4].

The standard layout and geometry optimization formulations do not include structural complexity measures; therefore, driven by mathematical derivations, the process will normally identify optimal layouts which include complex features, e.g. joints. Thus although structurally efficient, these forms may be impractical to construct due to high associated fabrication costs. To address this, one option is to include additional constraints, e.g. to restrict the number of members via the use of mixed integer linear programming (MILP) methods, as described in [5]. However, these can be computationally expensive to apply, rendering them unsuitable for use at the conceptual design stage. Alternatively, an incremental approach, involving several optimization phases, can be used to drive solutions towards more practical designs. Although this means that a global optimum is no longer guaranteed, various near-optimal solutions can be generated by utilizing a number of controlling parameters, creating a family of layouts which can be used as inspiration at the conceptual design stage. Structural complexity indices considered here include the number of members in the design as well as joint complexity.

The methods described, including basic layout and geometry optimization methods, have been programmed in a Rhino-Grasshopper plugin, providing an interactive modelling environment designed to meet the needs of designers.

The paper is organized as follows: firstly, the truss layout and geometry optimization formulations are outlined. Secondly, structural complexity indices are considered, and methods of including these in the formulation are outlined. Thirdly, the proposed optimization workflow is briefly discussed. Fourthly, a number of design examples are used to demonstrate the efficacy of the proposed methods. Finally, conclusions are drawn.

2. Layout and geometry optimization

Layout and geometry optimization involves a series of steps, as shown in figure 1 (see also [4]). Considering first layout optimization, the basic single load case formulation can be written as follows:

$$\min_{\boldsymbol{a},\boldsymbol{q}} V = \boldsymbol{l}^{\mathrm{T}}\boldsymbol{a}, \tag{1a}$$

subject to:

$$\boldsymbol{B}\boldsymbol{q} = \boldsymbol{f}, \tag{1b}$$

$$-\sigma_0\boldsymbol{a} \leq \boldsymbol{q} \leq \sigma_0\boldsymbol{a}, \tag{1c}$$

$$\boldsymbol{a} \geq 0, \tag{1d}$$

where, V is the volume of the structure, $\boldsymbol{l} = [l_1, l_2, \dots l_m]^{\mathrm{T}}$ and $\boldsymbol{a} = [a_1, a_2, \dots a_m]^{\mathrm{T}}$ are vectors containing member lengths and areas, respectively, with m denoting the number of members. $\boldsymbol{B}$ is a $3n \times m$ equilibrium matrix and $\boldsymbol{f} = [f_{1x}, f_{1y}, f_{1z}, \dots f_{nx}, f_{ny}, f_{nz}]^{\mathrm{T}}$ is a nodal force vector, with n denoting the number of nodes. Problem (1) is a LP problem, which means that solutions can be obtained efficiently. When large numbers of nodes are employed the layouts obtained using (1) will typically be quite complex in form, partly due to the use of a fixed nodal grid in the 'ground structure'. To rationalize these layouts, nodal positions can be subsequently optimized, leading to a geometry optimization problem, figure 1(d), solvable via NLP. Whilst geometry optimization is capable of generating clear, textbook-like, solutions, it does not directly address the issue of structural complexity.

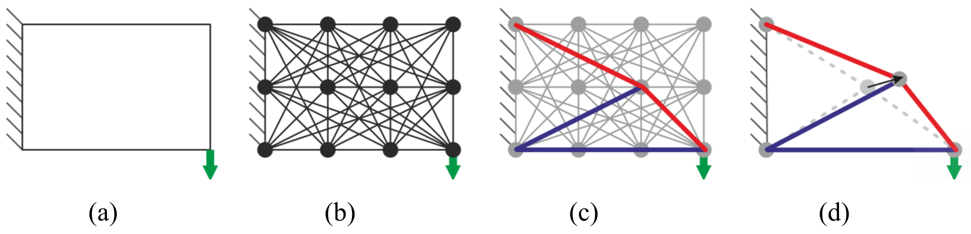


Figure 1: Steps in truss layout and geometry optimization: (a) design domain, load and support conditions specified; (b) nodes interconnected by potential members, forming a 'ground structure'; (c) optimal layout identified by solving underlying layout optimization problem; (d) geometry optimization post-processing step.

3. Addressing structural complexity

Two means of controlling structural complexity are considered here, with a view to increasing the practicality and buildability of the design solutions generated in the optimization process: (i) imposing limits on the total number of members, and (ii) controlling joint complexity (e.g. by limiting the number of members intersecting a joint). With a gradient-based optimization method, this necessitates introduction of additional mathematical terms, as new constraints and/or objective functions. Depending on the mathematical properties of these terms, the resulting formulation can be solved via LP or NLP.

3.1. Joint cost within LP optimization formulation

Joint costs can be introduced in the formulation by making a minor modification to problem (1). Using joint costs, the length vector in objective function (1a) is modified as follows (after [6]):

$$\min_{\boldsymbol{a},\boldsymbol{q}} V = (\boldsymbol{l} + 2\boldsymbol{s})^{\mathrm{T}} \boldsymbol{a}, \tag{2}$$

where $\boldsymbol{s}$ is a predefined coefficient vector containing the cost value of members connected at joints.

3.2. Post-processing via NLP optimization

Alternatively, an NLP problem can be formulated to control structural complexity. Note that gradient-based optimization methods, such as the interior point method [7], require smooth design variables in general. However, structural complexity measurements are normally expressed using non-smooth integer variables (e.g. the number of members). To address this, a smooth Heaviside projection $H(x)$ can be used to project a smooth design variable x to discrete integers 0 or 1 [8]:

$$H(x) = \coth(\mu)\tanh(\mu x), \tag{3}$$

where, μ is a predefined projection factor. As shown in figure 2, variable x is projected to 0 or 1, with μ determining the width of the smooth transition zone. In this paper, μ increases gradually from 2 to 20 to adaptively approximate integers 0 and 1.

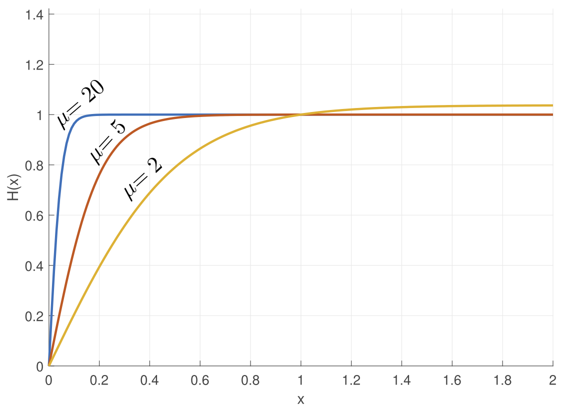


Figure 2. Heaviside projection

Using Heaviside projection (3), the 'existence' of a member i can now be expressed by its cross-section area a_i. Let a_{ref} denote a reference member area, the following is satisfied:

$$H(a_i/a_{\mathrm{ref}}) \approx \begin{cases} 0, & \text{if } a_i \ll a_{\mathrm{ref}} \\ 1, & \text{otherwise} \end{cases}. \tag{4}$$

This means that the total number of members can be minimized by replacing (1a) with:

$$\min_{\boldsymbol{a},\boldsymbol{q}} \Phi_{\mathrm{M}} = \sum_{i=1}^{m} H(a_i/a_{\mathrm{ref}}), \tag{5}$$

where, Φ_{M} is the structural complexity index regarding to the number of members. Similarly, the complexity function Φ_j for the jth joint can be written as:

$$\Phi_j = \sum_{i=1}^{m_j} H(a_{ji}/a_{\mathrm{ref}}), \tag{6}$$

where, m_j is the number of members connected, and a_{ji} is the area of the ith member at joint j. It is worth noting that, if Φ_j is simultaneously reduced for all joints, the objective function becomes $2\Phi_{\mathrm{M}}$, since every member is connected by two joints. However, if only complex joints are considered, e.g. joints with more than $n_{\max}$ connected members, the objective function can be written as:

$$\min_{\boldsymbol{a},\boldsymbol{q}} \Phi_{\mathrm{J}} = \sum_{j=1}^{\tilde{n}} \Phi_j, \tag{5}$$

where, Φ_{J} is the joint complexity index, and $\tilde{n}$ is the number of joints that are considered 'complex', determined by $n_{\max}$ ($n_{\max} = 4$ in this paper). Note that when the original objective function is replaced with a function involving Φ_{M} and/or Φ_{J} then it is necessary to define the acceptable tradeoff on structural efficiency by imposing a volume constraint:

$$\boldsymbol{l}^{\mathrm{T}}\boldsymbol{a} \leq (1+\epsilon)V_0, \tag{6}$$

where ϵ is the specified permitted volume increase ratio and V_0 is the initial volume before taking structural complexity into account.

4. Optimization workflow in an interactive Rhino-Grasshopper environment

The proposed workflow involves several phases. In the first phase the LP problem (1) is solved, allowing identification of a near-optimal layout based on the initial 'ground structure'. Joint costs can be included in the LP phase; due to the vast number of potential layouts that exist in the 'ground structure', inclusion of joint costs can lead to fundamentally different structural forms being generated.

In the second phase, geometry optimization is used to rationalize the layouts generated via LP, whilst in structural complexity can be addressed in the third phase, where the aforementioned NLP simplification methods involving the use of Φ_{M} and/or Φ_{J} are employed.

The use of the Rhino-Grasshopper parametric modelling environment allows designers to adjust the design domain, load and support configurations and then inspect the generated solutions at the conceptual design stage. Since layout optimization is computationally efficient, near-optimum layouts for problems involving up to a few thousand joints can be generated in seconds. This also allows a range of joint costs to be considered in order to produce a range of candidate layouts. Generated solutions can then be sorted by volume and/or structural complexity.

5. Design examples

In an interactive design process, the proposed simplification methods are normally utilized together to deliver practical designs. Nevertheless, in the first example they are used independently to show their various effects. Other examples are then used to demonstrate the efficacy of the proposed simplification techniques.

5.1 Simple MBB beam

The Messerschmidt-Blkow-Blohm (MBB) beam is a classical example used in the structural topology optimization literature [9]. Here the design domain is discretized using a nodal grid comprising 21×11 evenly distributed nodes. Layout and geometry optimization can be used to obtain the rationalized solution shown in figure 3(b). A key feature of the form identified is the fan-type structure radiating out from the loaded joint, which, despite being structurally efficient, is likely to be difficult to manufacture. To address this, the proposed simplification methods are used. With a range of controlling parameters, e.g. s in (2) and ϵ in (6), various solutions are generated, as shown in figures 3(c) - (n) (with parameters in increasing order, from left to right).

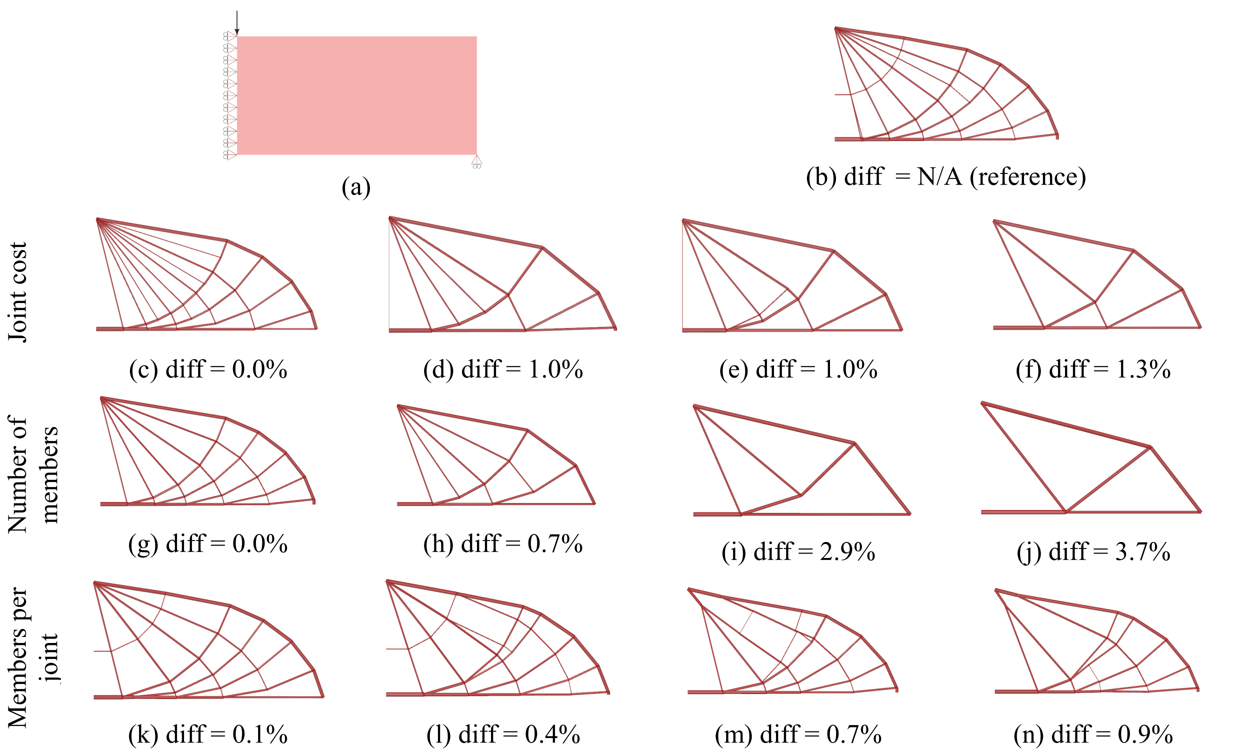


Figure 3. MBB beam example: (a) problem specification; (b) reference solution obtained using layout and geometry optimization against which other solutions are compared; (c) - (f): using joint cost; (g) - (j): reducing number of members, Φ_M; (k) - (n) by reducing the number of members per joint, Φ_J

It can be observed that generated structures are successfully simplified. Using the joint cost and Φ_M, simple layouts are reported when the control parameters are increased. On the other hand, with Φ_J the fan-type structure is simplified without significantly affecting the rest of the structural parts, even if a relatively large volume increase ratio ϵ is used.

5.2 Problems involving UDL and Euler buckling

To further demonstrate the effects of the methods described, consider another simple 2D problem, where a uniformly distributed load (UDL) is applied downwards at the base of the design domain, with pin and roller supports located at the two end points. In addition, Euler buckling is included in the geometry optimization phase to control the lengths of compression members [10]. Using the proposed simplification techniques various solutions are generated (figure 4). Although tensile bracing members between the arch and bottom chord are arranged differently, the structural forms generated are all quite similar. This is because, when the problem is relatively simple, gradient-based optimization methods are prone to move from different starting points towards the same local (or global) optimum.

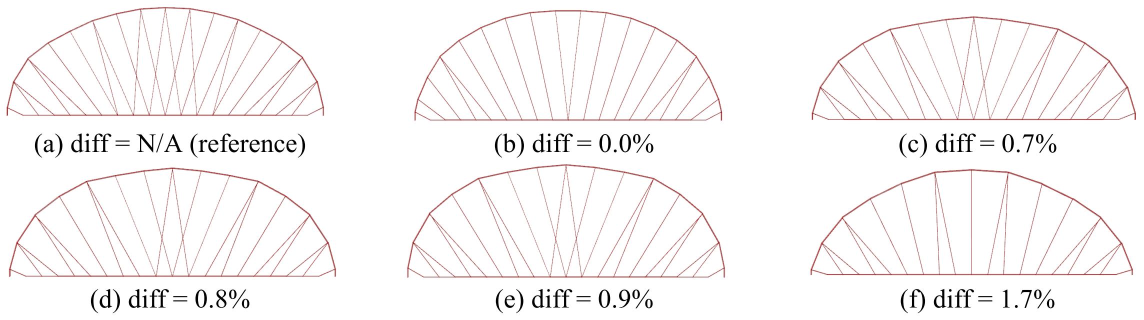


Figure 4. Various solutions generated in the 2D case under UDL (incl. Euler buckling) : (a) reference solution; (b) - (f), simplified solutions, with parameters (i.e. s and ϵ) progressively increased

For more complex design cases, the generated layouts can vary significantly when input parameters are varied (e.g. specified joint cost). For example, consider the 3D roof design example shown in figure 5. The structure is loaded in two regions, as shown in Figure 5 (a). The solution without simplification, shown on figure 5 (b), comprises a curved cantilever structure connecting the loaded points via underslung trusses. Whilst the layout in figure 5 (c) resembles the same form, the solution in figure 5 (e) does not have the supporting trusses, leading to a 31.5% increase of volume.

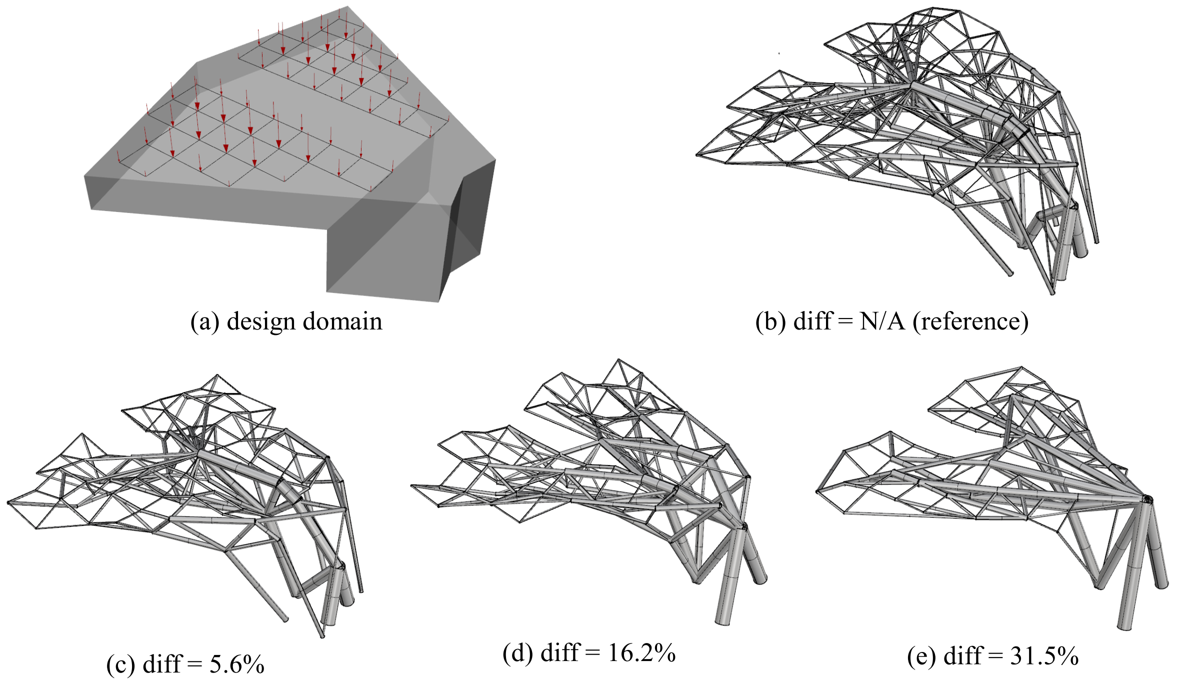
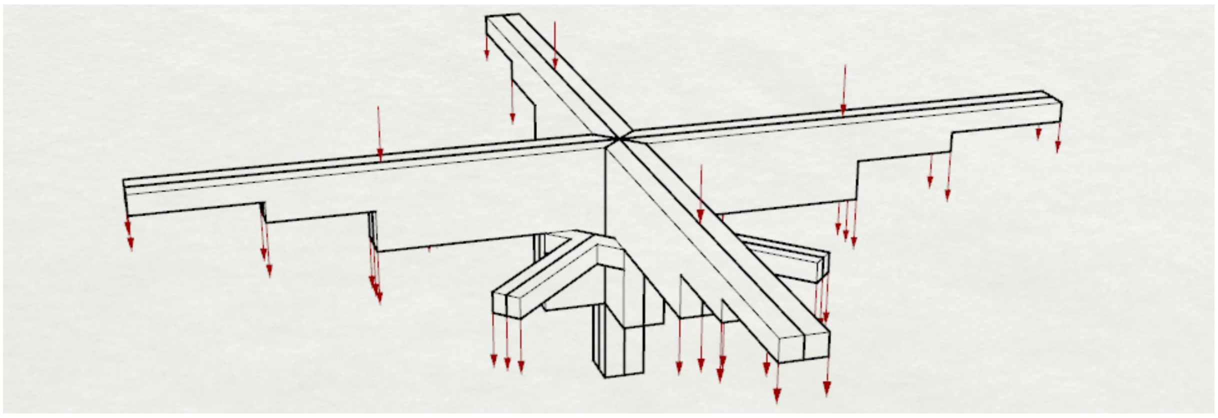
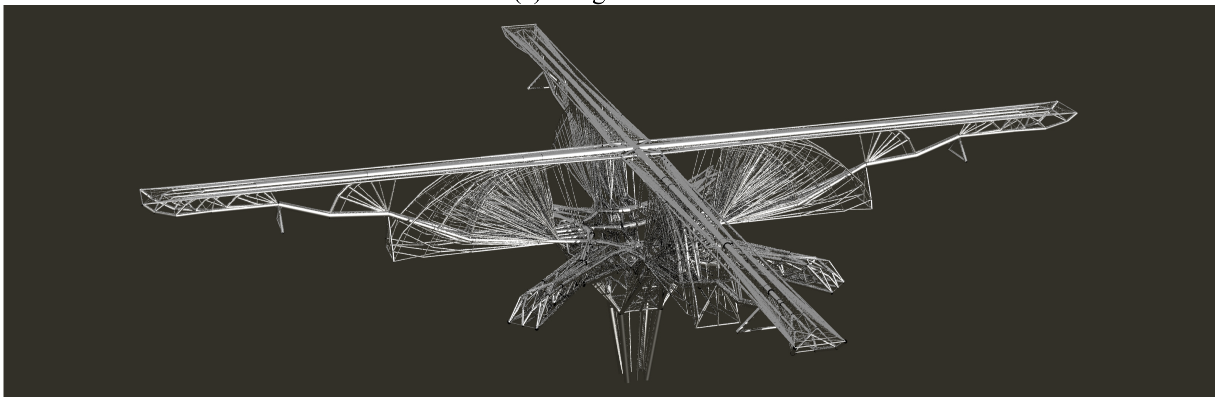


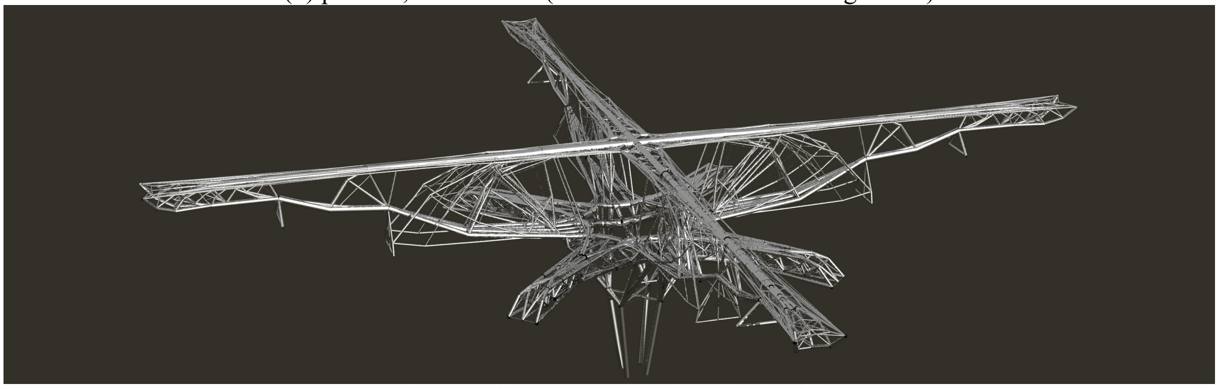
Figure 5. Various solutions generated in the roof design example (incl. Euler buckling): (a) design domain; (b) reference solution obtained using layout & geometry optimization; (c) - (e): simplified solutions, with parameters (i.e. s and ϵ) progressively increased

5.3 Complex design domain

When the design domains are complex, relatively dense nodal grids are employed in layout optimization to ensure nodes are distributed everywhere. As a result, outcome structures are prone to be very complex. In this case, the proposed simplification techniques can be utilized to improve the practicality of the designs. As shown in figure 6 (a), a relatively complex design domain is used, discretized with 1431 nodes (1/8 domain modelled due to symmetry). In the first phase, layout optimization is used to generate the initial solution, shown in figure 6 (b); this comprises a large number of members. Since Euler buckling is not incorporated in the standard layout optimization formulation, the generated structure can be resized to address this, leading to a 18.7% increase of volume. In the second phase, geometry optimization can be utilized to rationalize the layout, effectively reducing the volume difference to only 1.46% by reducing the lengths of members in compression, figure 6 (c). In phase 3, the structural complexity is significantly reduced, figure 6 (d), with negligible increase of volume (1.50%).


(a) design domain


(b) phase 1, diff = 0.0% (+18.7% after Euler buckling check)


(c) phase 2, diff = 1.46% (incl. Euler buckling)

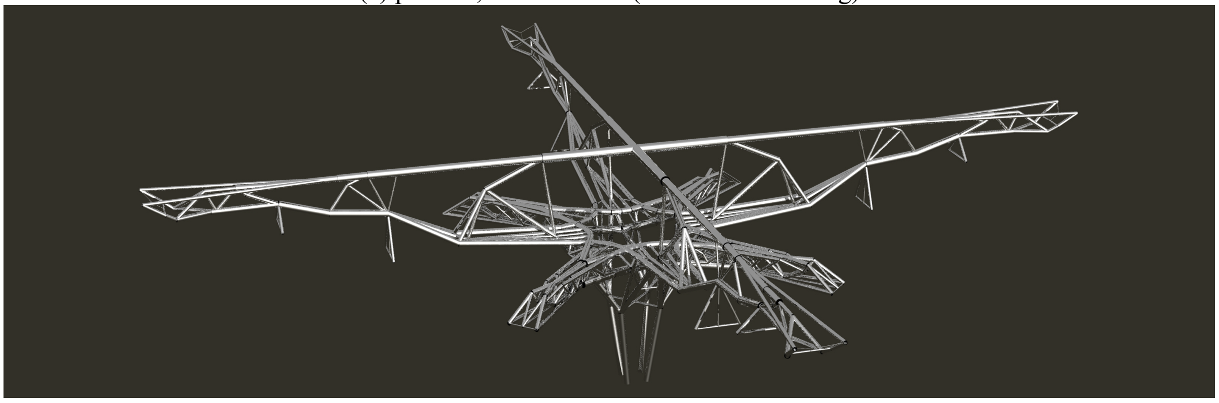

(d) phase 3, diff = 1.50% (incl. Euler buckling)

Figure 6. Optimization phases in a relatively complex design problem: (a) design domain; (b) phase 1, layout optimization; (c) phase 2, geometry optimization, (d) phase 3, simplification using Φ_M and Φ_J

6. Conclusions

Layout optimization employing the ‘ground structure’ approach provides a powerful and highly efficient means of generating theoretically optimal lightweight structures. However, the solutions are often complex in form, and simplification methods are needed to generate more practical designs.

Adding joint costs requires only very minor modification of the standard layout optimization formulation. In this case simpler but less efficient layouts can be identified from the ‘ground structure’. Simplification can also be performed in a post-processing step; by formulating complexity indices using smooth Heaviside functions, the number of members in the structure, and the number of members meeting at joints, can be minimized via a non-linear optimization step.

The techniques described have been implemented in a Rhino-Grasshopper environment to permit user-interaction during the design process. By supplying a range of controlling parameters, various candidate design solutions can be generated for use at the conceptual design stage. The efficacy of the proposed approach is demonstrated via examples, showing that layout optimization with simplification techniques potentially provides a powerful means of designing spatial structures.

Acknowledgements

This work is supported by Engineering and Physical Sciences Research Council (EPSRC) grants EP/N023471/1, EP/N023269/1, and EP/N019652/1. The first author appreciates Arup for providing the last case study example in this paper.

References

[1] O. Sigmund, "On the usefulness of non-gradient approaches in topology optimization", *Structural and Multidisciplinary Optimization*, vol. 43, pp. 589-596, 2011.

[2] W. S. Dorn, R. E. Gomory and H. J. Greenberg, "Automatic design of optimal structures", Journal de Mecanique, vol. 3, pp. 22-52, 1964.

[3] M. Gilbert, A. Tyas, A., "Layout optimization of large-scale pin-jointed frames", Engineering Computations, vol. 20, no. 8, pp. 1044-1064, 2003.

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

[5] H. Fairclough, M. Gilbert, C. Thirion, A, Tyas, "Balancing complexity and structural efficiency in the design of optimal trusses". In: *Proceedings of the IASS Symposium*, City: Boston, USA, 2018.

[6] E. W. Parkes, "Joints in optimum frameworks", *International Journal of Solids and Structures*, vol. 11, pp. 1017-1022, 1975.

[7] A. Wächter, L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming", *Mathematical Programming*, vol. 106, pp. 25-57, 2016.

[8] J. K. Guest, J. H. Prévost, T. Belytschko, "Achieving minimum length scale in topology optimization using nodal design variables and projection functions", *International Journal for Numerical Methods in Engineering*, vol. 61, pp. 238-254, 2004.

[9] M. P. Bendsøe, O. Sigmund, *Topology optimization: theory, methods, and applications.* Springer Science & Business Media, 2013.

[10] J. Schwarz, T. Chen, K. Shea, T. Stanković. (2018). "Efficient size and shape optimization of truss structures subject to stress and local buckling constraints using sequential linear programming", *Structural and Multidisciplinary Optimization*. Available: https://doi.org/10.1007/s00158-017-1885-z