This is a repository copy of *Layout optimization of simplified trusses using mixed integer linear programming with runtime generation of constraints*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/152827/

Version: Published Version

Article:

Fairclough, H. and Gilbert, M. orcid.org/0000-0003-4633-2839 (2020) Layout optimization of simplified trusses using mixed integer linear programming with runtime generation of constraints. Structural and Multidisciplinary Optimization, 61 (5). pp. 1977-1999. ISSN 1615-147X

https://doi.org/10.1007/s00158-019-02449-7

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Structural and Multidisciplinary Optimization
https://doi.org/10.1007/s00158-019-02449-7

RESEARCH PAPER

Layout optimization of simplified trusses using mixed integer linear programming with runtime generation of constraints

H. Fairclough[1] · M. Gilbert[1]

Received: 28 February 2019 / Revised: 12 September 2019 / Accepted: 5 November 2019
© The Author(s) 2020

Abstract

Traditional truss layout optimization employing the ground structure method will often generate layouts that are too complex to fabricate in practice. To address this, mixed integer linear programming can be used to enforce buildability constraints, leading to simplified truss forms. Limits on the number of joints in the structure and/or the minimum angle between connected members can be imposed, with the joints arising from crossover of pairs of members accounted for. However, in layout optimization, the number of constraints arising from 'crossover joints' increases rapidly with problem size, along with computational expense. To address this, crossover constraints are here dynamically generated and added at runtime only as required (so-called lazy constraints); speedups of more than 20 times are observed whilst ensuring that there is no loss of solution quality. Also, results from the layout optimization step are shown to provide a suitable starting point for a non-linear geometry optimization step, enabling results to be obtained that are in agreement with literature solutions. It is also shown that symmetric problems may not have symmetric optimal solutions, and that multiple distinct and equally optimal solutions may be found.

Keywords Layout optimization · Topology optimization · Discrete optimization · Truss optimization · Multiple load-cases · Mixed integer linear programming

1 Introduction

Reducing the volume of material consumed in construction projects is a challenge of increasing importance. The theory of minimum volume structures is well established (Michell 1904) and can provide benchmark values to help evaluate the structural efficiency of proposed designs. However, for practical usage, the truss-like continua generated by classical methods are often challenging or impossible to construct in practice. Whilst novel construction methods may allow more complex designs to be realized in the future, more immediate benefits may arise through the use of optimization methods capable of generating less complex solutions.

Responsible Editor: Jose Herskovits

✉ M. Gilbert
m.gilbert@sheffield.ac.uk

[1] Department of Civil and Structural Engineering, The University of Sheffield, Sir Frederick Mappin Building, Mappin Street, Sheffield, S1 3JD, UK

Numerical topology and layout optimization methods allow structurally efficient forms to be identified. These methods can be divided into continuum-based approaches and discrete truss/frame-based methods. Perhaps the best known of the continuum-based approaches is the SIMP method (Bendsøe and Sigmund 1999). This uses penalization to drive the solution to a distinct structural form. A number of extensions to the SIMP method have been suggested to control the complexity of the forms identified. For example, minimum length scale (Zhou et al. 2015) and maximum perimeter length (Park et al. 2018) constraints have been proposed. However, the quantities involved are not intuitive when considered in the context of a typical structural engineering design problem. More generally, interpreting solutions from a continuum topology optimization in a structural engineering context will often be challenging, and is likely to necessitate considerable manual post-processing effort. Furthermore, the fraction of the available design space occupied by structural members in a typical building or bridge structure will generally be very small, such that very high numerical resolutions are required to achieve accurate results (see, e.g. Aage et al. 2017).

Published online: 11 March 2020

Here, attention is therefore focussed on methods that directly represent the structure as a frame or truss composed of a series of discrete members. The most well-known numerical method of this type is the ground structure layout optimization method (Dorn et al. 1964), in which an optimal set of members is chosen from a dense initial ground structure. The commonly used plastic design formulation gives rise to a problem that can be solved using linear programming, allowing problems involving large numbers of potential members to be solved rapidly.

One potential method for limiting the complexity of a truss structure was suggested by Parkes (1975). This involves adding a volume penalty to represent the material required in the connections joining each member in the structure, known as a joint cost. The penalty is linearly related to the member volume, allowing this method to readily be integrated with numerical ground structure based layout optimization methods, whilst retaining computational efficiency. However, as this method penalizes short members, rather than the number of members, its efficacy is problem dependent.

Another, similar concept was presented by Prager (1977); however, here the cost of a joint was assumed to be constant. With Prager's approach minimum volume trusses with fixed numbers of joints are first found, with the ranges of costs for which they are optimal later established. The minimum volume trusses are based on use of a mesh-wise constant strain field, furnishing a set of geometrical rules for the joints. Mazurek et al. (2011) identify the same rules by direct optimization of joint positions for simple structures, building on the principles of graphical statics. These rules imply that the angles between members at every unsupported joint should be identical. Prager (1978) extends these rules to the case where $\sigma_T \neq \sigma_C$, where σ_T and σ_C are respectively the limiting stress in tension and compression. Each of the aforementioned authors applies their approach to three-force problems, such as the classical Michell cantilever problem (where two of the three forces are support reactions).

Of these approaches, only the linear joint cost method is applicable to multiple load-case problems. When multiple load-cases are present, the optimal solutions for elastic and plastic design problems diverge. The plastic problem gives rise to the simplest numerical formulation and, since it is accepted by many structural design codes, is considered here.

Optimality criteria for multiple load-case plastic design problems were first given by Prager and Shield (1967). For scenarios with two load-cases, it is possible to use the superposition principle derived by Nagtegaal and Prager (1973) to split the problem into two single load-case problems that can then be superimposed. The superposition principle was later extended to problems with more than two load-cases by Rozvany and Hill (1978), although this has a restricted range of applicability. However, numerical results for various problems demonstrate the 'overlapping' nature of many of the optimal layouts identified. In such cases, manually identifying the layout of a discretized structure becomes challenging.

Various means of characterizing the complexity of a given structure are possible. However, whatever the chosen complexity measure, it generally results in a non-smooth problem that can be challenging to solve numerically. Conceptually, the simplest measures of complexity are the numbers of joints or members in a given structure. Additionally, significant attention has been devoted to limiting the numbers of different cross sections present in a given solution.

Kanno and Fujita (2018) limit the number of joints in solutions whilst minimizing the compliance of the structure, considering both a heuristic method and a mathematical programming formulation including integer variables. However, although the resulting mixed integer second-order cone programming (MISOCP) problem could solve problems with up to 1500 potential members reasonably quickly (in a time of 112 s), conditions were not imposed to prevent intersecting or overlapping members. Therefore, many of their results contain members that cross, which would likely be interpreted as additional joints by practitioners, therefore limiting the usefulness of the results obtained.

A similar approach based on plastic design principles with volume minimization was used by Park (2013), resulting in a mixed integer linear programming (MILP) problem. Here, complexity was reduced by imposing limits on the numbers of members in the solution; he also used a similar approach to identify tensegrity forms.

Tensegrity forms were also identified by Kanno (2013), using another MILP formulation considering both compliance and stress constraints, as well as a number of practical considerations. This formulation prevents the inclusion of intersecting members by including constraints for every intersecting pair of potential members in the ground structure. As the problems considered are limited in size (with ground structures containing up to 18 nodes and 99 potential members), the number of potential intersection points is small (≤ 32). Nonetheless, the problems took up to 67,000 s to solve, and the number of intersection points and the time required would rapidly increase at higher resolutions.

The great majority of work on truss-based optimization methods with discrete constraints has focused on restricting the cross sections of each member to be chosen from a given list of catalogue sections. A comprehensive review of methods for approaching this problem can be found in Stolpe (2016), covering both deterministic (e.g., integer programming) and meta-heuristic methods. In this, it is observed that complex formulations generally restrict the

size of problems, such that these fall within the realm of 'size optimization' rather than true layout or topology optimization. Identification of global optima (deterministic methods) generally has a high computational cost; for example when Achtziger and Stolpe (2007) identified the global optimum for a problem with 131 potential members and 6 possible cross sections, the associated computational time was 488,592 s.

Heuristic methods are simple to implement and are therefore popular with many researchers (e.g. Ahrari et al. 2015; Mortazavi and Toğan 2016; Gonçalves et al. 2015). However, the number of independent design variables is then usually limited ($\ll$ 100). Therefore, these methods generally employ very low-resolution and/or restricted ground structures, often tailored to individual problems.

The moving morphable components (MMC) method, developed by Guo et al. (2014), combines a continuum level-set model and explicit description of geometry. Although this does not obviate the need to employ high resolutions, this does allow complexity to be controlled in intuitive ways. For example, Hoang and Jang (2017) limit the thickness of members, and Zhang et al. (2017) limit the number of 'effective components' ($\approx$ number of members). However, the problem is highly non-linear and prone to identification of local optima; there are also issues treating problems with low-volume fractions.

Identification of local optima is also an issue for ground structure–based methods involving continuous non-linear approximations, such as those of Asadpoure et al. (2015) and Torii et al. (2016), which are usually non-convex. Leng and Duan (2012) use a continuum approximation based on the Heaviside function to prevent intersecting bars; however, only problems with up to 68 potential members are considered, and there is no indication of the associated computational cost.

Ohsaki and Katoh (2005) also include a constraint on member intersection, as well as nodal stability and stress constraints. They use non-linear programming (NLP) and MILP in combination to provide both upper and lower bounds on solutions; problems solved have a maximum of 72 bars, and the ground structure is such that very few members can intersect. Times of up to 3000 s are reported.

Ohsaki (2016) also considers the truss topology optimization problem with discrete cross sections, and additionally considers the problem of combined topology and geometry optimization of trusses, i.e. where the nodal locations are also included as design variables. In this, it is noted that these problems may be both non-convex and non-smooth, and therefore solving them is very difficult. A number of other means of addressing this are also possible, such as the implicit programming approach of Achtziger (2007), and the post-processing rationalization method used by He and Gilbert (2015).

MILP formulations have also been used to incorporate a range of other constraints, including buckling (Groenwold and Stander 1997; Mela 2014), stress constraints (Kanno and Guo 2010), and the requirements of real-world design codes (Van Mellaert et al. 2018). Complex real-world and design code constraints have also been studied using metaheuristic methods (e.g. Koumousis and Georgiou 1994; Villar et al. 2016; Huang and Xie 2007). However, for both MILP and metaheuristic methods, the additional complexity that these constraints cause limits their applicability to size optimization or very low-resolution layout optimization (up to around 200 potential member) problems.

Most numerical approaches in the literature that consider buildability constraints can therefore be seen to fall into one of the following categories: (i) those that present topology optimization problem formulations of such complexity that only trivial scenarios can be solved, and (ii) those that present solution algorithms that produce structures with no guarantee or measure of optimality. The methodology presented in this paper seeks to extend the scale of truss layout optimization problems with basic buildability constraints that are solvable, so as to provide a potentially useful conceptual design tool for practitioners. To this end, an MILP formulation is used to find a globally optimal solution for a ground structure of finite resolution. The main contribution of this paper is to substantially increase the speed by which problems can be solved, and hence also the scale of problems solvable. This is achieved through the runtime generation of some constraints (so-called lazy constraints), as part of a two-stage design process. The MILP problem forms the first stage, followed by an optional second non-linear geometry optimization refinement stage. Application of the developed procedure to a range of problems allows observations to be drawn on the nature of the structures identified as optimal under the imposed buildability constraint, with results compared with analytical solutions in the case of one of the problems considered.

The paper is organized as follows: Section 2 describes the layout optimization formulations that will be employed in the present study; Section 3 describes the non-linear geometry optimization stage; Section 4 describes application of the methods described to a range of numerical example problems; conclusions from the study are then drawn in Section 5.

2 Truss layout optimization formulations

2.1 Linear programming formulation

The well-known plastic layout optimization formulation for volume minimization of trusses subject to stress constraints (Dorn et al. 1964) is used herein; the process involves setting

up a ground structure (Fig. 1) and solving the following linear programming problem:

$$\text{minimize } V = \mathbf{l}^T\mathbf{a} \tag{1a}$$
$$\text{subject to } (\mathbf{B}\mathbf{q}_k = \mathbf{f}_k)_{\forall k} \tag{1b}$$
$$(\sigma_T\mathbf{a} - \mathbf{q}_k \geq 0)_{\forall k} \tag{1c}$$
$$(\sigma_C\mathbf{a} + \mathbf{q}_k \geq 0)_{\forall k} \tag{1d}$$
$$\mathbf{a} \geq 0 \tag{1e}$$

where V is the total volume of all members, $\mathbf{l}$ is a vector of all potential member lengths, and $\mathbf{a}$ is a vector of all potential member areas. $\mathbf{B}$ is a suitable $2n \times m$ equilibrium matrix (for planar problems), where n is the number of nodes and m the number of potential truss members in the ground structure. $\mathbf{q}_k$ is the vector of member internal forces, and $\mathbf{f}_k$ is the externally applied loading, in load-case k. σ_T and σ_C are the limiting stresses in tension and compression. Note that in this formulation, buckling of members may be considered only in a highly simplified manner, by reducing σ_C.

Note that when limitations on structural complexity are imposed (see Section 2.2), it is useful to include overlapping members in the ground structure. For example, considering the problem shown in Fig. 1, if it is required that the solution contains no more than four joints (or members), it is evident that the structure shown in Fig. 1d is not feasible, whereas the structure shown in Fig. 1f is feasible, where these structures were generated respectively without and with overlapping members in the ground structure. In the former case, this would result in an alternative, suboptimal, solution being generated. Thus, overlapping members are included in the ground structures of all problems considered herein.

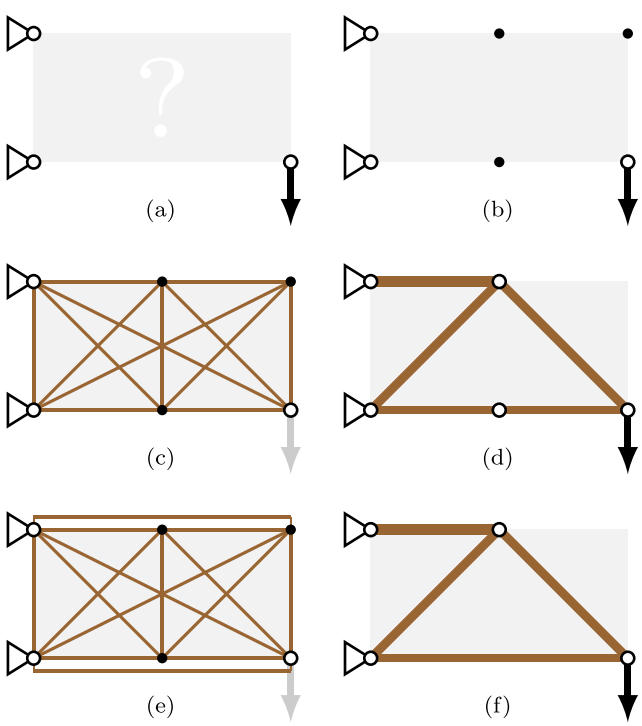


Fig. 1 Truss layout optimization: **a** problem specification (applied load(s), supports and design domain); **b** discretization with nodes; **c** ground structure without overlapping members; **d** unique optimal solution associated with (**c**); **e** ground structure including overlapping members (as used herein); **f** alternative optimal solution obtainable only when using (**e**)

2.2 MILP formulations

2.2.1 Addition of discrete flag variables

The previous formulation can be extended using mixed integer linear programming (MILP) to provide a flexible method capable of imposing a wide range of constraints, including constraints designed to increase the practicality and buildability of a truss structure. In this formulation, a new set of binary variables are added that represent, e.g., the existence of a given potential member or joint. In the case of member flag variables, these are set based on cross section area:

$$M\mathbf{w} - \mathbf{a} \geq 0 \tag{2a}$$
$$w_i \in \{0, 1\} \qquad i = 1, 2, ..., m \tag{2b}$$

where $\mathbf{w} = [w_1, w_2, ..., w_m]^T$ is the vector of flag variables for each potential member in the ground structure. M is a large number, which becomes effectively an upper bound on cross section area and must therefore be larger than any required cross section in the final solution. However, if M is too large, numerical issues can arise; here, M was pragmatically chosen to be 20 times the maximum magnitude of any point load divided by the minimum limiting stress.

To provide flag variables to represent the existence of a joint, the sum of the areas of all members linked to a given node is used:

$$\hat{M}v_j - \sum_{i \in J_j} a_i \geq 0 \qquad j = 1, 2, ..., n \tag{3a}$$
$$v_j \in \{0, 1\} \qquad j = 1, 2, ..., n \tag{3b}$$

where J_j is the set of member indices for all members connected to node j. $\mathbf{v} = [v_1, v_2, ..., v_n]^T$ is another vector of flag variables, where here v_j is the flag variable indicating the involvement of node j. $\hat{M}$ is another sufficiently large number. In this case, $\hat{M}$ should be bigger than the total area of members connected to any node; here, $\hat{M}$ was pragmatically taken to be 4 times M.

The modelling of (3a) is similar to that found in the literature (e.g. Kanno and Fujita 2018) when only the number of nodes are limited. However in light of the presence of integer flags for member existence in the

present formulation, an equivalent formulation whereby $\mathbf{v}$ and $\mathbf{w}$ are linked is possible. This was found to produce inferior performance compared to (3a) (for more details, see Appendix A).

These flag variables can then be used to form constraints to increase the practicality of the forms produced.

2.2.2 Limits on the number of joints, including 'crossover joints'

Due to the much lower number of nodes compared to potential members ($\propto n$ instead of $\propto n^2$), imposing a limit on the number of nodes is comparatively computationally efficient. The simplest method to limit the number of joints to some given value η is to add the following constraint

$$\sum_{j=1}^{n} v_j \leq \eta \tag{4}$$

and constraints (3a,b) to the problem given in (1a–e).

However, this formulation may produce a structure which contains members that intersect each other partway between their ends. These 'crossover joints' will appear to be additional joints from the point of view of the designer, but will by default not be counted as such in the formulation.

To prevent this, constraints can be added for each pair of intersecting members, preventing both of them from being present in the solution at the same time:

$$\sum_{j=1}^{n} v_j \leq \eta \tag{5a}$$

$$(w_h + w_i \leq 1)_{\forall \{h,i\} \in X} \tag{5b}$$

where X is a set containing unordered pairs of indices, $\{h, i\}$, for each pair of intersecting elements.

However, the size of X increases at a very high rate ($\propto n^4$), meaning that the full form of this problem will be extremely computationally expensive to formulate and solve. Fortunately, since only a small subset of these constraints are likely to be used, these can instead be generated on-the-fly, during the running of the solver. Most commercial solvers, including CPLEX (IBM Corp 2015) and Gurobi (2018), are capable of implementing these so-called lazy constraints by allowing user-defined code (often referred to as a callback function) to be called at intervals *during* a single run of the solver. As any intermediate solution which violates one or more potential constraints will be eliminated, this methodology does not alter the final solution, which remains identical to the solution of the full MILP problem containing all constraints from the beginning.

Lazy constraints have previously been used when solving the travelling salesman and related problems (Dantzig et al. 1954), where they have been shown to provide significant advantages in terms of computational efficiency. More recently, Haunert and Wolff (2010) have applied them to the simplification of building outlines for maps. Here it is shown that, when applied to truss structures, the use of lazy constraints enables true topology optimization problems to be solved, i.e., layout optimization problems utilizing fully connected, non-problem specific, ground structures to identify optimal topologies under various different constraints.

The problem is initially provided to the solver without the constraints of (5b) (see Appendix B for the full problem statement). In the initial problem, all member flags in $\mathbf{w}$ can be assumed to be equal to 1 without violating any of the initial constraints. This can be used as a partial warm start, although the speed advantage in explicitly doing so was found to be modest. When a feasible solution is found, the set of members with non-zero areas are identified and each pair from this set (of which there are several orders of magnitude fewer than all pairs of potential members) is checked to see if a crossover joint is produced. If a crossover is found, then an appropriate constraint of the form of (5b) is added to the problem, and remains present in the active problem until the final solution is found.

Note also that for multiple load-case problems, optimal solutions are often made up of multiple, almost independent, forms overlain on top of each other. Therefore, it is preferable to allow crossovers in the solutions, and to take account of them when computing the total number of joints. i.e. (5a,b) becomes:

$$\sum_{j=1}^{n} v_j + \sum_{g=1}^{b} \bar{v}_g \leq \eta \tag{6a}$$

$$(w_h + w_i - \bar{v} \leq 1)_{\forall h,i \in X} \tag{6b}$$

where $\bar{\mathbf{v}} = [\bar{v}_1, \bar{v}_2, ..., \bar{v}_b]^T$ is a vector of flag variables representing the existence of each possible crossover between members of the ground structure. The length of $\bar{\mathbf{v}}$, denoted by b, will be approximately proportional to n^4, although it will also depend on the exact positions of the ground structure nodes.

The constraints of (6a,b) describe the case where two lines overlap. A similar constraint could be derived for a point where three or more lines intersect. However, identifying these points becomes reliant on the tolerances used in the calculations, and such cases are unlikely to occur in practical situations. As such, this extension will not be considered further in this paper. Note that the general case of three members intersecting at three points (i.e. forming a triangle) is handled correctly by the constraints of (6a,b).

When the constraints of (6a,b) are implemented using lazy constraints, the size of $\bar{\mathbf{v}}$ can be greatly reduced. $\bar{\mathbf{v}}$ becomes a pool of variables, which are assigned to crossovers as required. The pool size should be chosen to

be larger than the number of lazy constraints expected to be added, but small enough to not require excessive memory. For the problems shown here, a pool size of 100 was found to be sufficient. In this case, the variables in $\bar{\mathbf{v}}$ can be set to zero without affecting the optimality of the problem as initially provided; this affects the problem in a similar manner to the variables in $\mathbf{w}$.

Algorithm 1 Callback function to limit number of joints, including 'crossover joints' (pseudocode).

 # Values from solver for newly found solution:
list of bar member areas, *areas*
 # Data stored from problem setup:
nodes attached to each member, *connectivity*
coordinates for each node, *nodeLocations*
list of pointers to $\bar{\mathbf{v}}$ variables, *variablePool*
list of pointers to $\mathbf{w}$ variables, *flagVars*
 # Persistent data throughout an optimization run:
index of most recently used value of $\bar{\mathbf{v}}$, $r = 0$
pairs of members for each lazy constraint, *poolPairs* = [ ]
Function called periodically during solver execution:
function CALLBACK( )
  if new integer solution identified **then**
    realBars $\leftarrow \{i \,|\, areas(i) > 0\}$
    for each pair $\{p, q\}$ from *realBars* **do**
      # Calculate if bars p and q intersect using data from *connectivity* and *nodeLocations*:
      if p and q intersect & $\{p, q\} \notin poolPairs$ **then**
        $r = r + 1$
        # Add the constraint:
        $flagVars[p]+flagVars[q]+variablePool[r] \leq 1$
        append $\{p, q\}$ to *poolPairs*
      end if
    end for
  end if
  # If no lazy constraints added, the solution is accepted as incumbent
end function

Initial tests suggested that it was more advantageous to check and impose these lazy constraints each time a feasible integer solution was identified, rather than each time a continuous relaxation was solved. This also reduced the number of times the check was performed, and meant a smaller pool of constraints could be used. This approach was therefore adopted here. As the solution is not changed by the proposed method, these heuristics impact only the speed with which the solution is obtained, and not the solution itself.

The procedure used to dynamically generate these constraints is shown in Fig. 2 and Algorithm 1. The process to instead forbid crossovers is similar, except that

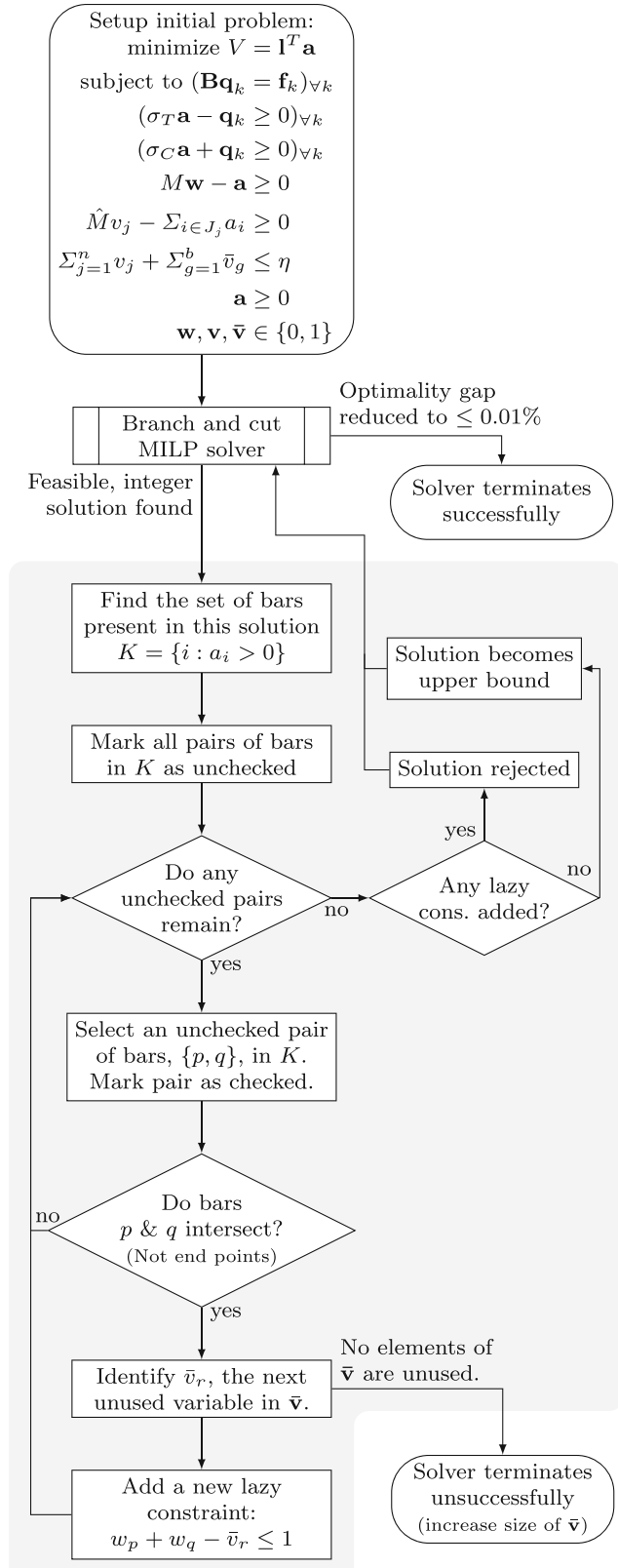


Fig. 2 Procedure for runtime constraint generation. Problem shown involves imposing a limit on the number of joints, including 'crossover joints'. The steps in the shaded region are performed within the callback function which is called by the MILP solver (e.g. Gurobi)

all references to $\bar{\mathbf{v}}$ are removed, and the newly added lazy constraint is instead $w_p + w_q \leq 1$.

In the procedure, once a constraint has been added to the current reduced problem, it will not be removed. This ensures that the solution obtained by successively adding dynamically generated constraints will converge to the globally optimal solution for the original full problem (i.e. the problem that includes all constraints from the outset). This will occur when the solution for the current reduced problem, comprising only a subset of all possible constraints, is also found to be feasible for the original full problem. For continuous linear optimization problems, a similar principle underpins the cutting plane method (Kelley 1960) and analogous column generation method (Dantzig and Wolfe 1960), which has been successfully used by Gilbert and Tyas (2003) to develop a computationally efficient 'member adding' procedure for large-scale truss layout optimization problems.

Finally, note that the optional geometry optimization post-processing step (see Section 3) is not shown in Fig. 2 and Algorithm 1, and is performed following a successful termination of the solver.

2.2.3 Imposing symmetry

For a given problem with symmetrical design domain, loads and supports, it is known (Stolpe 2010) that truss optimization with discrete cross sections may have an optimal solution that is not symmetrical; it will be shown that this is also the case when the MILP problem formulation proposed here is used to impose limits on the number of joints in the structure.

However, it is useful to consider how a requirement for a symmetrical solution can be imposed as an additional constraint, since symmetry will often be preferred for reasons of standardization or aesthetics. It also allows problem size to be significantly reduced, as only half of the design domain needs to be explicitly modelled. To impose a symmetry condition, each symmetrical pair of members is assigned only a single area variable. Additionally, only one integer flag is added to each symmetrical pair of members or nodes. Members that cross the defined line of symmetry, or 'mirror plane', are not included, as they can be approximately modelled using nodes that lie on the mirror plane, as shown in Fig. 3. Thus, the number of potential members (and therefore variables) will be reduced to approximately a quarter of the initial number.

To achieve this, some modifications to the constraints are needed. A node which lies on the mirror plane and which is connected only to members which are perpendicular to the mirror plane will not appear to be a joint in the final design. Therefore, for nodes lying on the mirror plane, (3a) is replaced by

$$\hat{M} v_j - \sum_{i \in J'_j} a_i \geq 0 \qquad j = 1, 2, ..., n \tag{7}$$

where J'_j is the set of member indices for members connected to node j, but not including those members which are perpendicular to the mirror plane.

The constraint on the total number of joints must also be modified such that joints that are formed by crossovers or nodes lying remote from the mirror plane are counted twice in the computations.

2.2.4 Limits on the angle between members

A feature that can make Michell structures difficult to manufacture is the presence of small angles between adjacent members, especially in fan-type regions. To prevent this, integer constraints can be added in the layout optimization stage as follows:

$$(w_h + w_i \leq 1)_{\forall \{h,i\} \in D} \tag{8}$$

where $D = \{\{h_1, i_1\}, \{h_2, i_2\}, ...\}$ is the set containing unordered pairs of indices for all pairs of members that form an angle that is smaller than μ, the minimum permitted joint angle. Note that this angle may be formed either at a node which is common to both members or at a point where the members intersect, partway along one or both of their lengths.

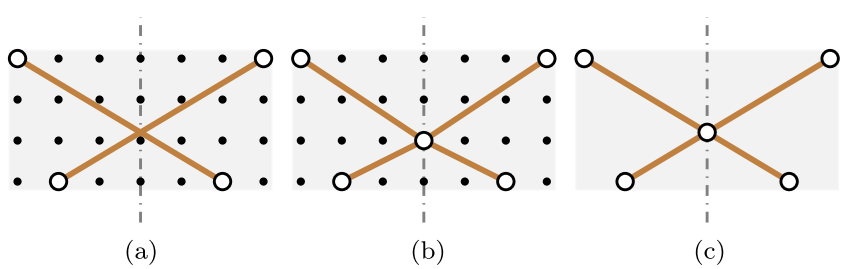


Fig. 3 Approximation of members crossing a mirror plane in part of a layout optimization solution: **a** members crossing the mirror plane (shown as a dash-dotted line) permitted, noting that the crossover joint will be counted due to (6a); **b** members crossing the mirror plane not permitted, with a node on the mirror plane used to approximate (**a**); **c** geometry optimization used improve (**b**), with the node on the mirror plane in this case moving to the same location as in (**a**)

Again, the full formulation is very time consuming to compute as the number of these constraints is approximately proportional to n^4. These constraints are therefore also implemented as lazy constraints. The procedure is similar to that outlined in Fig. 2; however, references to $\bar{\mathbf{v}}$ are removed, and the check on each pair of members, p & q, is to identify if they form an angle (either at a crossover point or at an end node) which is outside the permitted range.

3 Geometry optimization post-processing step

Once the integer programming stage has been completed, the resulting structure can be further refined by the use of the geometry optimization post-processing rationalization method developed by He and Gilbert (2015). This adds the nodal positions of the structure as design variables, resulting in a non-linear and non-convex problem. This stage is optional as the structure generated by solving the MILP problem will satisfy all the specified design constraints. However it is attractive as it further reduces the volume and may allow results which match analytically derived solutions from the literature to be obtained.

As the geometry optimization problem is non-convex, it is not generally possible to solve this to a guaranteed globally optimal solution; thus this method relies on the starting point provided by the topology optimization being sufficiently close to the optimal point. A finding of this study is that the solution of the MILP problem can successfully be used as the starting point for geometry optimization.

During the geometry optimization stage, it will be important to ensure that the solutions continue to be feasible with respect to the practical constraints imposed in the previous sections. The remainder of this section will outline the required modifications to the algorithm proposed by He and Gilbert (2015). Note that the original buildability constraints, involving integer variables, need not be included at this stage, as the overall topology is generally not significantly changed. Instead the buildability constraints are reformulated as constraints on the nodal positions, leading to a non-linear but continuous problem.

When a limited number of joints are permitted, no additional constraints will be required in the geometry optimization (GO) stage. This is because the number of joints in the problem generally remain constant, or will reduce due to joint merging, or due to all connected members at a joint vanishing.

The only possible means by which new joints can appear is if the topology is changed by a member passing over another joint, as shown in Fig. 4. However, this situation can easily be avoided by converting the topology between the MILP and GO stages such that all 'crossover joints' become standard joints.

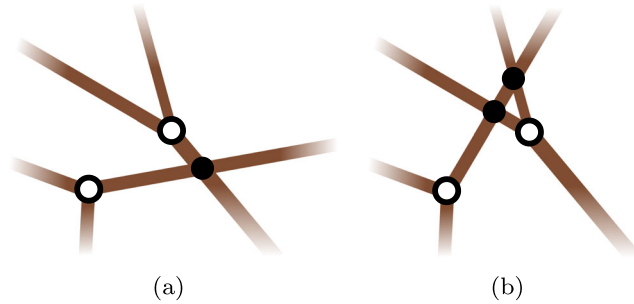


Fig. 4 Detail of a layout, showing the importance of converting 'crossover joints' to standard joints between the MILP and geometry optimization post-processing stages when imposing a limit on the total number of joints: **a** before geometry optimization, containing 3 joints in this region; **b** after geometry optimization, now containing 4 joints in this region (due to a 'crossover' joint not first being converted to a standard joint)

When a limit on the angle between members has been imposed, this must be converted to a continuous constraint on the joint coordinates. This will be in the form:

$$\frac{\overrightarrow{\mathrm{CA}} \cdot \overrightarrow{\mathrm{CB}}}{|\overrightarrow{\mathrm{CA}}||\overrightarrow{\mathrm{CB}}|} \leq \cos(\mu) \tag{9}$$

where C is the joint common to two members, A and B are the other joints of each member, and μ is the imposed minimum angle. Note that this requires point C to be a standard joint; therefore, 'crossover joints' must

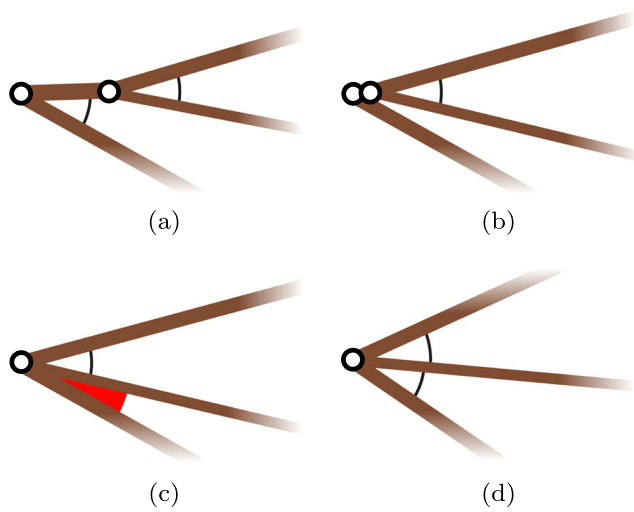


Fig. 5 Detail of area containing branched members during the geometry optimization post-processing step: **a** initial branched member topology; **b** joints moving closer together, though as the bottom and middle member do not meet, the angle between them is not checked; **c** joints are merged, but the bottom and middle members now have an angle constraint, which is violated; **d** the feasibility restoration phase finds a point at which all constraints are satisfied, but the geometry is now quite different to the MILP starting point

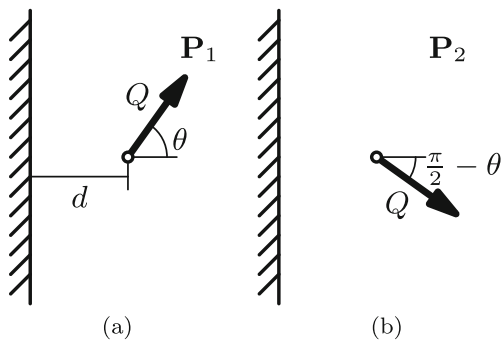


Fig. 6 Simple cantilever: details of applied load cases, **a** load case $\mathbf{P}_1$, and **b** load case $\mathbf{P}_2$

again be converted to standard joints before the geometry optimization stage.

In some cases where a minimum angle is imposed, additional parameters may be required for the geometry optimized results to be meaningful. For example, when a branch-type form (as shown in Fig. 5a) is identified as the optimal solution of the integer programming problem, the branch joint tends to move towards the 'root' joint, as in Fig. 5b. The joint merging phase will then combine the two joints, and there then may be pairs of members which violate the angle constraint (Fig. 5c). If the permitted movement radius of the joints is sufficiently large, the feasibility restoration phase of the non-linear solver is likely to be able to find a new feasible solution using the new topology (Fig. 5d).

This new form may now be notably different from the initial starting point, and therefore potentially inefficient. However, simply eliminating the joint merging step would result in a final design where two joints were infinitesimally close together. A minimum member length constraint can be added to prevent this, i.e.

$$|\overrightarrow{\mathrm{AB}}| \geq l_{\min} \tag{10}$$

where A and B are the end points of the member, and $l_{\min}$ is the minimum permitted length. This constraint is only explicitly required in the geometry optimization stage, since during layout optimization, short members can simply be omitted from the ground structure. Here, length limits have generally been added only in the geometry optimization stage, with $l_{\min}$ being set at or below the nodal spacing of the original ground structure.

Within the examples of this paper, it has been observed that the geometry optimization procedure did not need to make use of several considerations of He and Gilbert (2015), due to the simplified nature of the starting structures. For example, new crossover joints were added only at the initialization of geometry optimization. Also, the joint merging procedure was triggered only in cases with branching type structures. From this, it can be seen that the structures produced by geometry optimization are generally topologically very similar to the solutions obtained at the end of the initial MILP stage.

4 Numerical examples

All numerical example problems were run on an Intel Core i7-6700HQ CPU @ 2.60 GHz, with 16GB of RAM. Gurobi 7.0.1 (2018) was used to solve the MILP problems, with 4 physical cores available for use. All problems mentioned were solved to a 0.01% optimality gap (the default value when using Gurobi). For practical usage, this level of accuracy may not be required, and a higher value can be used to reduce computation time. All other Gurobi parameters were set to their default value. The computational times reported are wall clock times, and include the time taken to set up the problem.

4.1 Simple cantilever

The first example involves the setup shown in Fig. 6. This consists of two load-cases denoted as $\mathbf{P}_1$ and $\mathbf{P}_2$. These each contain a single point load, which are each applied at a given angle θ to the global coordinate axes at point $(d, 0)$, and have magnitude, Q.

Table 1 Simple cantilever: comparison of theoretical and numerical methods for problem with 3 joints

θ	Theoretical			MILP results				MILP + GO results		
	$y_A(d)$	$y_B(d)$	$V\left(\frac{dQ}{\sigma}\right)$	$y_A(d)$	$y_B(d)$	$V\left(\frac{dQ}{\sigma}\right)$	Time (s)	$y_A(d)$	$y_B(d)$	$V\left(\frac{dQ}{\sigma}\right)$
$\frac{\pi}{4}$	0.657	0.657	2.553	0.660	0.660	2.553	10	0.659	0.659	2.553
θ_2	1.308	0.116	2.491	1.300	0.120	2.491	7	1.222	0.174	2.491
	1.222	0.174								
$\frac{3\pi}{8}$	1.117	0.414	2.158	1.120	0.420	2.162	3	1.117	0.414	2.158
$\frac{\pi}{2}$	1.000	1.000	2.000	1.000	1.000	2.000	2	1.000	1.000	2.000

Where y_A and y_b refer to the vertical distance between the height at which the point loads are applied, and the upper and lower support locations respectively. θ_2 is as marked in Fig. 7b (for more information, see Appendix C)

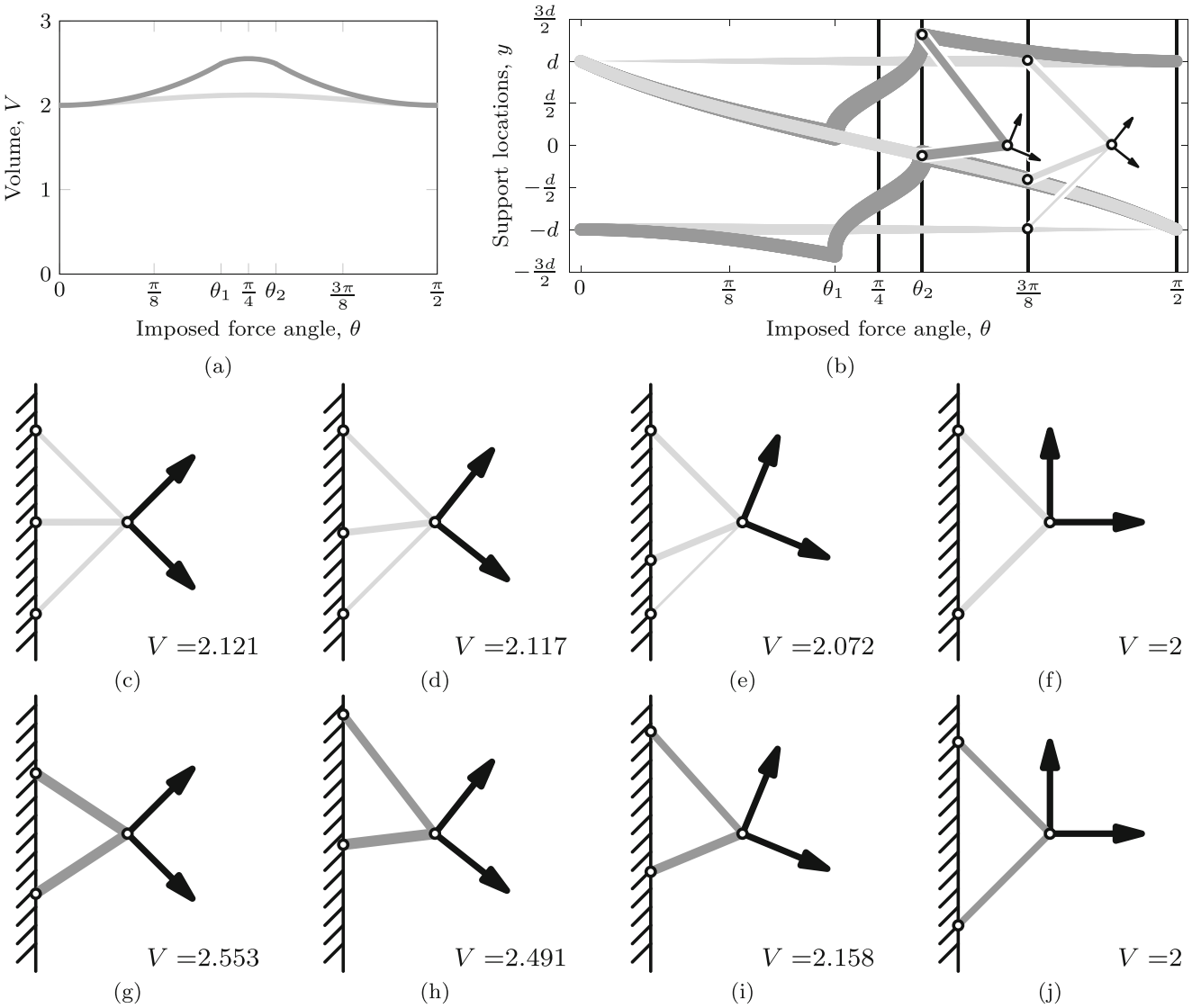


Fig. 7 Simple cantilever: theoretical results, **a** volumes of optimal structures with unlimited complexity (light grey) and maximum 3 joints (dark grey) for values of θ between 0 and $\frac{\pi}{2}$; **b** locations of support points for optimal structures with unrestricted complexity (light grey) and maximum 3 joints (dark grey), with line width proportional to the area of the member which connects there (forms **e** and **h** shown for context); **c–f** forms of optimal structures with unlimited complexity for $\theta = \frac{\pi}{4}, \theta_2, \frac{3\pi}{8}, \frac{\pi}{2}$; **g–j** forms of optimal structures with only 3 joints for $\theta = \frac{\pi}{4}, \theta_2, \frac{3\pi}{8}, \frac{\pi}{2}$

Derivations of the minimum volume structures, both unconstrained and with a maximum of 3 joints, can be found in Appendix C. The optimal member sizes and support locations are shown in Fig. 7.

It can be seen that even for this very simple problem, the behaviour is quite complex and unintuitive. The rationalized structures are not closely linked to the equivalent unrationalized form, which may make such structures difficult to obtain by intuitive or mathematical post processing. The volume penalty caused by imposing the three joint limit is at most 20.3%.

Table 1 shows the difference between the theoretical and numerical results for this example. The MILP results were found with nodes permitted only along the support, at $0.02d$ spacing over $-1.5d \leq y \leq 1.5d$ (i.e. 152 nodes, including the loaded node). As none of the members in the ground structure crossed, crossover constraints were not required. Geometry optimization (GO) was also performed to further refine the results.

The numerical and analytical volumes can be observed to be in close agreement. At $\theta = \theta_2$ a discontinuity occurs, where two distinct results are equally optimal (see Appendix C.2.3). Here, the MILP solver identifies one of these solutions. Then, use of the interior point method for the GO stage perturbs this solution, resulting in identification of the alternative solution.

When a limit on the angle between members is imposed, solutions may be more complex in form, involving members that do not simply directly connect the loaded point and a point on the support. Therefore, numerical solutions for joint angle limits have been calculated using a 5×13 grid of nodes, and a fully connected ground structure ($m = 2080$).

Figure 8 shows results for the problem with angle $\theta = \frac{3\pi}{8}$ radians, i.e. as shown in Fig. 7e and i. The scenario is subjected to limits on the minimum angles between members of 35° and 45°. The unusual topologies identified demonstrate the difficulty in trying to identify optimal solutions for this problem analytically or manually.

The problem was solved firstly by imposing all angle constraints from the outset ('basic formulation'), and then by implementing the angle limit (i.e. Eq. (8)) using lazy constraints. Both implementations produced identical final results; however, it can be seen that the use of lazy constraints reduced the time required by approximately a factor of 20.

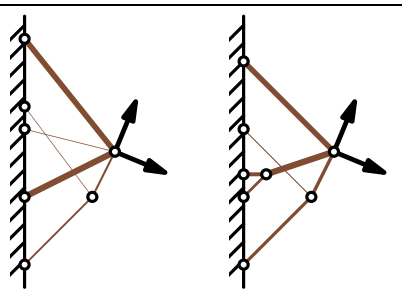
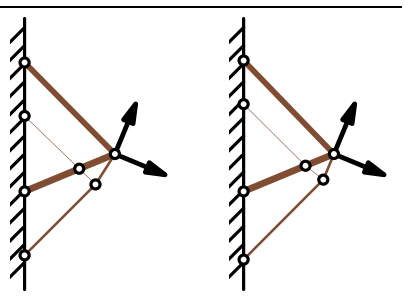
	Minimum angle	35°	45°
Results after MILP			
	Volume	2.160	2.198
	Time (basic formulation)	167 seconds	180 seconds
	Time (lazy constraints)	7 seconds	11 seconds
	Speedup	23.8×	16.3×
Results after MILP + GO			
	Volume	2.082	2.099

Fig. 8 Simple cantilever: numerical results for problem with specified minimum angles between members (load inclination $\theta = \frac{3\pi}{8}$ radians)

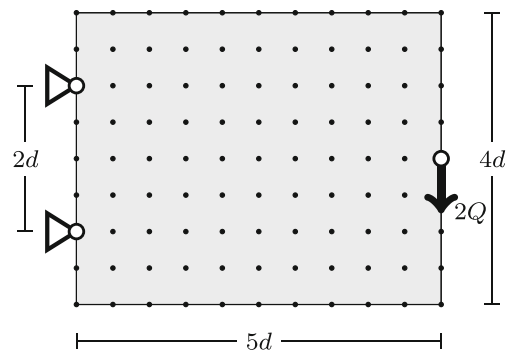


Fig. 9 Michell cantilever: problem specification (fully connected ground structure used)

Figure 8 also shows the result of applying geometry optimization post-processing. Despite the different topologies identified in the MILP stage, similar topologies are found after geometry optimization. Additionally, it can be observed that the volumes after geometry optimization are both lower than the three-joint form shown in Fig. 7i, demonstrating that the more complex topology is beneficial, albeit only by 3.6% and 2.8% respectively. This suggests that, although the geometry optimization process is non-convex and therefore cannot be proven to identify a global optimum, the starting points provided by the MILP formulation appear to be sufficient to provide good results.

4.2 Michell cantilever

4.2.1 Problem specification

The proposed methods are now applied to a classical Michell cantilever problem, as shown in Fig. 9. The theoretical minimum volume can be found using equations derived by Chan (1960) to be $V_T = 39.43Qd$. Discretized versions of this problem have been studied by Prager (1977) and Achtziger and Stolpe (2007). In both cases, the topology

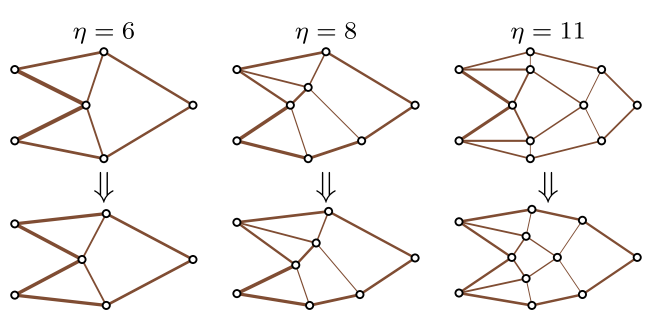


Fig. 10 Michell cantilever: Results after MILP (top) and GO (bottom) stages. The volume difference between MILP and GO solutions is 1%, 1.4% and 1.6% respectively

Table 2 Michell cantilever: results with limits imposed on the total number of joints

Permitted no. of joints	Number of lazy constraints added	Time[a] (s)	Volume after		Optimal for joint costs		
			MILP	MILP+GO			
3	0	10	+31.9%	+31.9%	15.11%	$\leq c_j \leq$	∞
4	0	16	+17.4%	+16.8%	6.25%	$\leq c_j \leq$	15.11%
5	3	74	+10.7%	+10.5%	6.24%	$\leq c_j \leq$	6.25%
6	4	72	+5.3%	+4.3%	0.69%	$\leq c_j \leq$	6.24%
7	4	117	+4.9%	+3.9%		N/A	
8	0	257	+4.3%	+2.9%	0.45%	$\leq c_j \leq$	0.69%
9	3	291	+3.9%	+2.8%		N/A	
10	9	196	+3.6%	+2.6%		N/A	
11	17	133	+3.1%	+1.6%	0.09%[b]	$\leq c_j \leq$	0.45%
12[c]	240	44	+3.0%	+1.5%			
20[d]	—	4	+2.6%	+0.8%			

[a]Elapsed time for MILP stage

[b]Assumes next optimal design is the continuous layout optimization solution; designs with 13-19 joints may increase this value

[c]Result of preventing crossovers without limiting the total number of joints

[d]Continuous layout optimization result, after crossover generation

of the optimal structures was manually inferred from the continuum form; however, their observations are useful for comparative purposes.

A fully connected ground structure of 99 nodes is used here; this contains 4851 potential members. The solution to the standard layout optimization problem at this resolution has a volume of $40.45Qd$, an increase of 2.6% over V_T, although this reduces to +0.8% after GO is applied, and the resulting solution has 20 joints (Table 2).

4.2.2 Limiting the number of joints

To set up the problem with all crossover constraints from the beginning requires checking 11,763,675 pairs of

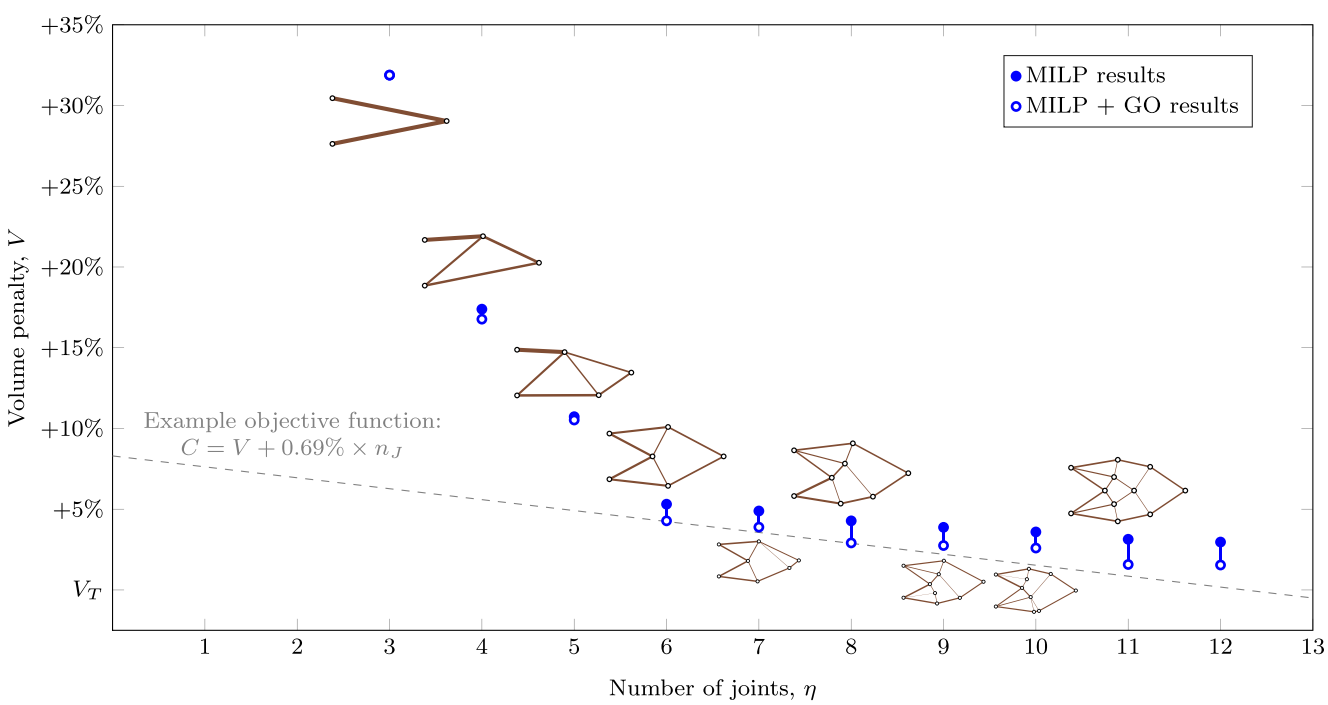


Fig. 11 Michell cantilever: results with limits imposed on the total number of joints. (Volume shown as percentage above theoretical minimum volume, $V_T = 39.43Qd$. Forms shown after MILP and GO. An example cost function is also shown where each joint has a constant cost, equal to an increase in volume 0.7% of V_T.)

Table 3 Michell cantilever example: results with limited joint angle (Volume shown as percentage above theoretical minimum volume.)

Permitted angle between members	Number of lazy constraints added	After MILP			After MILP+GO	
		Time (s)	Min. angle	Volume	Min. angle	Volume
45°	3462	2341	47.7°	+6.3%	45°	+5.4%
40°	1415	2009	47.7°	+6.3%	40°	+4.6%
35°	805	234	37.9°	+5.1%	35°	+4.2%
30°	1377	537	33.7°	+5.0%	30°	+2.5%[a]
25°	665	344	26.6°	+4.4%	25°	+1.8%
20°	349	91	22.8°	+4.3%	21.9°	+1.6%
15°	73	21	18.4°	+3.8%	15°	+1.3%

[a]GO with a length limit of of $0.5d$ (i.e. equal to shortest length in ground structure) gives volume of +2.51%; GO without a length limit and after joint merging and subsequent feasibility restoration gives a volume of +2.55%

members, of which 2,795,779 produce a constraint. (Simply performing these checks was found to take nearly 20 minutes with the C++ code used here.)

Alternatively, using lazy constraints, the problem can be set up almost instantly, producing an initial problem with 14,553 variables and 14,844 constraints. The problem was first solved using lazy constraints to prevent crossovers but without limiting the number of joints. This produced a structure with 12 joints, and required 240 lazy constraints, around 0.01% of the full number.

The problem was then solved with limits imposed on the maximum numbers of joints, η, from 3 to 11. The results can be seen in Table 2 and Figs. 10 and 11, with further details available in Online Resource 1. Note that the longest time taken to solve any of these problems was under 5 minutes, around a quarter of the time needed just to formulate the full problem.

It may be observed that the forms for 3, 6 or 11 joints agree with those found by Prager (1977)[1]. Also the unsymmetrical 8 joint solution is somewhat similar to the unsymmetrical solution presented by Mazurek et al. (2011, fig 22).

4.2.3 Other related problems

Prager extended his results to postulate a solution to the related problem of minimizing total cost, comprising a material cost and a fixed cost per joint. It is possible to reformulate the integer programming problem to consider this directly, by changing the objective function to be of the form

$$\text{minimize } \mathbf{l}^T\mathbf{a} + c_j\mathbf{v} \tag{11}$$

where c_j is the normalized cost of a joint.

[1]Note that Prager (1977) gives a volume of $36.41Qd$ for the 11 joint solution; this can clearly be seen to be incorrect, as it is lower than the minimum value from the equations of Chan (1960). However, the form given by Prager is correct.

However, (11) may alternatively be expressed in the form of an equation, plotted as a straight line on Fig. 11. The example objective function shown on Fig. 11 is a line of constant cost when the joint cost, c_j, is equal to the cost of a volume increase of 0.7% of the minimum volume, V_T. In this case, the solutions with 6 and 8 joints are equally optimal. However, the 7-joint solution has a higher cost; it is therefore not optimal for any objective function in the form of (11).

Prager's solution to this problem over a range of values for c_j consisted of only the 3, 6 and 11 joint solutions. From Fig. 11, this can now be extended to add solutions with 4, 5 and 8 joints. The ranges of joint cost c_j for which each solution is optimal is shown in the final column of Table 2.

Limiting the number of members in a solution is another concern for ensuring practicality. As this is a single load-case problem, and due to the Simplex solver used to solve the LP sub-problems of the MILP problem, the optimal structures identified are all likely to be statically determinate, meaning that the number of joints is directly linked to the number of members (number of members $= 2\eta - 4$). Therefore, this method can also be used as a proxy for limiting the number of members.

4.2.4 Limiting the angles between members

Solutions for the same Michell truss problem, but with imposed minimum angle limits, from 15° to 45° are shown in Table 3 and Fig. 12, with further details provided in Online Resource 2. It can be seen that the topologies shown in Fig. 12 are symmetrical, and several are distinct from those shown in Fig. 11.

In the layout optimization stage, only a limited number of member angles are available; therefore, the structures identified do not have a minimum angle that exactly corresponds to the limit. Generally, once the geometry optimization post-processing step has been applied, the angle limits become active, although this is not always the

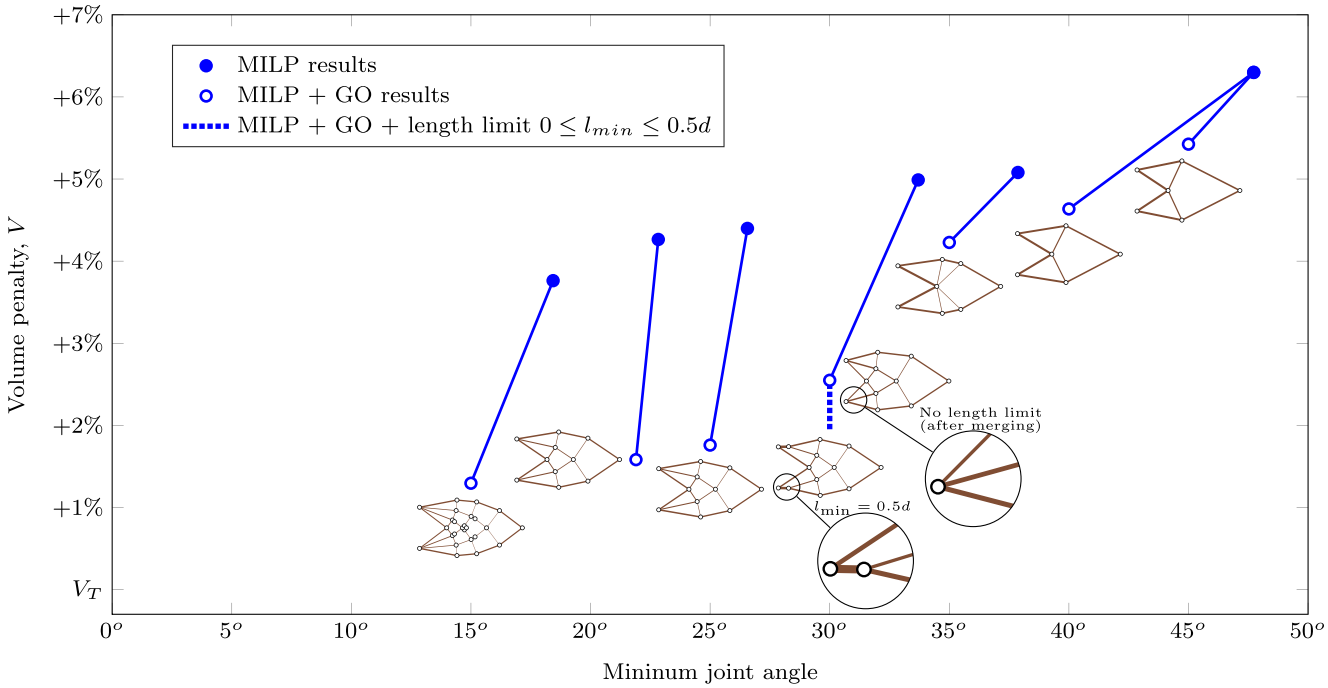


Fig. 12 Michell cantilever: results with limited joint angle. (Volume given as percentage above theoretical minimum volume. Forms shown after MILP and GO. For branched forms, results both with and without a length limit are shown)

case (e.g. in the case of the 20° limit). In some cases, such as the 40° and 45° solutions, the same initial result is identified for multiple angle limits, and the solutions only diverge in the geometry optimization stage.

Most results in Fig. 12 are shown after geometry optimization with no length limit imposed. However, for the limit of 30°, a branching form similar to that shown in Fig. 5 was identified. During the geometry optimization stage for this result, the merging of the 'root' and 'branch' joint occurred, leading to a significant change in topology. The solution reduces in volume as the distance between the branching joints approaches zero, but then increases again in order to ensure compliance with the new angle constraint.

A length limit was therefore imposed to produce more meaningful results. For practical purposes, this is likely to be defined by the same manufacturing process that dictates the minimum angle between members; here $l_{\min}$ will be set at or below the length of the shortest member in the ground structure ($0.5d$). When $l_{\min} = 0.5d$, the new volume was only slightly smaller (+2.51% vs. +2.55%). However, for small values of $l_{\min}$, the volume reduced to +1.9%; this is shown as a dotted line in Fig. 12. Both the results with no length limit and with $l_{\min} = 0.5d$ are illustrated in Fig. 12.

By comparing the results in Tables 2 and 3, it can be seen that the angle limits require a greater number of lazy constraints to be added, leading to correspondingly longer execution times. This is likely to be due to the fact that, when a limit on the number of joint is imposed, the initial constraints significantly reduce the number of feasible integer topologies before any lazy constraints are required. However, the maximum number of lazy constraints added in Table 3 was at most 0.3% of the total number possible (for the 45° limit), showing that the advantage of using lazy constraints is still very significant.

4.3 Spanning example

A more complex, two load-case problem is now considered. This consists of two point loads which are applied separately, and transmitted to a pair of pinned supports; the problem specification is shown in Fig. 13.

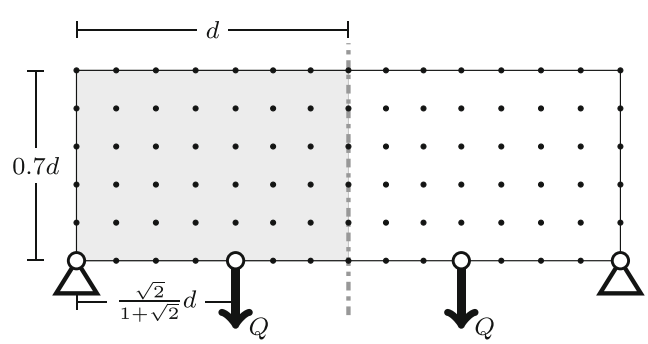


Fig. 13 Spanning example: problem specification, after Sokół and Rozvany (2013) (The two point loads are applied separately. Grey shading shows the area modelled when the symmetry condition is imposed.)

Table 4 Spanning example: results with symmetry condition imposed and limits on numbers of joints (Bold values highlight when the two methods produce different results)

Permitted number of joints	Counting 'crossover joints'				Preventing 'crossover joints'			
	No. of lazy cons. added	Time[a] (s)	Volume after		No. of lazy cons. added	Time[a] (s)	Volume after	
			MILP	MILP+GO			MILP	MILP+GO
5	0	7	+59.5%	+58.1%	0	6	+59.5%	+58.1%
7	0	56	+8.8%	+8.4%	0	31	+8.8%	+8.4%
9	7	52	+5.9%	+5.1%	14	77	+5.9%	+5.1%
11	26	345	+5.2%	+4.1%	33	154	+5.2%	+4.1%
13	24	528	**+4.5%**	+3.3%	154	446	**+4.5%**	+3.3%
15	28	436	**+3.5%**	**+2.7%**	48	184	**+3.7%**	**+2.5%**
17	40	504	**+3.4%**	**+2.3%**	444[b]	455[b]	**+3.5%**	**+2.4%**
19	76	8150	+3.1%	+2.4%				
21	68	6833	+3.0%	+2.0%				
31	90	138265	+2.7%	+1.8%				
45[c]	–	6	+2.3%	+1.2%				

[a]Elapsed time for MILP stage

[b]Refers to forbidding crossovers with no limit on number of joints, which produces a structure with 17 joints

[c]Layout optimization result, which produces structure with 45 joints

This is a symmetrical problem, and therefore the minimum volume structure is also symmetrical when no discrete buildability constraints are imposed. The minimum volume solution, V_T, is given by Sokół and Rozvany (2013) as $3.44363\frac{Qd}{\sigma}$. The design domain is discretized using a grid of 90 nodes. The layout and geometry optimization solution for this resolution had a volume 1.2% greater than the theoretical optimal value.

The problem was first solved without imposing a requirement for a symmetrical solution. Solutions with maximum numbers of joints, η, ranging from 5 to 9 were found. Solutions with odd numbers of joints were found to be symmetrical, and were equal to the corresponding results shown in Fig. 15 and Table 4. However, the solutions with 6 and 8 joints were asymmetric, as shown in Fig. 14. Note that the 8 joint example approximately consists of one-half from each of the topologies with 7 and 9 joints.

Due to these findings, and the general preference in practice for symmetrical designs, the model was modified to explicitly impose a symmetry condition about the centre line, using (7) and the method outlined in Section 2.2.3. Solutions were again sought for 6 and 8 joints; however, the optimal solutions were found to be identical to the solutions for 5 and 7 joints respectively. This demonstrates that the lack of a symmetrical optimal solution, a characteristic previously noted in the solutions of truss optimization problems with discrete cross sections, is also a characteristic of the problem with continuous cross sections when limits are imposed on the numbers of joints.

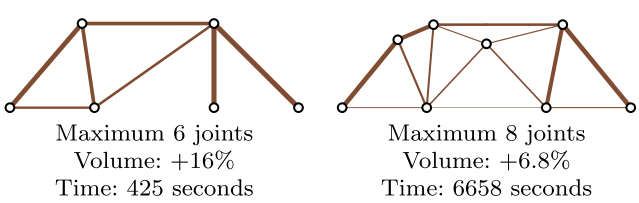


Fig. 14 Spanning example: results after MILP and GO showing asymmetric optimal solutions

Results for various numbers of joints are given in Table 4 and Fig. 15, with further details provided in Online Resource 3. The constraints (5a,b) (to prevent 'crossover joints') and (6a,b) (to include 'crossover joints' in the total number of joints to be limited) have both been tested. When 'crossover joints' are not permitted, only structures with up to 17 joints can be identified; results in the range $17 < \eta < 45$ can only be identified by allowing 'crossover joints' and explicitly including them in the total limit.

The problem of including the 'crossover joints' is a more relaxed version of the problem where 'crossover joints' are prevented. Therefore, solutions from the MILP problems that take account of crossovers must be at least as good as solutions found when these are prevented. However, the geometry optimization stage is non-linear, and therefore local optima may result, depending on the initial point provided. It can be seen that for the 15 and 17 joint solutions, local optima have been identified; both methods

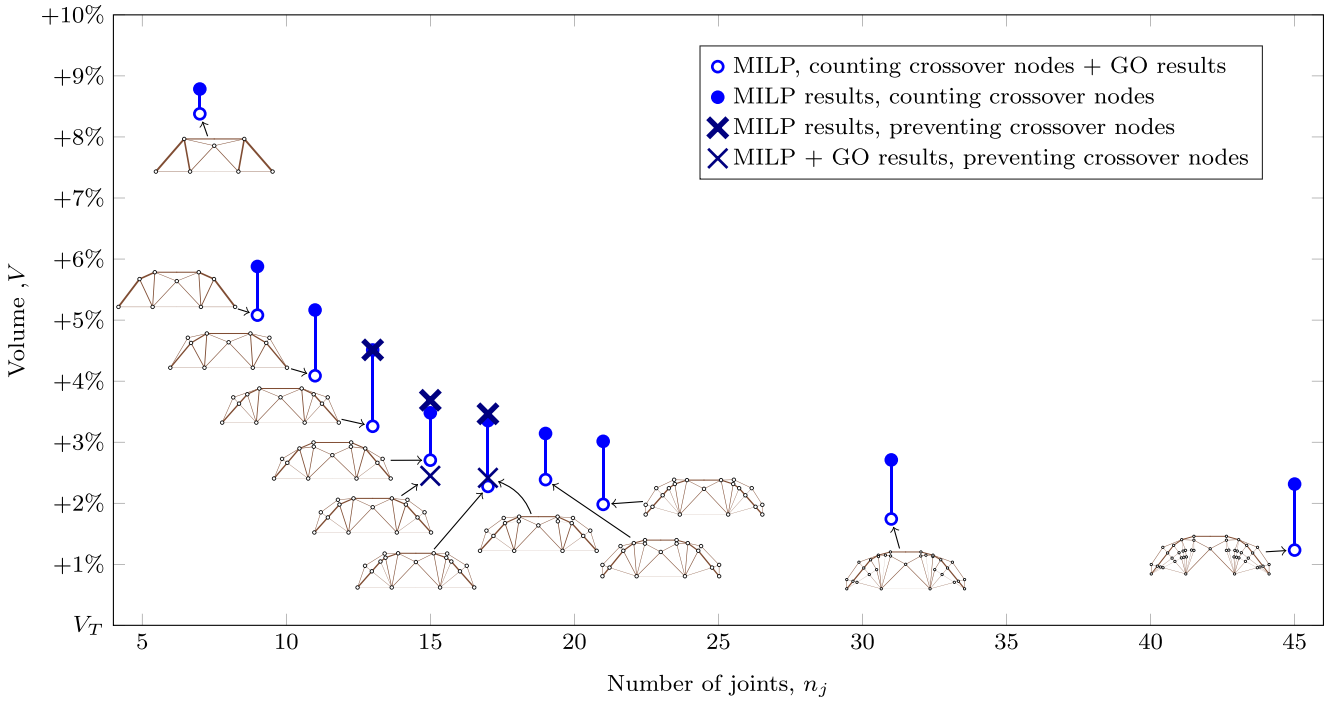


Fig. 15 Spanning example: results with limited numbers of joints (Volume shown as percentage above theoretical minimum volume. Forms shown are after MILP and GO, with symmetrical solutions required. Results preventing crossover joints are shown only where they differ from the result of counting the crossover joints)

appear to be susceptible to this. However, the volume difference is less than 0.2%, demonstrating that at this point, multiple solutions of similar volume and complexity are available, any one of which would likely be suitable for practical application. Many commercial solvers (e.g. IBM Corp 2015; Gurobi Optimization LLC 2018) provide the ability to record a pool of nearly optimal solutions, which may be of use in addressing this issue.

As a multiple load-case problem, the solutions identified are generally not statically determinate, and therefore there is not a direct relationship between number of members and number of joints. However, Fig. 16 shows that there is still a very strong correlation between the number of members and the number of joints. Therefore, for practical purposes, this method is still likely to produce useful results when structures with few members are desired.

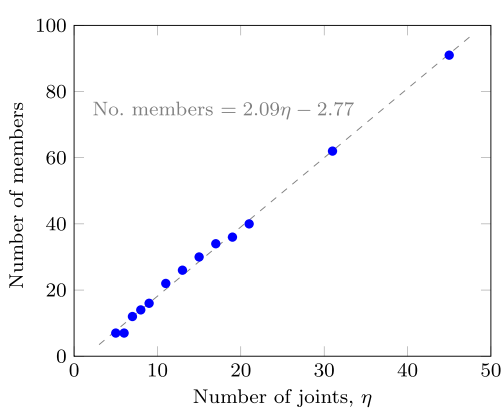


Fig. 16 Spanning example: number of joints and members in solutions where the number of joints has been limited, also showing best fit line. ($R^2 = 0.997$)

4.4 Commentary

The numerical examples described here have demonstrated the applicability of the method to single and multiple load case problems in 2D. The method described could also be immediately applied to 3D problems, if crossovers are considered to occur at points where the centrelines of two members intersect exactly, or to within some predefined tolerance. Some modification of the approach described would be necessary in order to prevent the outer faces of members intersecting, taking into account the chosen member cross section form. As is generally the case with layout optimization methods, a greater number of nodes would be required to fill a 3D domain to a similar density compared to a 2D domain, increasing the computational requirements.

The method has proved effective at identifying simple truss structures. However, from a practical point of view, the simplest structure may not be a truss. For example, in the case of the spanning example considered in Section 4.3, a single beam along the base of the domain would

generally be considered to provide a simpler, albeit less efficient, solution. When bending is involved the chosen cross-section form needs to be taken into account, and the associated numerical formulation is somewhat more complex. However, many of the principles described herein are still applicable in this case.

5 Conclusions

It was found that the use of dynamically generated lazy constraints can significantly reduce the computational time required to solve layout optimization problems with discrete buildability constraints. Specifically, the use of lazy constraints permitted 'crossover joints' to be dealt with in a computationally efficient manner. Improvements in speed of over a factor of 20 were observed for relatively small problems; this difference is likely to increase further as problem size increases. This allows the proposed method to be used for problem sizes that would be intractable using the standard formulation.

Rationalized structures with limited numbers of joints or limited angles between adjacent members have been identified for a range of problems, including those with multiple load-cases, using a two-stage process incorporating a layout optimization stage and a geometry optimization stage. Using this process, results were found to agree with analytically derived results from the literature, suggesting that the proposed separation of topology and geometry/shape optimization is effective, and that MILP solutions are suitable starting points for a non-linear optimization stage.

The rationalized structures were often found to have a volume within a few percent of the corresponding minimum volume Michell structure, whilst being far more feasible to construct. A number of interesting features of these solutions have been observed:

- Symmetrical problems do not always have symmetrical optimal solutions when limits on the numbers of joints are imposed. Therefore, the decision to use symmetry to reduce the computational expense of a problem should be made with care.
- Multiple optimal or near optimal solutions are possible. Many numerical methods, such as geometry optimization, will identify only one of these, although there may be many that would be acceptable for practical use.
- When the angle between adjacent members is limited, 'branching' type structures may occur. This may then require the addition of a minimum length constraint to produce practical results.
- When the number of joints is limited, it was found that the number of members in the solution was strongly linked to the number of joints. This may provide a computationally efficient proxy problem.

Acknowledgements The authors would like to thank LimitState Ltd and Linwei He for access to code implementing the geometry optimization procedure.

Funding information The financial support of Expedition Engineering and EPSRC (under grant number EP/N023471/1) is gratefully acknowledged.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

Replication of results A detailed derivation of the analytical results mentioned in the paper can be found in Appendix C.

Details of the results found by the proposed numerical method and shown in Figs. 11, 12, 14 and 15 are available in the Online Resources.

Implementation of the proposed numerical method was undertaken using the C++ interface of Gurobi.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommonshorg/licenses/by/4.0/.

Appendix A: Comparison of problem formulations

Several equivalent formulations are possible to ensure that the flag variables $\mathbf{v}$ accurately represent the existence of each node. The method that is used within this paper links the value of v_j to the sum of the members connected to node j using (3a), reproduced here:

$$\hat{M} v_j - \sum_{i \in J_j} a_i \geq 0 \qquad j = 1, 2, ..., n \tag{12}$$

This formulation will be referred to as formulation $\mathcal{A}$.

An alternative formulation,z which may be considered to be more standard in the general integer programming community, is to link the value of v_j to the flag variables of the members connected to node j:

$$N v_j - \sum_{i \in J_j} w_i \geq 0 \qquad j = 1, 2, ..., n \tag{13}$$

In this formulation, the arbitrary large number, $\hat{M}$, is replaced by N, the maximum number of members which will be permitted to connect to any node. This formulation will be referred to as formulation $\mathcal{W}$.

Table 5 Speed comparison between formulations for defining the node flag variables, **v**

Permitted	Formulation $\mathcal{A}$		Formulation $\mathcal{W}$	
number of joints	No. of lazy cons. added	Time (s)	No. of lazy cons. added	Time (s)
3	0	10	0	14
4	0	16	0	58
5	3	74	8	137
6	4	72	0	95
7	4	117	7	222
8	0	257	0	686
9	3	291	1	571
10	9	196	10	376
11	17	133	17	224

Problem is the Michell cantilever with limited number of joints, as shown in Table 2 and Fig. 11

For both $\mathcal{A}$ and $\mathcal{W}$, the remainder of the formulation is as described in Section 2.2. The two formulations produce identical solutions; however, the computational requirements may differ.

To investigate this, both formulations have been used to test the Michell cantilever problem with a limit imposed on the number of joints. Full results using form $\mathcal{A}$ are given in Table 2. A comparison of the speed of the two formulations is given in Table 5.

It can be seen that formulation $\mathcal{A}$ takes roughly half the time to solve the problems compared to formulation $\mathcal{W}$. This is as expected if the characteristics of the two formulations prior to the addition of any lazy constraints is considered. Formulation $\mathcal{A}$ initially begins with all the member flags, **w**, unconstrained; i.e., they may all be set equal to 1 without making any potential solution infeasible. It only becomes necessary to begin to branch on any variable w_i once the member i is part of an added crossover constraint.

In contrast, formulation $\mathcal{W}$ couples all integer variables from the outset, leading to a much more challenging initial problem. This outweighs the potential benefits of eliminating $\hat{M}$.

Formulation $\mathcal{A}$ has therefore been used to generate all results contained in the main body of the present paper.

Note that the findings in this section apply only to problems where the number of joints is limited, as limiting the angle between members does not require the presence of node flags, v_j.

Appendix B: Full problem statements

For clarity, full problem statements for the problems solved in this paper are given here. Symbols are as defined in the

Table 6 Symbols used in problem statements

Symbol	Description
$\mathbf{a} = [a_1, a_2, ..., a_m]$	Vector of member cross sectional areas
$\mathbf{B}$	Matrix of member direction cosines
b	Total number of intersections between potential members
D	Set containing all pairs of potential members which form an angle of less than μ
f_k	Vector of external forces at nodes in loadcase k
g	Index of a potential 'crossover joint'
h, i	Indices of potential members
j	Index of ground structure node
k	Index of load-case
$\mathbf{l} = [l_1, l_2, ..., l_m]$	Vector of member lengths
$M, \hat{M}$	Large numbers
m	Number of potential members in the ground structure
n	Number of nodes in the ground structure
$\mathbf{q}_k$	Vector of member internal forces in loadcase k
$\mathbf{v} = [v_1, v_2, ..., v_n]$	Vector of flags indicating existence of ground structure nodes
$\bar{\mathbf{v}} = [v_1, v_2, ..., v_b]$	Vector of flags indicating existence of joints at crossovers
$\mathbf{w} = [w_1, w_2, ..., w_n]$	Vector of flags indicating existence of members
X	Set containing all pairs of potential members which intersect
σ_T, σ_C	Allowable stress in tension and compression respectively
η	Number of joints permitted
μ	Minimum permitted joint angle

main body of the paper, and are also summarized in Table 6. Firstly, the problem to restrict the number of joints in a solution, whilst preventing crossover nodes, is given by:

$$\begin{aligned}
\text{minimize } V &= \mathbf{l}^T\mathbf{a} && (14a)\\
\text{subject to } (\mathbf{B}\mathbf{q}_k &= \mathbf{f}_k)_{\forall k} && (14b)\\
(\sigma_T\mathbf{a} - \mathbf{q}_k &\geq 0)_{\forall k} && (14c)\\
(\sigma_C\mathbf{a} + \mathbf{q}_k &\geq 0)_{\forall k} && (14d)\\
M\mathbf{w} - \mathbf{a} &\geq 0 && (14e)\\
\hat{M}v_j - \sum_{i\in J_j} a_i &\geq 0 \qquad j = 1, 2, ..., n && (14f)\\
\sum_{j=1}^{n} v_j &\leq \eta && (14g)\\
(w_h + w_i &\leq 1)_{\forall\{h,i\}\in X} && (14h)\\
\mathbf{a} &\geq 0 && (14i)\\
w_i &\in \{0, 1\} \qquad i = 1, 2, ..., m && (14j)\\
v_j &\in \{0, 1\} \qquad j = 1, 2, ..., n && (14k)
\end{aligned}$$

The problem initially supplied to the solver is as above but excluding the constraints of (14h), which are generated as required during the running of the solver.

The problem to count 'crossover joints' as contributing to the limiting number of joints is fully stated as:

$$\begin{aligned}
\text{minimize } V &= \mathbf{l}^T\mathbf{a} && (15a)\\
\text{subject to } (\mathbf{B}\mathbf{q}_k &= \mathbf{f}_k)_{\forall k} && (15b)\\
(\sigma_T\mathbf{a} - \mathbf{q}_k &\geq 0)_{\forall k} && (15c)\\
(\sigma_C\mathbf{a} + \mathbf{q}_k &\geq 0)_{\forall k} && (15d)\\
M\mathbf{w} - \mathbf{a} &\geq 0 && (15e)\\
\hat{M}v_j - \sum_{i\in J_j} a_i &\geq 0 \qquad j = 1, 2, ..., n && (15f)\\
\sum_{j=1}^{n} v_j + \sum_{g=1}^{b} \bar{v}_g &\leq \eta && (15g)\\
(w_h + w_i - \bar{v} &\leq 1)_{\forall h,i\in X} && (15h)\\
\mathbf{a} &\geq 0 && (15i)\\
w_i &\in \{0, 1\} \qquad i = 1, 2, ..., m && (15j)\\
v_j &\in \{0, 1\} \qquad j = 1, 2, ..., n && (15k)\\
\bar{v}_g &\in \{0, 1\} \qquad j = 1, 2, ..., b && (15l)
\end{aligned}$$

When implemented using lazy constraints, the constraints of (15h) are omitted from the initially provided problem, and generated generated as required during the running of the solver. In addition, b may be reduced to a value less than the total number of potential crossover points.

The problem of eliminating small angles between members can be stated as:

$$\begin{aligned}
\text{minimize } V &= \mathbf{l}^T\mathbf{a} && (16a)\\
\text{subject to } (\mathbf{B}\mathbf{q}_k &= \mathbf{f}_k)_{\forall k} && (16b)\\
(\sigma_T\mathbf{a} - \mathbf{q}_k &\geq 0)_{\forall k} && (16c)\\
(\sigma_C\mathbf{a} + \mathbf{q}_k &\geq 0)_{\forall k} && (16d)\\
M\mathbf{w} - \mathbf{a} &\geq 0 && (16e)\\
(w_h + w_i &\leq 1)_{\forall\{h,i\}\in D} && (16f)\\
\mathbf{a} &\geq 0 && (16g)\\
w_i &\in \{0, 1\} \qquad i = 1, 2, ..., m && (16h)
\end{aligned}$$

When implemented using lazy constraints, the problem is initially provided to the solver omitting constraints (16f), which are generated as required during the running of the solver.

Appendix C: Derivation of global solutions for simple cantilever problem

C.1 Minimum volume solution

To provide a global solution for validation, a simple problem is considered. This consists of 2 load-cases denoted as $\mathbf{P}_1$ and $\mathbf{P}_2$. These each contain a single point load, which are both applied at the point with coordinates $(d, 0)$, and with the same magnitude, Q. The two loads are applied orthogonally, and the load in $\mathbf{P}_1$ is at an angle of θ to the horizontal (Fig. 17a, b), cases where $0 \leq \theta \leq \frac{\pi}{2}$ will be considered. Two special cases of this, with $\theta = 0$ and $\theta = \frac{\pi}{4}$, were studied by Rozvany et al. (2014).

First the component load-cases are calculated; the sum component load-case[2] $\mathbf{P}_1^* = (\mathbf{P}_2 + \mathbf{P}_1)/\sqrt{2}$ contains a point load of magnitude Q and inclined at an angle of $\theta - \frac{\pi}{4}$ (Fig. 17c). The difference component $\mathbf{P}_2^* = (\mathbf{P}_2 - \mathbf{P}_1)/\sqrt{2}$ is also of magnitude Q and inclined at an angle of $\theta - \frac{3\pi}{4}$ (Fig. 17d).

The solution for $\mathbf{P}_1^*$ consists of a single member inclined at the same angle as the force (Fig. 17e), i.e. connecting to the support at $y = -d\tan(\theta - \frac{\pi}{4})$. The member has an internal force of $-Q$. The length of the member is $\frac{d}{\cos(\theta - \frac{\pi}{4})}$. Therefore, the component volume is

$$V_1^* = \frac{Qd}{\sigma\cos(\theta - \frac{\pi}{4})} \tag{17}$$

For $\mathbf{P}_2^*$, the external load is again of magnitude Q and its direction varies by $\frac{\pi}{4}$ either side of vertical, the solution to this was given my Rozvany et al. (1995). This consists of

[2]Note that here we use the component load form of Rozvany and Hill (1978) which allows for generalizations to more than two load cases.

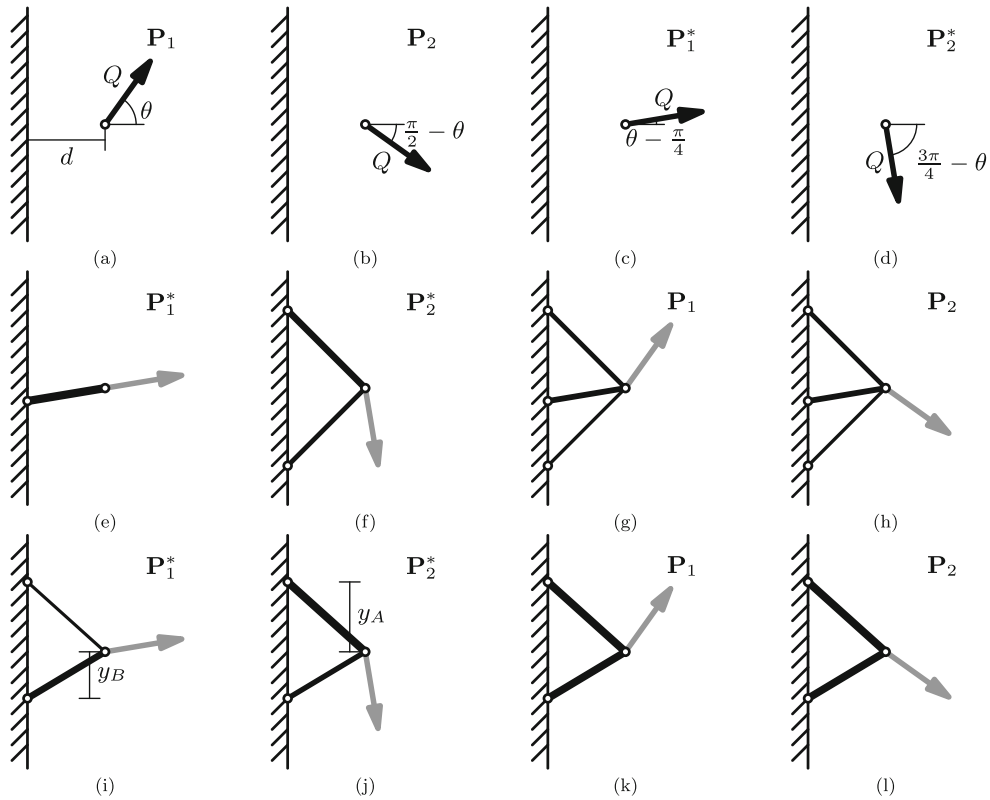


Fig. 17 Simple cantilever: problem specification and component load-cases, **a**–**b**, imposed load-cases; **c**–**d**, component load-cases; **e**–**f** component solutions with unlimited complexity; **g**–**h** final optimal solution with unlimited complexity; **i**–**j** component solutions with only 2 members. **k**–**l** final solution with only 2 members

2 symmetrical, orthogonal members, which connect to the support at heights $y = \pm d$ (Fig. 17f). The length of each member is $d\sqrt{2}$. The internal force in the top member is $Q \sin\theta$ and in the bottom member is $Q \cos\theta$. Therefore, the volume in this component is

$$V_2^* = \frac{Qd}{\sigma}\sqrt{2}\cos\theta + \sqrt{2}\sin\theta \tag{18}$$

By the superposition principle, these two component solutions are combined to give the optimal design (Fig. 17g–h). The total volume is given by:

$$V = \frac{V_1^* + V_2^*}{\sqrt{2}} = \frac{Qd}{\sigma}\left(\frac{1}{\sqrt{2}\cos(\theta - \frac{\pi}{4})} + \cos\theta + \sin\theta\right) \tag{19}$$

As there are no co-incident members, the final member areas are given by dividing the component areas by $\sqrt{2}$.

C.2 Limited complexity solution

C.2.1 Internal member forces

It is now required to find the minimum volume solution for the same problem, but with the additional constraint that only 3 joints are permitted. This permits only a single topology; two members reaching from the point of application of the forces to the support line. Therefore, all potential solutions to this problem can be enumerated using 2 degrees of freedom, the vertical locations of the two support points, which will be denoted y_A and y_B (Fig. 17i-j).

The two component load-cases are identical to the previous section. The lengths of the members are given by $l_A = \sqrt{y_A^2 + d}$ and $l_B = \sqrt{y_B^2 + d}$ respectively.

The member forces are found from equilibrium equations at the loaded point. In $\mathbf{P}_1^*$ the member forces in members A and B are

$$q_1^A = Q\frac{(\sin(\frac{\pi}{4}+\theta)y_B - \cos(\frac{\pi}{4}+\theta)d)\sqrt{y_A^2+1}}{d(y_A - y_B)} \tag{20}$$

$$q_1^B = Q\frac{(-\sin(\frac{\pi}{4}+\theta)y_A + \cos(\frac{\pi}{4}+\theta)d)\sqrt{y_B^2+1}}{d(y_A - y_B)} \tag{21}$$

In $\mathbf{P}_2^*$, the member forces are

$$q_2^A = -Q\frac{(-\cos(\frac{\pi}{4}+\theta)y_B - \sin(\frac{\pi}{4}+\theta)d)\sqrt{y_A^2+1}}{d(y_A - y_B)} \tag{22}$$

$$q_2^B = -Q\frac{(\cos(\frac{\pi}{4}+\theta)y_A + \sin(\frac{\pi}{4}+\theta)d)\sqrt{y_B^2+1}}{d(y_A - y_B)} \tag{23}$$

The superposition principle is again used to find the overall design (Fig. 17k, l), and define the total volume as:

$$V = \frac{V_1^* + V_2^*}{\sqrt{2}} = \frac{(|q_1^A l_A| + |q_1^B l_B|) + (|q_2^A l_A| + |q_2^B l_B|)}{\sigma\sqrt{2}} \tag{24}$$

By inspection of the graph for all values of $0 \leq \theta \leq \frac{\pi}{2}$ (see examples in Fig. 18), it can be seen that the minimum volume solution falls within or on the border of the region where the member forces in $\mathbf{P}_1^*$ are in the same direction, and the member forces in $\mathbf{P}_2^*$ are in opposing directions. Based on this, the expression for V is re-written without using the absolute value operator, for the purposes of this derivation using the sign convention that tensile stresses are negative.

$$V = \frac{-q_1^A l_A - q_1^B l_B - q_2^A l_A + q_2^B l_B}{\sqrt{2}} \tag{25}$$

Additionally, the cusps of the plots in Fig. 18 must be considered; these define the limits of the region within which (25) is valid. They are given by:

$$q_1^A \leq 0 \qquad q_2^A \leq 0 \tag{26}$$

$$q_1^B \leq 0 \qquad q_2^B \geq 0 \tag{27}$$

For a given value of θ, these equations each define a vertical plane. Equations (26) are planes with constant y_B, and (27) are planes with constant y_A.

C.2.2 Optimal values in each region

The equations which describe the optimal solution vary depending upon the value of θ. Three regions are possible, and each of these are considered separately.

When $\theta > \theta_2$ (where θ_2 is a critical value, approximately equal to 0.9), the minimal value is found on the cusp of the volume function defined by $q_1^A = 0$. The optimal point lies on the minima of the cusp line, i.e. where the partial derivative $V_{y_A} = 0$. In this region, the optimal values of y_A and y_B, and the optimal volume V are given by:

$$y_A = \frac{1}{\tan(\frac{\pi}{4} + \theta)} + \frac{\sqrt{2}}{\sin(\frac{\pi}{4} + \theta)} \tag{28a}$$

$$y_B = \frac{-1}{\tan(\frac{\pi}{4} + \theta)} \tag{28b}$$

$$V = \frac{\sqrt{2}\left(\sin(\theta + \frac{\pi}{4}) + 2\sqrt{2} + 3\cos(\theta + \frac{\pi}{4})\right)}{2\sin^2(\theta + \frac{\pi}{4})} \tag{28c}$$

Similarly, for values of $\theta \leq \theta_1$ (where $\theta_1 \approx 0.65$), the minimal value is found on the intersection of $V_{y_B} = 0$ and $q_1^B = 0$. This gives

$$y_A = \frac{1}{\tan(\frac{\pi}{4} + \theta)} \tag{29a}$$

$$y_B = \frac{-1}{\tan(\frac{\pi}{4} + \theta)} + \frac{\sqrt{2}}{\sin(\frac{\pi}{4} + \theta)} \tag{29b}$$

$$V = \frac{\sqrt{2}\left(\sin(\theta + \frac{\pi}{4}) + 2\sqrt{2} - 3\cos(\theta + \frac{\pi}{4})\right)}{2\sin^2(\theta + \frac{\pi}{4})} \tag{29c}$$

In the inner region, where $\theta_1 \leq \theta \leq \theta_2$, the minimum volume structure is found at the local minima of (25), i.e. where $V_{y_A} = 0$ and $V_{y_B} = 0$. In this region, values for pairs of y_A and y_B are given by the equation:

$$\begin{aligned} 0 = \; & & & & y_B^4 \\ & & -2y_A^2 y_B^3 & +2y_A y_B^3 & \\ & -2y_A^3 y_B^2 & & +4y_A y_B^2 & \\ & +2y_A^3 y_B & +4y_A^2 y_B & +4y_A y_B & -2y_B \\ & +y_A^4 & & -2y_A & -2 \end{aligned} \tag{30}$$

To calculate the corresponding value of θ for such a pair, the values of y_A and y_B are simply substituted into either $V_{y_A} = 0$ or $V_{y_B} = 0$. For this region, it is quite difficult to

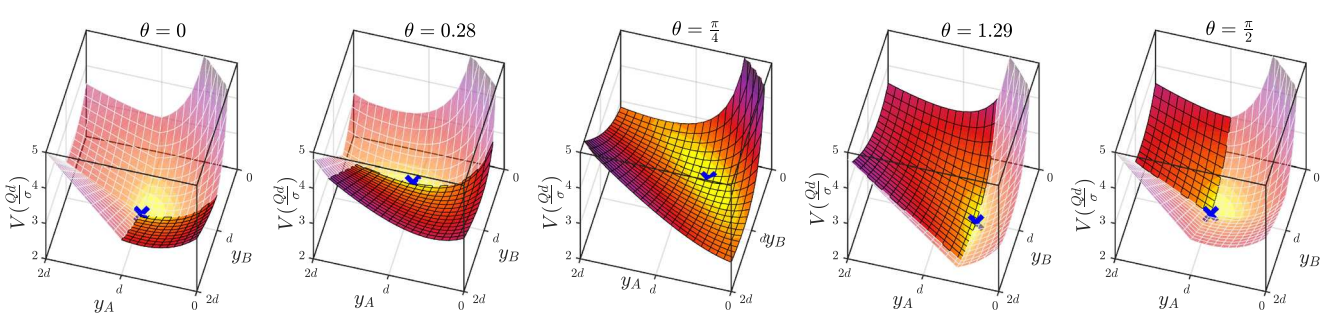
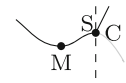
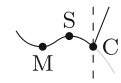
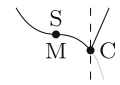


Fig. 18 Simple cantilever: volume of two member truss with for $0 \leq y_A \leq 2d$ and $0 \leq y_B \leq 2d$ for various force inclinations, θ. (The solid region shows where $q_1^A < 0$, $q_1^B < 0$, $q_2^A < 0$ and $q_2^B > 0$, the blue cross shows the globally minimum volume/design to resist each set of forces)

Table 7 Simple cantilever: behaviour of volume function in vicinity of optimal values for force inclinations in the vicinity of θ_2. (Point M is a local minimum, point S is a saddle point, and point C is the minimum value along the cusp at line $q_1^A = 0$. Values in boldface are the optimal points for that problem, note that when $\theta = 0.90068$ two equally optimal solutions exist)

θ	Point	y_A	y_B	V	
0.90011	C	1.30835	0.11521	2.49230	
	S				
	M	1.19519	0.19318	**2.49217**	
0.90068	C	1.30786	0.11580	**2.49136**	
	S	1.27886	0.13492	2.49138	
	M	1.22241	0.17384	**2.49136**	
0.90087	C	1.037709	0.11599	**2.49106**	
	S	1.25026	0.15444	2.49108	
	M				

begin with a value of θ and calculate the optimal values of y_A and y_B.

C.2.3 Boundaries between regions

The final task is to establish the boundary values, θ_1 and θ_2 between the outer and inner regions. To do this, some points of interest must first be defined. The point C is defined as the minimum volume point lying on the cusp $q_1^A = 0$, this is the point given by the (28a–c), and is the optimal value when $\theta \geq \theta_2$.

Next, the stationary points of (25) are considered, these lie on the line defined by (30). There are at most 2 stationary points in the region in which this function is valid. To characterize these, the discriminant of this function, $\Delta = V_{y_A y_A} V_{y_B y_B} - (V_{y_A y_B})^2$ is calculated. The stationary point at which $\Delta > 0$ is defined as point M, this is a local minima, and additionally represents the optimal value in the central region ($\theta_1 \leq \theta \leq \theta_2$). The stationary point where $\Delta < 0$ is defined as S, this is a saddle point of the function. The critical value θ_2 is the point at which the optimal value switches from point C to point M.

Table 7 gives values of y_A, y_B and V for these three points at notable values of θ. Additionally it provides illustrations of the topography of the volume function in the vicinity of C, M and P; these illustrations show the approximate profile the volume function at the bottom of a 'valley' which runs roughly parallel to the plane $y_A + y_B = 1$. This valley may also be observed in the plots in Fig. 18, particularly when $\theta = \frac{\pi}{4}$

When $\theta < 0.900110$, the point S is outside of the valid region for (25). Point S enters the valid region when the relations from (28a, b) are substituted into (30), giving $\theta = 0.900110$. Here, points S and C are co-incident and have a volume which is 0.005% greater than the optimal value at M.

At the point where $V_{y_A} = 0$, $V_{y_B} = 0$ and $\Delta = 0$, a single degenerate stationary point is formed as points S and M become co-incident. This occurs when $\theta = 0.900874$, and the volume at points S and M is 0.0008% higher than the volume at C. Therefore the value θ_2, at which points M and C are equally optimal must lie in the region $0.900110 \leq \theta_2 \leq 0.900874$.

The value of θ_2 is found at the point where $V_{y_A} = 0$, $V_{y_B} = 0$ and the right-hand side of (28c) is equal to the right-hand side of (25) (where the values of y_A and y_B refer to the point M). From this, it is found that $\theta_2 = 0.9006836427$. By a similar logic, $\theta_1 = 0.6701126839$.

It has been shown that within the the region $0.900110 \leq \theta \leq 0.900874$, the range of volumes is small ($< 0.005\%$) over a wide range of possible values for y_A and y_B (of up to $0.1d$). This may cause problems for numerical methods if accuracy levels are not set high enough. Additionally, when $\theta = \theta_1$ or θ_2, two distinct solutions are equally optimal.

References

Aage N, Andreassen E, Lazarov BS, Sigmund O (2017) Giga-voxel computational morphogenesis for structural design. Nature 550(7674):84

Achtziger W (2007) On simultaneous optimization of truss geometry and topology. Struct Multidiscip Optim 33(4-5):285–304

Achtziger W, Stolpe M (2007) Truss topology optimization with discrete design variables - guaranteed global optimality and benchmark examples. Struct Multidiscip Optim 34(1):1–20

Ahrari A, Atai AA, Deb K (2015) Simultaneous topology, shape and size optimization of truss structures by fully stressed design based on evolution strategy. Eng Optim 47(8):1063–1084

Asadpoure A, Guest JK, Valdevit L (2015) Incorporating fabrication cost into topology optimization of discrete structures and lattices. Struct Multidiscip Optim 51(2):385–396

Bendsøe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. Arch Appl Mech 69(9–10):635–654

Chan ASL (1960) The design of Michell optimum structures. Tech. rep., College of Aeronautics Cranfield
Dantzig GB, Wolfe P (1960) Decomposition principle for linear programs. Oper Res 8(1):101–111
Dantzig G, Fulkerson R, Johnson S (1954) Solution of a large-scale traveling-salesman problem. J Oper Res Soc Am 2(4):393–410
Dorn W, Gomory RE, Greenberg HJ (1964) Automatic design of optimal structures. J de Mecanique 3:25–52
Gilbert M, Tyas A (2003) Layout optimization of large-scale pin-jointed frames. Eng Comput 20(8):1044–1064
Gonçalves MS, Lopez RH, Miguel LFF (2015) Search group algorithm: a new metaheuristic method for the optimization of truss structures. Comput Struct 153:165–184
Groenwold A, Stander N (1997) Optimal discrete sizing of truss structures subject to buckling constraints. Struct Optim 14(2-3):71–80
Guo X, Zhang W, Zhong W (2014) Doing topology optimization explicitly and geometrically—a new moving morphable components based framework. J Appl Mech 81(8):081009
Gurobi Optimization LLC (2018) Gurobi optimizer reference manual. http://www.gurobi.com
Haunert JH, Wolff A (2010) Optimal and topologically safe simplification of building footprints. In: Proceedings of the 18th sigspatial international conference on advances in geographic information systems. ACM, pp 192–201
He L, Gilbert M (2015) Rationalization of trusses generated via layout optimization. Struct Multidiscip Optim 52(4):677–694
Hoang VN, Jang GW (2017) Topology optimization using moving morphable bars for versatile thickness control. Comput Methods Appl Mech Eng 317:153–173
Huang XH, Xie Y (2007) Bidirectional evolutionary topology optimization for structures with geometrical and material nonlinearities. AIAA J 45(1):308–313
IBM Corp (2015) CPLEX User's manual. https://www.ibm.com/analytics/cplex-optimizer
Kanno Y (2013) Topology optimization of tensegrity structures under compliance constraint: a mixed integer linear programming approach. Optim Eng 14(1):61–96
Kanno Y, Fujita S (2018) Alternating direction method of multipliers for truss topology optimization with limited number of nodes: a cardinality-constrained second-order cone programming approach. Optim Eng 19(2):327–358
Kanno Y, Guo X (2010) A mixed integer programming for robust truss topology optimization with stress constraints. Int J Numer Methods Eng 83(13):1675–1699
Kelley Jr JE (1960) The cutting-plane method for solving convex programs. J Soc Ind Appl Math 8(4):703–712
Koumousis VK, Georgiou PG (1994) Genetic algorithms in discrete optimization of steel truss roofs. J Comput Civ Eng 8(3):309–325
Leng G, Duan B (2012) Topology optimization of planar truss structures with continuous element intersection and node stability constraints. Proc Institut Mech Eng Part C: J Mech Eng Sci 226(7):1821–1831
Mazurek A, Baker WF, Tort C (2011) Geometrical aspects of optimum truss like structures. Struct Multidiscip Optim 43(2):231–242
Mela K (2014) Resolving issues with member buckling in truss topology optimization using a mixed variable approach. Struct Multidiscip Optim 50(6):1037–1049
Michell AGM (1904) The limits of economy of material in frame-structures. Philos Mag 8(47):589–597
Mortazavi A, Toğan V (2016) Simultaneous size, shape, and topology optimization of truss structures using integrated particle swarm optimizer. Struct Multidiscip Optim 54(4):715–736
Nagtegaal J, Prager W (1973) Optimal layout of a truss for alternative loads. Int J Mech Sci 15(7):583–592
Ohsaki M (2016) Optimization of finite dimensional structures. CRC Press
Ohsaki M, Katoh N (2005) Topology optimization of trusses with stress and local constraints on nodal stability and member intersection. Struct Multidiscip Optim 29(3):190–197
Park P (2013) Application of design synthesis technology in architectural practice. PhD thesis, University of Sheffield
Park J, Sutradhar A, Shah JJ, Paulino GH (2018) Design of complex bone internal structure using topology optimization with perimeter control. Comput Biol Med 94:74–84
Parkes E (1975) Joints in optimum frameworks. Int J Solids Struct 11(9):1017–1022
Prager W (1977) Optimal layout of cantilever trusses. J Optim Theory Appl 23(1):111–117
Prager W (1978) Optimal layout of trusses with finite numbers of joints. J Mech Phys Solids 26(4):241–250
Prager W, Shield R (1967) A general theory of optimal plastic design. J Appl Mech 34(1):184–186
Rozvany G, Hill R (1978) Optimal plastic design: superposition principles and bounds on the minimum cost. Comput Methods Appl Mech Eng 13(2):151–173
Rozvany GIN, Bendsoe MP, Kirsch U (1995) Layout optimization of structures. Appl Mech Rev 48(2):41–119
Rozvany GI, Sokół T, Pomezanski V (2014) Fundamentals of exact multi-load topology optimization–stress-based least-volume trusses (generalized Michell structures)-Part I: plastic design. Struct Multidiscip Optim 50(6):1051–1078
Sokół T, Rozvany G (2013) On the adaptive ground structure approach for multi-load truss topology optimization. In: 10th World Congress on Structural and Multidisciplinary Optimization, 19th-24th May, Orlando
Stolpe M (2010) On some fundamental properties of structural topology optimization problems. Struct Multidiscip Optim 41(5):661–670
Stolpe M (2016) Truss optimization with discrete design variables: a critical review. Struct Multidiscip Optim 53(2):349–374
Torii AJ, Lopez RH, Miguel LF (2016) Design complexity control in truss optimization. Struct Multidiscip Optim 54(2):289–299
Van Mellaert R, Mela K, Tiainen T, Heinisuo M, Lombaert G, Schevenels M (2018) Mixed-integer linear programming approach for global discrete sizing optimization of frame structures. Struct Multidiscip Optim 57(2):579–593
Villar JR, Vidal P, Fernández MS, Guaita M (2016) Genetic algorithm optimisation of heavy timber trusses with dowel joints according to Eurocode 5. Biosyst Eng 144:115–132
Zhang W, Zhou J, Zhu Y, Guo X (2017) Structural complexity control in topology optimization via moving morphable component (mmc) approach. Struct Multidiscip Optim 56(3):535–552
Zhou M, Lazarov BS, Wang F, Sigmund O (2015) Minimum length scale in topology optimization by geometric constraints. Comput Methods Appl Mech Eng 293:266–282

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.