



# THE UNIVERSITY *of* EDINBURGH

## Edinburgh Research Explorer

### Computational layout design optimization of frame structures

**Citation for published version:**

Ye, J, Shepherd, P, He, L, Gilbert, M, Davison, B, Tyas, A, Gondzio, J, Weldeyesus, A & Fairclough, H 2017, Computational layout design optimization of frame structures. in Proceedings of the IASS Annual Symposium 2017. International Association for Shell and Spatial Structures (IASS), IASS Annual Symposium 2017, Hamburg, Germany, 25-28 September.

**Link:**

[Link to publication record in Edinburgh Research Explorer](#)

**Published In:**

Proceedings of the IASS Annual Symposium 2017

**General rights**

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact [openaccess@ed.ac.uk](mailto:openaccess@ed.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.



## Computational layout design optimization of frame structures

Jun Ye\*, Paul Shepherd<sup>a</sup>, Linwei He<sup>b</sup>, Matthew Gilbert<sup>b</sup>, Buick Davison<sup>b</sup>, Andy Tyas<sup>b</sup>,  
Jacek Gondzio<sup>c</sup>, Alemseged G. Weldeyesus<sup>c</sup>, Helen Fairclough<sup>b</sup>

\* Department of Architecture & Civil Engineering, University of Bath, Bath, UK.  
j.ye@bath.ac.uk

<sup>a</sup> Department of Architecture and Civil Engineering, University of Bath

<sup>b</sup> Department of Civil & Structural Engineering, University of Sheffield

<sup>c</sup> School of Mathematics, University of Edinburgh

### Abstract

Engineers often expend considerable effort identifying the most efficient cross-section sizes for the individual structural members forming a structure, but may neglect to check whether members are optimally positioned, or are even needed at all. This can lead to far more material being used to form a building structure than is needed. To address this, layout optimization can potentially be used early in the design process to identify efficient arrangements of structural members. This paper introduces an interactive design approach that combines parametric modelling and layout optimization, using an adaptive 'member adding' technique to allow large scale problems to be solved on a standard desktop PC. Incorporation of the approach in Rhino-Grasshopper allows integration of geometric modeling and structural layout optimization within a single interactive modeling environment. This paper briefly outlines the underlying theory and implementation details, and then describes application of the approach to benchmark problems and a case study problem, a three-centred space frame arch roof. In this case it is shown that a 30% reduction in material usage can potentially be achieved through the use of a layout optimization-based approach, but that measures to improve the practicality of the solutions for use in practice are required. This is being addressed as part of a new collaborative research project involving the Universities of Bath, Sheffield and Edinburgh.

**Keywords:** Layout optimization, Grasshopper, parametric design, building structures

### 1. Introduction

Space frames are widely used for long-span structures due to their inherent light weight, high stiffness and constructional efficiency, including off-site manufacture potential. Designers generally need to decide very early on in the design process the layout of the constituent structural members, long before detailed sizing of members has been undertaken. Once a given layout is chosen it becomes difficult and expensive to make changes subsequently, since many other aspects of the design depend on it (such as column positions, façade design, roof cladding, etc. It is therefore important that practical tools are made available to help the designer quickly and effectively identify efficient layouts of structural members.

Although the theory behind ground-structure based numerical layout optimization has been understood for more than half a century, solution techniques have tended to be computationally expensive, limiting its application in practice. Thus, conceptual designs for structural forms such as long-span roofs or high-rise buildings have tended to be developed based on a designer's intuition. However, nowadays there are demands for ever more complex building geometries, and it is normally not obvious what the most efficient structural layout might be. Thus relying purely on intuition is unlikely to lead to the most efficient designs being identified. There is therefore a need for practical tools

which can quickly and efficiently identify an efficient structural layout for a given set of practical design constraints.

This paper introduces a new plugin for Grasshopper which can identify optimal structural layouts using the adaptive layout optimization algorithm developed by Gilbert and Tyas [1]. The approach is capable of taking into account practical design considerations such as multiple load cases and joint costs. After demonstrating its application to benchmark problems, the tool is applied to a long-span roof structure, highlighting the potential savings that can be made in terms of material usage.

## 2. Numerical formulation

In order to optimize large-scale truss structures, an efficient layout optimization formulation is required. Both elastic and plastic formulations have been explored by the research community. The elastic formulation, which seeks to minimize compliance of a structure, involves the use of a stiffness matrix. However, during the optimization process the stiffness matrix may become singular if members in the structure have cross-sectional areas which approach zero [2]. In contrast the plastic layout optimization formulation instead involves the use of a force equilibrium matrix and does not suffer from this problem. The plastic layout optimization formulation has been found effective to achieve an optimum solution and has been extended to deal with problems involving multiple load cases, self-weight, and member buckling [3]. This latter formulation will therefore be used here.

### 2.1. The plastic layout optimization formulation

Consider a 3-dimensional pin-jointed truss structure comprising  $n$  nodes,  $m$  axially loaded members and  $M$  load cases. In this case the plastic layout optimization formulation designed to find the minimum volume of material that satisfies force equilibrium and limiting stress criteria can be written as follows:

$$\min V = \mathbf{l}^T \mathbf{a} \quad (1)$$

subject to:

$$\mathbf{B}\mathbf{q}^\alpha = \mathbf{f}^\alpha, \quad \alpha = 1, \dots, M \quad (2)$$

$$\left. \begin{array}{l} a_i \geq \left\{ \frac{q_i^+}{\sigma_i^+} + \frac{q_i^-}{\sigma_i^-} \right\}^\alpha \\ \left\{ q_i^+ \right\}^\alpha, \left\{ q_i^- \right\}^\alpha \geq 0 \\ a_i \geq 0 \end{array} \right\} \alpha = 1, \dots, M; \quad i = 1, \dots, m \quad (3)$$

where  $V$  is the total volume of structural material,  $\mathbf{l}^T = \{l_1, l_2, \dots, l_m\}$  and  $\mathbf{a}^T = \{a_1, a_2, \dots, a_m\}$  are the vectors of lengths and cross-sectional areas of the bars, respectively.  $\mathbf{B}$  is the nodal equilibrium matrix of size  $3n \times 2m$ ,  $\mathbf{q}^T = \{q_1^+, q_1^-, q_2^+, q_2^-, \dots, q_m^+, q_m^-\}$ , where  $q_i^+, q_i^-$  are respectively the tensile and compressive forces in member  $i$ . It should be noted that for any active member in the optimum structure, the internal force can be either tensile or compressive (i.e. only one value of  $q_i^+, q_i^-$  will be non-zero).  $\mathbf{f}^T = \{f_1^x, f_1^y, f_1^z, \dots, f_i^x, f_i^y, f_i^z, \dots, f_n^x, f_n^y, f_n^z\}$  is the nodal force vector.  $\sigma_i^+, \sigma_i^-$  are respectively the limiting tensile and compressive stresses, whilst  $\alpha$  is used to denote the index of  $M$  load cases.

This is a linear programming problem that can be solved efficiently using modern interior point solvers. For a given nodal discretization a globally optimal solution is guaranteed. The adaptive ‘member adding’ technique [1] can be used to significantly improve computational efficiency, with no effect on the solution obtained.

## **2.2. Transmissible loads**

In the standard optimization formulation presented in Section 2.1, the nodal loads in the vector  $\mathbf{f}^T$  of equation (2) require that there must be a node present at their point of application. This means that the location of the point of application of the load(s) has a strong influence on the resulting optimal layout. To mitigate this, so-called transmissible loads can be introduced. In this case a load is applied to a group of nodes along a vertical line of action instead of to one, and the load is distributed optimally within this group in order to reduce the volume of material required. More details on the transmissible load formulation and its application can be found in papers by Gilbert et al. [4] and Darwich et al. [5].

## **3. Implementation**

The formulation described in section 2 above has been made available by the University of Sheffield in a plugin for the Rhinoceros-based geometric modeling tool Grasshopper [6], providing an interactive parametric modelling environment for users. An example of the components in use is shown in Figure 1, where they can be grouped into the following types:

- (a) 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.
- (b) Design. A number of components are then used to assign material properties such as yield stress, Young’s modulus, and material density. The support and load conditions need to be defined, with multiple load cases specified as necessary.
- (c) 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 in the structure is also reported.
- (d) Visualization. A component is provided to visually present the resulting members, joint positions, internal forces and cross-sectional areas.

## **4. Example problems**

To demonstrate the effectiveness of the Grasshopper optimization tool presented above, a range of example problems will now be considered, starting with standard benchmark problems and moving on to the design optimization of a three-centered arch space frame roof structure.

The optimization problems were solved using linear programming and the adaptive ‘member adding’ procedure was used in the optimization process. With respect to the optimization solvers employed, all problems were solved using the MOSEK interior point solver [7]. The problems were run on a 64-bit Windows machine with 3.4GHz Intel CPU and 16GB of installed memory.

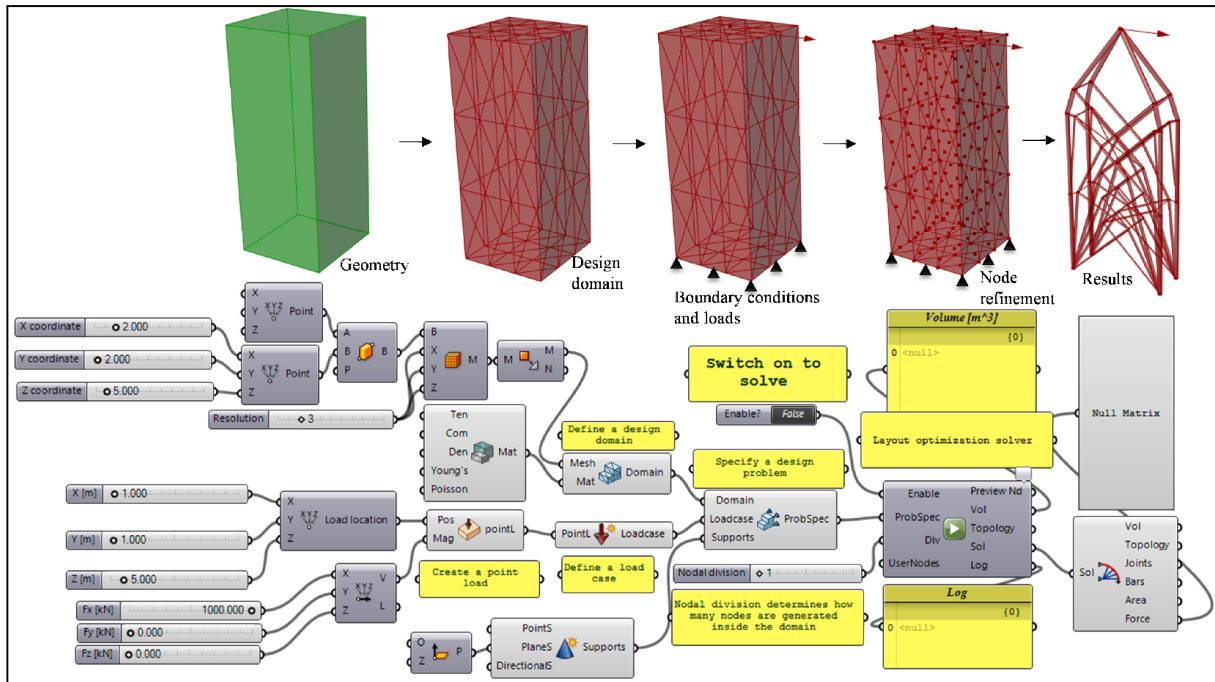


Figure 1: Grasshopper layout optimization plugin: inputs and components

#### 4.1. Benchmark problems

The first benchmark problem presented here is the cantilever truss previously considered by Hemp [8]. Hemp solved the problem analytically and also provided an approximate value for the volume of  $V = 4.34Pl / \sigma$ , where  $P$  is the magnitude of the point load,  $l$  is the span and  $\sigma$  is the limiting stress, and where  $\sigma^+ = \sigma^- = \sigma$ . A more accurate value for the volume was calculated more recently by Tomasz Lewinski;  $V = 4.32168Pl / \sigma$  (personal communication).

Using a  $41 \times 41$  node discretization (1412040 potential members), as shown in Figure 2(a), and taking  $P$  and  $\sigma$  as unity, a numerical solution of  $V = 4.343l$  was achieved, which is within 0.5% of the exact analytical value. The resulting layout is shown in Figure 2(b). Note that the same solution is also obtained when using a transmissible load, with  $P$  applied to the group of nodes along the right hand side of the design domain.

Now consider the case when the limiting tensile stress is increased to 3 times that of the limiting compressive stress. With the point load again applied to the node at the mid-height of the domain, a volume of  $2.8643l$  was obtained. It can be seen from Figure 2(c) that the compressive members have larger cross-sectional areas compared to the tensile members. However, when a transmissible load was used in this case the load was found to move downwards to effectively reduce the lengths of the compressive members, as shown in Figures 2(d). The volume was reduced by 1.6% to  $2.819l$ .

It can be evident from Figure 2 that the resulting trusses contain numerous joints and are therefore impractical for use in practice, at least when conventional fabrication techniques are employed. To address this two potential rationalization techniques have recently been considered by He and Gilbert [9]; here the use of a simple joint cost penalty is used. This entails adding a fixed length to each member to account for the expense of fabricating joints, effectively penalizing short members. Figure 3 shows the effect of including joint lengths in the cantilever problem. This shows that for an increase in volume of a few percent, the complexity is significantly reduced.

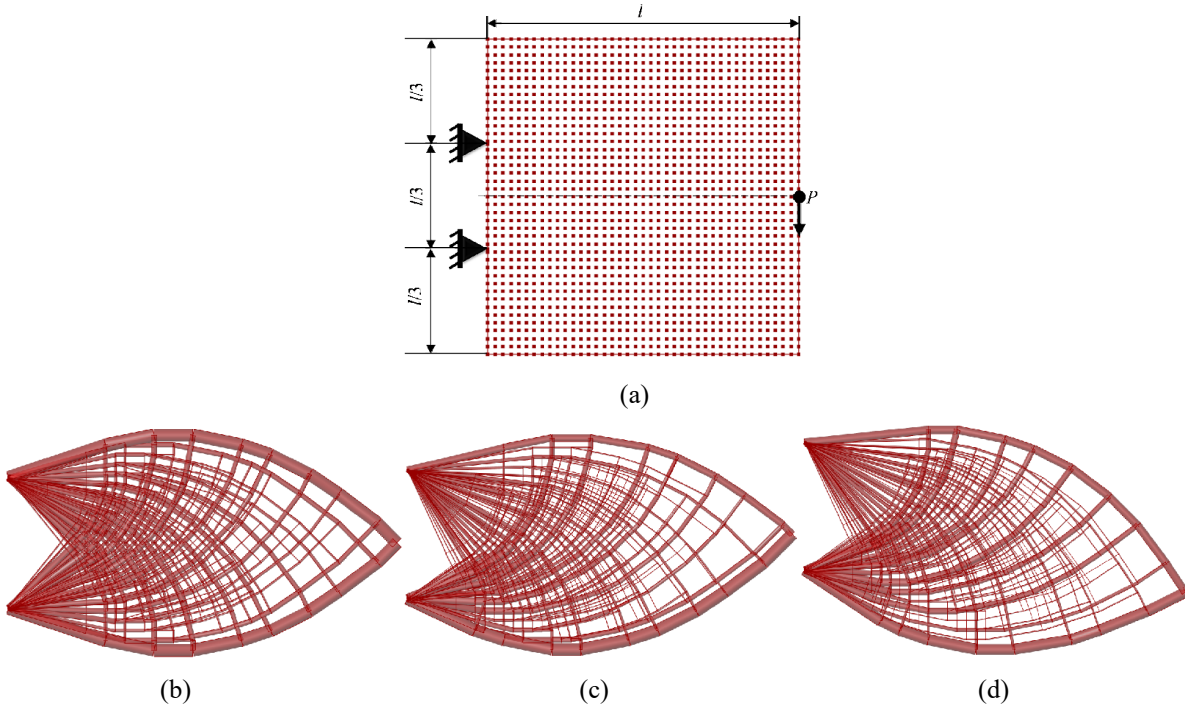


Figure 2: Benchmark 2D cantilever problem: (a) design domain; (b) equal stress,  $V = 4.343l$ ; (c) as (b) but with unequal limiting stress  $\sigma^+ = 3\sigma^-$ ,  $V = 2.8643l$ ; (d) as (c) but with transmissible load,  $V = 2.819l$ .

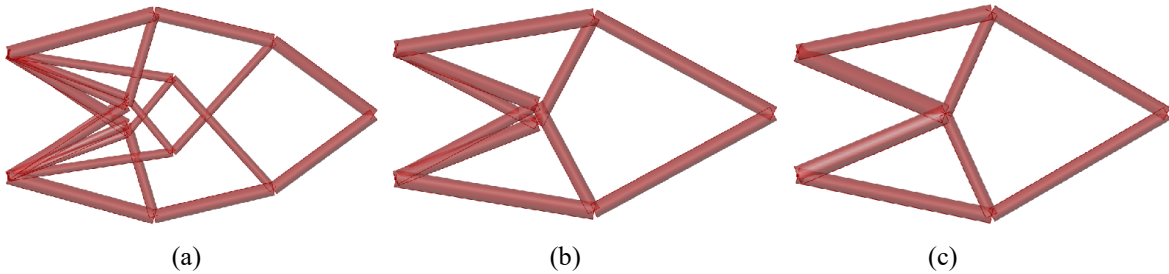


Figure 3: Cantilever problem with joint lengths of: (a) 0.05 ( $V = 4.477l$ ); (b) 0.1 ( $V = 4.582l$ ); (c) 0.20 ( $V = 4.596l$ )

#### 4.2 Barrel-Vault Roof

The next problem involves the redesign of a long span three-centered arch space frame roof structure. Such structures are commonly used as coal storage structures in China. A key design requirement is to ensure enough space for storage of the coal and the operation of a stacker / reclaimer, as shown in Figure 4.

The original three-centered roof design used a grid structural form comprising repeating quadratic pyramid cells with a size of  $4.35m \times 4m$ . The outer radius and the central angle of the middle arc are  $R = 98.2m$  and  $\varphi_1 = 51^\circ$ , respectively. The two side arcs have an outer radius  $r = 27.8m$  and a central angle of  $\varphi_2 = 55^\circ$ . The total height and span of the roof are  $32.6m$  and  $112.3m$  respectively, while the depth of the grid is  $4m$ . The standard values of loads for the roof design are: static load  $D_k = 0.2kN / m^2$ , live load  $L_k = 0.5kN / m^2$ , snow load  $S_k = 0.3kN / m^2$  and the basic wind pressure is

$W_k = 0.44kN / m^2$ . The terrain category is *A* according to the original design and the Chinese Standard for actions, while the pressure coefficients were determined from wind tunnel tests and are shown in Figure 4. The self-weight of the structure used in the optimisation is taken as  $G_k = 0.4kN / m^2$ . According to the original design, the controlling load case combinations are:

$$Q_1 = 1.2(G_k + D_k) + 1.4L_k \quad (5)$$

$$Q_2 = 1.2(G_k + D_k) + 1.4L_k + 0.84W_k \quad (6)$$

Assuming wind load from both the left and right directions, there are three controlling load cases. The steel usage in the original design was  $47kg / m^2$ ; for a  $100m$  long building the total volume of steel required was  $67.237m^3$ .

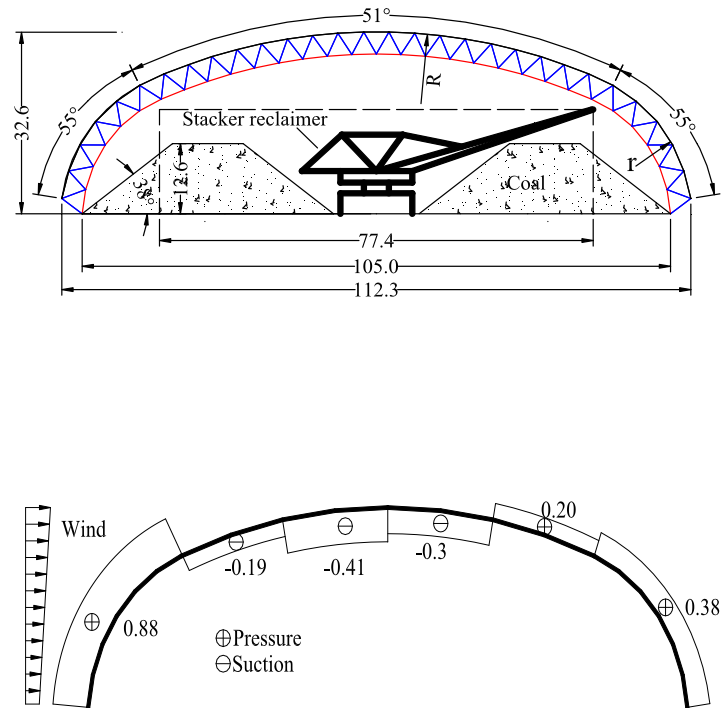


Figure 4: Section through the original design of a three-centered space-frame arch roof for coal storage and the wind pressure coefficients used in the design

#### 4.2.1 2D roof optimization

It was initially assumed that the roof structure comprised a series of 2D plane trusses designed for the  $113.2m \times 32.6m$  2D domain with a longitudinal length of  $1m$ . The design domain was then discretized using a very coarse discretization of 273 nodes, distributed randomly, leading to 37128 potential members. A grasshopper component was developed to convert surface pressures to point loads for use in the optimization. The computed optimum material volume  $V_1$  and designs are shown in Table 1 and Figure 5. The computational time for each optimization was less than 10s, which shows the efficiency of plane truss optimization. Using joint lengths of  $0.1m$  and  $0.2m$  led to a reduction in the number of



bars (by 53.4% and 56.1% respectively) and number of joints (by 34.3% and 38.6% respectively), with only modest increases in volume (0.23% and 0.55% respectively).

Table 1: Comparison between optimum solutions and CPU cost for 2D layout optimization of the roof

Joint length (m)	$V_1$ (m <sup>3</sup> )	Numbers of bars	Numbers of joints	CPU time* (s)
0	0.061393	365	137	6.1
0.1	0.061537	170	90	4.7
0.2	0.061732	160	84	5.3

\* Includes the time to setup the problem and complete the optimization process

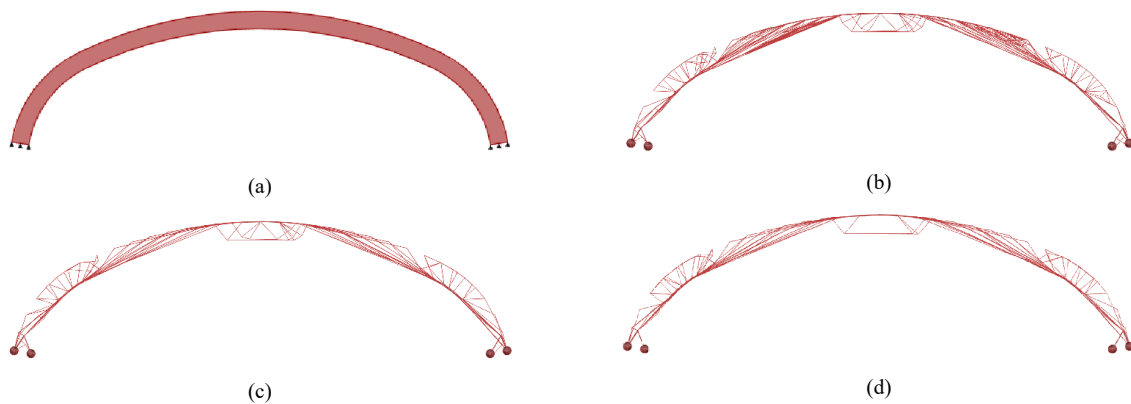


Figure 5: 2D layout optimization solution of the roof using different joint lengths: (a) design domain; (b) joint length = 0m; (c) joint length = 0.1m; (d) joint cost = 0.2m (the layouts are not symmetrical partly due to the use of a randomly generated nodal grid)

#### 4.2.2 3D roof optimization

A 3D design optimization was also performed. In this case boundary surfaces were meshed using  $4m \times 4m$  quadrilateral facets, resulting in 2096 vertices. Nodes were again randomly distributed across the design domain.

Table 2 shows the optimum volume with and without transmissible load ( $V_1$ ). In addition, the volume required to also withstand Eurocode 3 [10] buckling requirements has also been calculated ( $V_2$ ). The member sections used for the design are taken from the list of standard circular hollow sections presented in [11]. Table 2 shows that the minimum volume of material is about  $6.1m^3$ , approx. 100 times the plane truss optimization case as expected (since a longitudinal of  $100m$  rather than  $1m$  was used). However, the initial optimization neglected buckling and to address this the volume of material required needed to be increased by approx. 9 times in a post-processing step; more effective ways of treating buckling are currently under development. However, the volumes are still less than that of the original design presented in Figure 5 (depending upon the number of nodes and joint length used, the saving ranges from 5.9%-20.6% in the case of a fixed position pressure load, and 6.9%-33.9% when transmissible loads are used).

Other random distributions of nodes were tried, but the results were found not to vary markedly, indicating that the specific distribution does not have a major influence on the outcome obtained.



Table 2: Optimum solutions for 3D layout optimization of the roof with and without transmissible loads.  
 (Volume  $V_1$  and  $V_2$  are respectively the raw volume and the volume after buckling has been taken account of.)

Number of randomly distributed nodes	Joint length (m)	Without transmissible loads				With transmissible loads					
		Volume $V_1$ ( $m^3$ )	Numbers of bars	Numbers of joints	CPU time* (min)	Volume $V_2$ ( $m^3$ )	Volume $V_1$ ( $m^3$ )	Numbers of bars	Numbers of joints	CPU time* (min)	Volume $V_2$ ( $m^3$ )
1000	0	6.154	4982	1530	6.1	58.930	6.154	4812	1713	6.0	44.391
	0.1	6.156	2899	1312	6.0	60.252	6.155	4767	1706	6.1	49.573
	0.2	6.168	2779	1275	6.5	61.509	6.159	4621	1656	5.9	53.708
2000	0	6.142	5006	1658	7.6	55.317	6.131	6647	2234	7.5	48.46
	0.1	6.150	3320	1425	7.6	60.700	6.139	6255	2212	8.2	56.172
	0.2	6.157	3235	1394	7.9	62.670	6.141	6217	2194	7.2	58.942
3000	0	6.137	5295	1799	18.1	53.370	6.047	7741	2727	17.6	48.090
	0.1	6.149	3585	1498	18.4	61.413	6.062	7190	2544	18.1	57.729
	0.2	6.154	3330	1428	18.9	63.249	6.071	7012	2484	18.2	62.589

\* Includes the time to setup the problem and complete the optimization process

#### 4.2.3. Form-finding of the coal storage roof

The three-centered space frame truss was initially designed using a form comprising three arcs, as shown in Figure 5(a). However, this leads to a severely constrained design domain. Therefore the optimization tool was also used with a less constrained domain, shown in Figure 6 (where the extent of the domain is indicated by red lines). This domain was generated by subtracting a 18.6m high trapezoid, representing the clear space required for storage and operational space, from a rectangular design domain. Supports were defined along the base of the domain. The vertical load combination shown in equation (5) was distributed horizontally as point loads every 4m, shared amongst vertical columns of nodes using the transmissible load formulation. Two cases are considered here: one with a 32.6m high rectangular design domain (as per the original design) and another with a 50m high domain, to leave the flexibility to discover a more efficient form. A nominal joint length was introduced to simplify the resulting designs, which are shown in Figure 7.

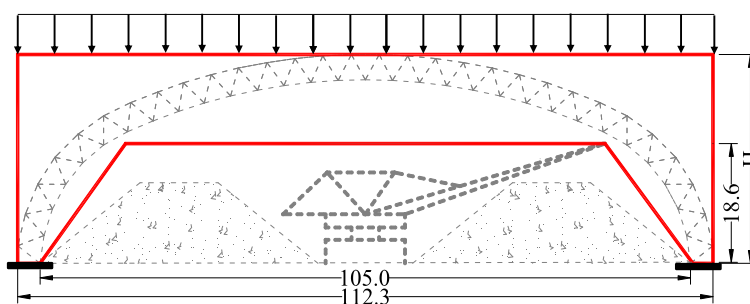


Figure 6: Revised design domain for roof, using transmissible loads

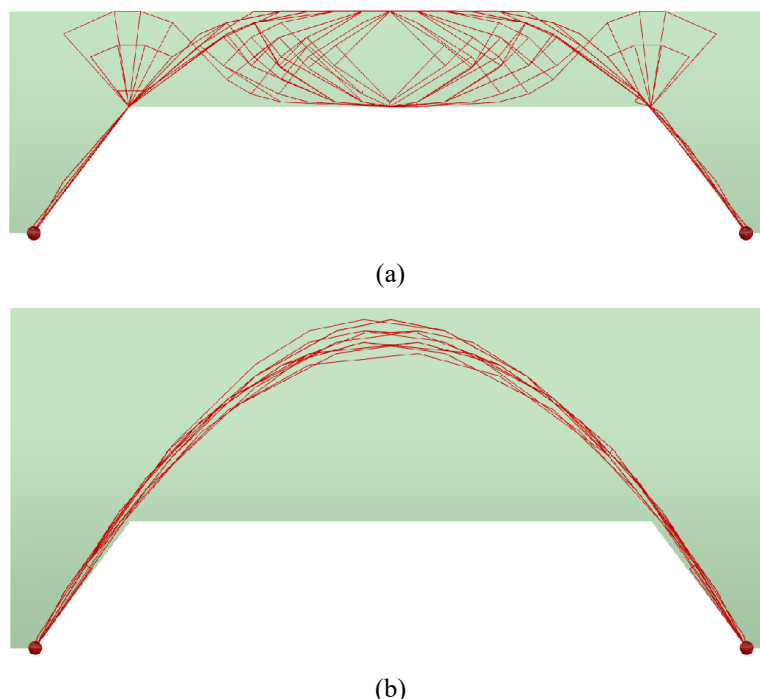


Figure 7: Form-finding solutions for roof structure: (a) domain height  $H=32.6m$  (b) domain height  $H=50m$  (with joint length =  $0.1m$  in both cases)

With the height of design domain as per the original design, the optimum layout resembles a plane truss in the mid-span region, with inclined members at the edges taking the forces down to ground level. However, the layout changes markedly when the domain height is increased to  $50m$ , which leads to a parabolic form being identified. In both cases it is evident that rationalization of the solutions would be required in order for these to be realised in practice. (Also, the use of additional load cases would be required to ensure robustness.)

## 5. Conclusions

This paper has briefly outlined how numerical layout optimization can be used at the conceptual design stage to identify structurally efficient layouts. A design approach that combines parametric modelling and layout optimization, using an adaptive ‘member adding’ technique, has been developed to allow large scale problems to be solved on a standard desktop PC. Incorporation of the approach in Rhino-Grasshopper allows integration of geometric modeling and structural layout optimization within a single interactive modeling environment.

It has been found that including joint lengths in the optimization can significantly simplify the resulting layout, with only a small increase in material consumption. Also, for the practical case-study problem considered, it was found that material savings of up to approx. one third could be achieved; savings were greatest when transmissible loads were used.

However, considerable further work is required in order to enable the full potential of the layout optimization technique to be realised. For example, there is a need to incorporate buckling in the formulation in a computationally efficient manner, and to take account of practical considerations such as specified minimum intersection angles between members at joints. These issues are currently being addressed as part of a new collaborative research project involving the Universities of Bath, Sheffield and Edinburgh.

## **Acknowledgements**

This work described is supported by Engineering and Physical Sciences Research Council (EPSRC) grants EP/N023269/1, EP/N023471/1, and EP/N019652/1.

## **References**

- [1] Gilbert M, Tyas A. Layout optimization of large-scale pin-jointed frames. *Engineering Computations*. 2003;20:1044-64.
- [2] Rozvany GIN. On design-dependent constraints and singular topologies. *Structural and Multidisciplinary Optimization*. 2001;21:164-72.
- [3] Darwich W. Novel computational implementations for ultimate limit state analysis and design. PhD thesis; The University of Sheffield; 2010.
- [4] Gilbert M, Darwich W, Tyas A, Shepherd P. Application of large-scale layout optimization techniques in structural engineering practice. 6th World Congresses of Structural and Multidisciplinary Optimization. 2005.
- [5] Darwich W, Gilbert M, Tyas A. Optimum structure to carry a uniform load between pinned supports. *Structural and Multidisciplinary Optimization*. 2010;42:33-42.
- [6] McNeel R. Grasshopper generative modeling for Rhino. Computer software (2011b), <http://www.grasshopper3d.com>. 2010.
- [7] MOSEK A. Denmark (2016). The MOSEK Optimization Tools Manual Version 7.1.
- [8] Hemp W. *Optimum Structures*. Clarendon Press, Oxford. 1973.
- [9] He L, Gilbert M. Rationalization of trusses generated via layout optimization. *Structural and Multidisciplinary Optimization*. 2015;52:677-694.
- [10] CEN. Eurocode 3: Design of Steel Structures. Part 1-1: General Rules and Rules for Buildings. Brussels: European Committee for Standardization; 2005.
- [11] Cold Formed Circular Hollow Sections to BS EN 10219-2: 2006. URL: [http://www.roymech.co.uk/Useful\\_Tables/Sections/Cir\\_Sect.html](http://www.roymech.co.uk/Useful_Tables/Sections/Cir_Sect.html)