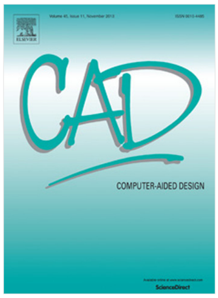
Journal Pre-proof



Computational Design of Lightweight Trusses

Caigui Jiang, Chengcheng Tang, Hans-peter Seidel, Renjie Chen,
Peter Wonka

PII: S0010-4485(21)00087-7
DOI: https://doi.org/10.1016/j.cad.2021.103076
Reference: JCAD 103076

To appear in: *Computer-Aided Design*

Received date : 20 May 2021
Accepted date : 22 June 2021

Please cite this article as: C. Jiang, C. Tang, H.-p. Seidel et al., Computational Design of Lightweight Trusses. *Computer-Aided Design* (2021), doi: https://doi.org/10.1016/j.cad.2021.103076.

Computational Design of Lightweight Trusses

Caigui Jiang[a,b,c], Chengcheng Tang[d], Hans-peter Seidel[c], Renjie Chen[e,*], Peter Wonka[b]

[a]*College of Artificial Intelligence, Xi'an Jiaotong University, Xi'an, China*
[b]*King Abdullah University of Science and Technology, Thuwal, KSA*
[c]*The Max Planck Institute for Informatics, Saarbrücken, Germany*
[d]*Facebook Reality Labs, San Francisco, USA*
[e]*University of Science and Technology of China, Hehui, China*

Abstract

Trusses are load-carrying light-weight structures consisting of bars connected at joints ubiquitously applied in a variety of engineering scenarios. Designing optimal trusses that satisfy functional specifications with a minimal amount of material has interested both theoreticians and practitioners for more than a century. In this paper, we introduce two main ideas to improve upon the state of the art. First, we formulate an alternating linear programming problem for geometry optimization. Second, we introduce two sets of complementary topological operations, including a novel subdivision scheme for global topology refinement inspired by Michell's famed theoretical study. Based on these two ideas, we build an efficient computational framework for the design of lightweight trusses. We show that our method achieves trusses with smaller volumes and is faster compared with recent state-of-the-art approaches.

Keywords: truss, topology optimization, geometry.

1. Introduction

Trusses are crucial and fundamental structures in multiple modern engineering domains. They consist of bar elements that are connected by pin joints. Because of their efficiency and lightweight nature, trusses see considerable amount of usage in industrial design and architectural construction, e.g., for support structures of buildings, bridges, transmission towers, or even domes in playgrounds.

Designing a lightweight truss typically starts with a functional specification, e.g., in the form of external forces that the structure has to withstand. The design problem can then be formulated as an optimization problem to determine the geometry, topology, and the cross-sections of the truss. In other words, we have to find answers to the following questions: Where to put the intermediate joints? How to connect the joints with bars? What are the cross-section areas of the bars? These tasks are notoriously challenging because the optimization of geometry, topology, and cross-sections is interrelated, and there exists an infinite number of possible topologies which are difficult to classify and quantify. Even for a simple case, the optimal topology is not intuitive. As shown in Figure 2, to support two pairs of opposing forces lying on two straight lines, the simplest truss on the left with two bars, one in tension (blue) and another in compression (red), may be intuitively considered as the lightest truss. However, a lighter design with more intermediate joints and connections can be found as shown in the Figure 2 right. Another simple functional specification problem is called the three forces problem (3FP) [1, 2]. 3FP is formulated as follows: find the lightest fully stressed truss transmitting three self-equilibrated co-planar forces. Although there are only three forces, the problem is still unsolved analytically for general cases.

In this paper, we mainly follow previous work and take functional specifications in the form of supporting points and applied forces as input. Our goal is then to construct a lightweight truss with optimal joint positions, topology, and cross-sections. As an example shown in Figure 3 left, two supporting points and one external force are given as inputs. Our computational method generates the topology automatically and optimizes the nodal positions and cross-section areas as shown in Figure 3 right.

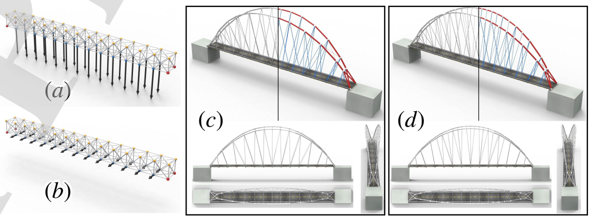


Figure 1: A bridge design problem from [3]. The input is an initial structure of 258-bars with supporting points (red) and two sets of forces: (a) downward loads of magnitude 1 and (b) horizontal loads of magnitude 0.2 perpendicular to the bridge's main direction. The result in [3] has a total volume of 408.8 and took over 1000s to compute. (c) Our optimal geometry and topology viewed from different angles. The total volume is 333.4 and the running time is less than 10s. (d) Further structure refinement based on (c) to achieve a total volume of 331.9.

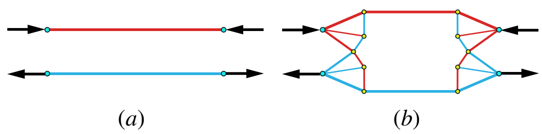


Figure 2: Two truss designs for the same functional specification. Left: a straightforward design with two bars. Right: a more complex and less intuitive design with less material usage.

There are two major strategies to tackle this problem in previous work. The first strategy is to start with a densely-connected

*Corresponding author
Email address: renjiec@ustc.edu.cn (Renjie Chen)

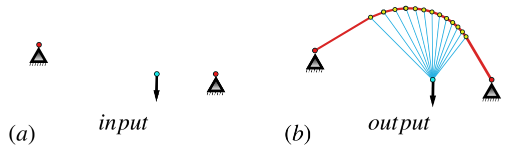


Figure 3: (a): a functional specification including two supporting points and one external force. (b): an optimal truss.

structure and to subsequently identify which bars to remove (e.g., the ground structure method [4] and its variations [5, 6]). The main limitation of this strategy is its sub-optimality due to its heavy dependence on the initialization as it is extremely unlikely that the optimal structure is a sub-structure of the initial one. The other strategy is to start with a sparse structure and to iteratively add new joints and bars. One of the most famous methods in [7] adds one joint at a time and can only deal with one single load 2D problem.

Even though topology optimization is such a longstanding and fundamental problem in structural engineering, one can identify large possible improvements to the current state of the art. First, the search space for truss topology is not properly explored by previous algorithms. We could observe that it is very difficult to optimize the topology in a single stage. Much better results can be achieved by proceeding in two stages of topology optimization: computing a coarse truss and truss subdivision. Our novel subdivision approach is inspired by Michell's pioneering theoretical treatment of optimal truss design in [8]. Second, the geometry optimization used in previous work is not efficient. To tackle this problem we decompose geometry optimization into alternating linear programming formulations to reduce the running time.

Our main contributions are as described in the following

- We propose two categories of complementary topology operations, local and global. While local operations have been used in previous work, our global operations based on subdivision are our original contribution.
- We introduce a novel algorithm for geometry optimization based on alternating linear programming (ALP) that jointly optimizes joint positions and bar cross sections.
- Based on these two technical contributions, we build a framework for lightweight truss design, a longstanding and important problem in structural engineering, architecture, graphics, and design. Compared with recent state-of-the-art approaches, our method creates trusses with smaller volumes, can handle more complex functional specifications, and is over two orders of magnitude faster.

2. Previous Work

In recent years, combining geometric modeling together with realistic engineering considerations, especially static equilibrium and manufacturability, have attracted the interests of many researchers in the graphics community. Beyond applications in the virtual world [9], those previous works enable novel and functional designs manufacturable with 3D printing [10, 11], laser cutting [12], masonry structure [13, 14, 15, 16, 17, 18], for toys [19, 20], furniture [21, 22], and architecture [23, 24]. The most relevant works to ours are [25] and [26]. Jiang et al. propose a framework to design and optimize space structures where only a small set of cross-section areas are allowed. Therefore, Jiang et al. [25] compute a specialized form of truss, but we focus on the classical problem of truss design without discrete restrictions on the cross sections. The main practical difference is that our proposed method can generate a truss from scratch, whereas Jiang et al. relies on a reasonable truss being given as input. In our results we also demonstrate that our proposed optimization technique ALP produces better results on our problem formulation than the geometric optimization technique used in [25]. Kilian et al. [26] provide an interesting geometric understanding of "optimality" of surface-like lightweight structures. Compared with their work, we tackle a problem for common and general trusses in both 2D and 3D, instead of focusing on load-carrying surfaces.

The problem of designing a truss with a minimal volume of material that supports imposed external forces was first studied in [8]. In the milestone paper, Michell proved that an optimal truss must follow orthogonal networks of lines of maximal and minimal strains in a constant-magnitude strain field. An optimal truss is usually called a Michell truss. Following his work, research on the topic of optimal truss can be divided into two categories: exact-analytical formulations and approximate-discretized formulations.

An exact-analytical formulation assumes that the truss is a continuum structure connected by an infinite number of bars with infinitesimally small cross-sections. In analytical formulations, the theoretical optimal design is determined exactly through the simultaneous solution of a system of equations expressing the conditions for optimality. The basic principles were establish in [27] and [8] and a more general treatment was outlined in [28] and [29]. Recent works on deriving exact solutions were presented in [30], [31], [32], and [33] for a series of benchmark problems. Basically, the analytical solutions are very hard to obtain and only available for some special boundary conditions. While they are less practical in most of the generic scenarios, these solutions could be used as references to verify the performance of numerical methods.

Discretized numerical formulations are more practical and efficient approaches for structural design tasks presented in the real world. The most influential method is the ground structure method (GSM) which was first proposed in [4]. This method consists of generating a fixed grid of joints and adding bars in some or all of the possible connections between the joints as potential structural or vanishing bars. The optimized structure for the imposed functional specification is found using the cross-section areas as design variables, and the whole problem is formulated as a linear programming problem. Its optimal topology is achieved by eliminating the zero-area cross sections. The ground structure method has been recently improved in [5] and [6].

Besides GSM, some other numerical methods are proposed recently, such as the method in [7], carries out geometry optimization in conjunction with a heuristic 'joint adding' algorithm, generating an increasingly complex truss structure from a relatively simple initial layout. However, this algorithm can only add one joint per time and only works for single load cases. An efficient algorithm proposed by He and Gibbert [34] combines layout optimization with geometry optimization. Similar to GSM, its layout optimization starts from a densely connected truss and is formulated by a linear programming problem, and its geometry optimization is formulated by a non-linear optimization as a post-processing step.

3. Overview

Our framework has the following major components:

- **Functional specification** (C1). The input to our framework is the functional specifications including the external forces and supporting points together with a set of structural constraints, e.g., design regions, geometric obstacles, and material properties. (See Section 4.1)
- **Initialization** (C2). To obtain an initial truss, we create a grid of intermediate joints and densely connect them. This grid is located inside the design region and its size is proportional to the bounding box of the points in the input specification. (See Section 4.2)
- **Local topology operations** (C3). We locally manipulate the topology through some geometry operations such as removing bars with vanishing cross-section areas and joints without any connection, merging close joints, etc. (See Section 4.3)
- **Global topology refinement** using subdivision (C4). Use an optimized coarse truss as input, we further refine the truss through subdivision. (See Section 4.4)
- **Geometric optimization** using ALP (C5). Given a fixed topology, we propose an alternating linear programming algorithm (ALP) to reduce the total volume of the truss by adjusting the joint positions and cross-section areas of bars. This algorithm is an essential component and its details are introduced in Section 5.

4. Design Framework

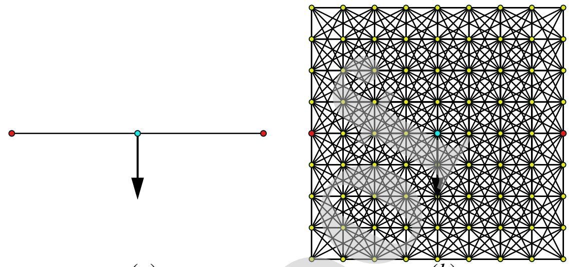
We provide a framework for the computational design of lightweight trusses. In this section, we describe the input specification, the initialization, local topology operations, and global topology refinement.

4.1. Functional Specification

The input to our framework is the functional specification including the external forces and supporting points together with a set of structural constraints, e.g., design regions, geometric obstacles, and material properties. Throughout the paper, we visualize supporting joints as red dots, joints with active forces as blue dots, and intermediate joints as yellow dots. We also visualize bars in tension in blue and bars in compression in red. In addition, the thickness of the bars is visualized in proportional to the computed cross-section areas. Note that when the external forces are in self-equilibrium, the input specification may have no supporting points, for example, the three forces problem (3FP) in 2D.

4.2. Truss Initialization

We build on previous work to compute an initial truss. There are two approaches to tackle this problem. One simply adds connections between provided joints in the functional specification (supporting joints and joints with active forces). For example, as shown in Figure 5 left, two bars connecting the joints with active forces (blue) and the supporting joints (red) are set as an initial truss. In some cases, this initialization is too simple to construct an equilibrium force system. Another method adds a grid of intermediate points over the design region and densely connects them as shown in Figure 5 right. The increasing method such as the work in [7] used the first initialization. The GSM usually uses the latter one with a large number of intermediate joints.



(*a*) (*b*)

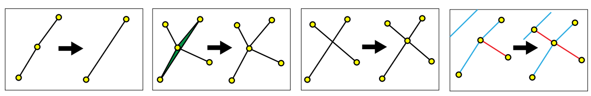
Figure 5: Two kinds of initializations. (a): connect force application points and supporting points. (b): a densely connected initial structure for GSM.

In our framework, we first add some intermediate joints and connections. For instance, a size of $n \times n$ 2D grid points or $n \times n \times n$ 3D grid points and their dense connections, where n is a user specified parameter. The default value of n is the number of joints specified in the functional specification. This is quite similar to the GSM, the difference is that the number of new joints that we add is usually much less.

4.3. Local Topology Operations

We use the following local topology operations:

- Removing the bars with cross-section areas less than a small threshold ϵ_1.
- Removing joints without any attached bars.
- Merging joints that are closer to each other than a small threshold ϵ_2.
- Removing intermediate joints with valence two, as shown in Figure 6(a).
- Deleting the longest bar of a long narrow triangle as shown in Figure 6(b).
- Adding a new joint for each pair of intersecting bars. Split this pair of bars into four new bars and connect them at the new joint as shown in Figure 6(c).
- Fixing non-boundary T-junctions by adding a bar and a new joint connecting the new bar and the original truss as shown in Figure 6(d). The new joint is created at the point closest to the extension of the existing bar creating the T-junction.



(*a*) (*b*) (*c*) (*d*)

Figure 6: Four types of local topology operations: (a) Delete a joint of valence two. (b) Remove an ill-shaped narrow triangle. (c) Add an additional joint for a pair of intersecting bars.(d) Fix non-boundary T-junction.

These operations change the local topology and update joint positions.

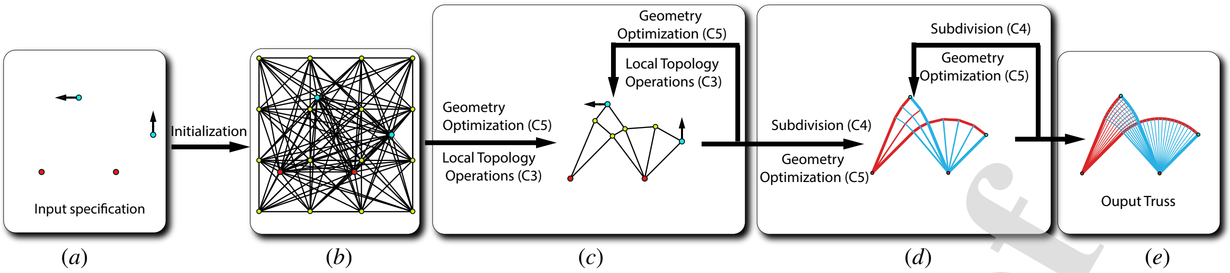


Figure 4: Framework overview: Based on an input functional specification (a), our system creates an initial truss (b). Then, we proceed in two phases, coarse truss optimization (c) and structure refinement through subdivision (d). In the first phase, we interleave geometric optimization using ALP with local topology operations. In phase 2, we interleave subdivision with geometry operation using ALP. The output truss is shown in (e).

4.4. Global Topology Operation — Subdivision

The main idea of global topology refinement is to add joints and bars to the truss to be able to reduce its volume after geometry optimization. While previous work, e.g., [7], also proposes to add joints and bars, they add only one joint at a time by testing a large number of candidate locations. This results in a very expensive algorithm. By contrast, we propose to add new joints and bars based in a systematic manner. Our algorithm is inspired by two observations. First, Michell's theory [8] concludes that the minimum-weight truss should follow two families of continues curves which are orthogonal to each other, one in tension and one in compression. Second, an interesting aspect of truss design is that trusses with more bars can often be lighter than trusses with fewer bars, as more degrees of freedom are provided to approximate an analytical limit. Our algorithm refines the discrete equivalent of such families of curves by subdivision in an efficient and coordinated manner. Most importantly, we insert multiple bars in one step.

We first calculate a pair of tension-compression directions at each joint. As shown in Figure 7(a), for each joint, we separately average the bars connected with this joint according to their force signs (+ for compression (red) and - for tension (blue)) with their force magnitudes as weights. Figure 7(b) shows the calculated nearly-orthogonal directions on a truss. Then we calculate the new joints for the bars which are estimated to be split. Take the bar in Figure 7(c) for example, we know the coordinates of its two ends, p_i and p_j, and the tension-compression directions at its two ends. For a bar in tension, we calculate the two directions, v_i and v_j, which are orthogonal to the compression directions (red) at its two ends. Using (p_i, p_j, v_i, v_j), we calculate a Bézier curve and set the mid-point of the curve as the new joint. For a bar in compression, we follow a similar procedure.

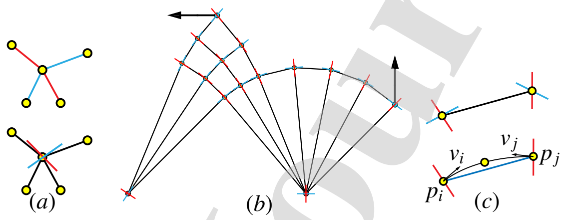


Figure 7: Subdivision of a truss: (a) construction of compression-tension directions at each joint; (b) a compression-tension field for a truss; (c) the strategy to calculate an edge mid-point.

The purpose of truss subdivision is to improve the orthogonality of the bars in tension and in compression. Given an initial coarse truss, we know its geometry, topology, and the axial force of each bar. Consider the truss as a graph, we extract triangles and quadrilaterals. The triangles are usually formed by bars connected to joints specified in the functional specification. For a triangle as shown in Figure 8, lower row, we add a new joint for the bar whose force sign is different from the other two and connect the new joint with the opposite joint. A quadrilateral is subdivided if it has two non-adjacent bars in compression and the other two non-adjacent bars in tension. As shown in Figure 8 upper row, we add four new joints for its four bars and one more joint at its face center initialized as the average of the previous four, and connect the face-center joint with each edge-middle joint. In the subdivided truss, we remove each bar where a new joint is added, and connect its two ends with the new joint as shown in Figure 8. Figure 9 show the results of different levels of subdivisions using the functional specification in Figure 4(a).

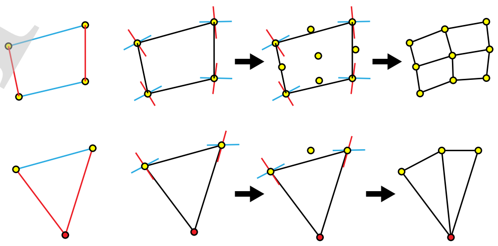


Figure 8: Truss subdivision strategies for quadrilaterals (top) and triangles (bottom).

Although the above illustrations are for 2D cases, we can use the same subdivision strategies for 3D trusses. We extract all triangles and quads and test if they should be subdivided. In 3D, we use the same conditions as in 2D (see Fig. 8).

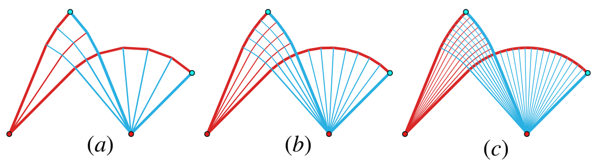


Figure 9: Optimal truss designs in different subdivision levels. The input functional specification is given in Figure 4.

5. Alternating LP

In this section, we introduce the alternating linear programming (ALP) step which optimizes joint positions and cross-section areas of bars for a given topology. ALP serves as the backbone of the proposed approach. Both the local and global topology operations in the previous section are based on optimization of joint positions and cross-section areas. Directly optimizing joint positions and axial forces is a highly nonlinear problem. Therefore, we split the problem into two linear problems. In Algorithm a, we solve force densities alone without changing the joint positions by the ground structure method. In Algorithm b, we update joint positions and force densities jointly based on results from Algorithm a.

5.1. Algorithm a: The Ground Structure Method

Let us first recall the basic plastic formulation of the ground structure method [35], which solves a continuous linear programming problem to minimize the total volume of material under the premise of force balance with feasible axial forces:

$$
\begin{aligned}
\underset{a_i, s_i}{\text{minimize}} \quad & \sum_{i=1}^{|E|} l_i a_i, && (1)\\
\text{subject to} \quad & \mathbf{B}^T \mathbf{s} = -\mathbf{f}, && (1a)\\
& a_i + s_i \geq 0, \qquad i = 1, \ldots, |E| && (1b)\\
& a_i - s_i \geq 0, \qquad i = 1, \ldots, |E| && (1c)
\end{aligned}
$$

where each scalar a_i is the cross-section area of the i-th bar. $\mathbf{B}^T$ is the nodal equilibrium matrix, built from the directional cosines of the bars. More details about this matrix are given in the additional materials. $\mathbf{s}$ is a vector with the internal (axial) force for all bars, and $|E|$ is the number of bars. $\mathbf{f}$ is a vector of the external force for all joints. The internal force s_i should be within the range of admissible axial forces $[-\sigma_T a_i, \sigma_C a_i]$. Here, we assume that the maximal compressive and tensile strains are the same, $\sigma_C = \sigma_T = \sigma$, which is a constant value. We set $\sigma = 1$ in the formulation. The inequality constraints, 1b and 1c are equivalent to $a_i \geq |s_i|$. As the length of each bar, l_i, is positive, $l_i > 0$, the objective function requires the cross-section area of each bar, a_i, to be its smallest permissible value, just enough to support the actual axial force of that bar. Then, we have $a_i = |s_i|$ and the following formulation.

$$
\begin{aligned}
\underset{s_i}{\text{minimize}} \quad & \sum_{i=1}^{|E|} l_i |s_i|, && (2)\\
\text{subject to} \quad & \mathbf{B}^T \mathbf{s} = -\mathbf{f}. && (2a)
\end{aligned}
$$

The above formulation is equivalent when we use force densities $w_i = s_i / l_i$ as variables instead of axis forces s_i. The new formulation is transformed to:

$$
\begin{aligned}
\underset{w_i}{\text{minimize}} \quad & \sum_{i=1}^{|E|} l_i^2 |w_i|, && (3)\\
\text{subject to} \quad & \mathbf{C}^T \mathbf{w} = -\mathbf{f}. && (3a)
\end{aligned}
$$

Here, in Equation 3a, the matrix $\mathbf{C}$ is a simpler expression than $\mathbf{B}$ because its elements are linear combinations of joint positions. More details about the matrix $\mathbf{C}$ are given in the additional materials.

5.2. Algorithm b: Relocation of Joints

In Algorithm a, the joint positions are assumed to be fixed and the axial forces are the only variables. To further reduce the total volume of material, we complement it with Algorithm b and calculate the displacements of joints to leverage more degrees of freedom. We assume the initial values of force densities, $\mathbf{w}$, are known by solving an LP problem in Equation 1 and set the difference of joint positions, $\mathbf{u}$, and the difference of force densities of bars, $\Delta\mathbf{w}$, as variables. By directly rewriting Equation 3, we have

$$
\begin{aligned}
\underset{u_i, \Delta w_i}{\text{minimize}} \quad & \sum_{i=1}^{|E|} \mathbf{sgn}(w_i + \Delta w_i)(l_i + \Delta l_i)^2 (w_i + \Delta w_i), && (4)\\
\text{subject to} \quad & (\mathbf{C} + \Delta\mathbf{C})^T (\mathbf{w} + \Delta\mathbf{w}) = -\mathbf{f}. && (4a)
\end{aligned}
$$

Here, we assume that the change of force densities, $\Delta\mathbf{w}$, is small and that the signs of force densities remain the same, $\mathbf{sgn}(w_i + \Delta w_i) = \mathbf{sgn}(w_i)$. As Algorithm b is applied after Algorithm a, the values, l_i, w_i, $\mathbf{sgn}(w_i)$, and $\mathbf{C}$, are all known.

To simplify the problem which has a cubic objective function and quadratic constraints, our goal is to approximate the above formulation with a linear programming problem and solve it in a sequential manner in conjunction with Algorithm a. By expanding the objective function, we have $(l_i + \Delta l_i)^2 (w_i + \Delta w_i) = (l_i^2 w_i + 2 l_i \Delta l_i w_i + \Delta l_i^2 w_i + l_i^2 \Delta w_i + 2 l_i \Delta l_i \Delta w_i + \Delta l_i^2 \Delta w_i) \approx (l_i^2 w_i + 2 l_i \Delta l_i w_i + l_i^2 \Delta w_i)$. Here, we remove the higher order terms, and use the fact that $l_i^2 w_i$ is constant. The objective function is approximated by $\sum_{i=1}^{|E|} \mathbf{sgn}(w_i)(2 l_i \Delta l_i w_i + l_i^2 \Delta w_i)$. As shown in Figure 10, $\Delta l_i \approx \mathbf{d}_i \cdot (\mathbf{u}_{i2} - \mathbf{u}_{i1})$, where $\mathbf{d}_i$ is the unit direction vector of the i-th bar connecting the joints $i1$ and $i2$. The objective function is linear with $\mathbf{u}_j$ and Δw_i as variables, where $j = 1, \ldots, |V|$ and $i = 1, \ldots, |E|$.

The force equilibrium constraint, $(\mathbf{C} + \Delta\mathbf{C})^T (\mathbf{w} + \Delta\mathbf{w}) = -\mathbf{f}$, is equivalent to $\mathbf{C}^T \Delta\mathbf{w} + \Delta\mathbf{C}^T \mathbf{w} + \Delta\mathbf{C}^T \Delta\mathbf{w} = \mathbf{0}$ because $\mathbf{C}^T \mathbf{w} = -\mathbf{f}$ is ensured by Algorithm a. Here we remove the small higher order term $\Delta\mathbf{C}^T \Delta\mathbf{w}$. Then the force balance constraint is linearized as $\mathbf{C}^T \Delta\mathbf{w} + \Delta\mathbf{C}^T \mathbf{w} = \mathbf{0}$. The matrix $\mathbf{C}$ and the vector $\mathbf{w}$ are known from Algorithm a and the elements in matrix $\Delta\mathbf{C}$ are linear combinations of nodal variations $\mathbf{u}_j = (u_{jx}, u_{jy}, u_{jz})$. Then the constraint is also linear with respect to $\mathbf{u}_j$ and Δw_i.

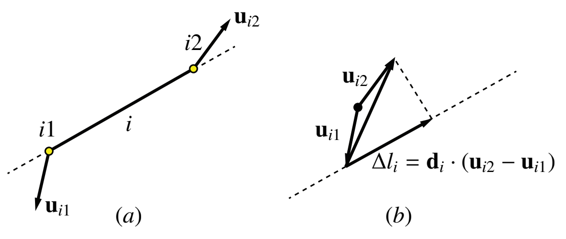


Figure 10: (a) The i-th bar connects the joints $i1$ and $i2$; $\mathbf{u}_{i1}$ and $\mathbf{u}_{i2}$ are joint displacements. (b) The length change along the bar direction, $\Delta l_i \approx \mathbf{d}_i \cdot (\mathbf{u}_{i2} - \mathbf{u}_{i1})$.

Finally, the formulation for Algorithm b is written as

$$
\begin{aligned}
\underset{u_i,\Delta w_i}{\text{minimize}} \quad & \sum_{i=1}^{|E|} \mathbf{sgn}(w_i)(2 l_i w_i \mathbf{d}_i \cdot (\mathbf{u}_{i2} - \mathbf{u}_{i1}) + l_i^2 \Delta w_i), && (5)\\
\text{subject to} \quad & \mathbf{C}^T \Delta \mathbf{w} + \Delta \mathbf{C}^T \mathbf{w} = \mathbf{0}, && (5a)\\
& -\delta_i \le \Delta w_i \le \delta_i;\ i = 1, \ldots, |E|, && (5b)\\
& -\lambda_j \le u_{jx}, u_{jy}, u_{jz} \le \lambda_j;\ j = 1, \ldots, |V|. && (5c)
\end{aligned}
$$

where λ_j and δ_i are the bounds of the variables u_i and Δw_i. In each iteration, we set small values for these bounds, e.g., $\delta_i = 0.1|w_i|$ and $\lambda_i = 0.1\bar{l}$, where $\bar{l}$ is the average length of all bars.

5.3. Alternating Scheme

The above two algorithms are formulated as two LP problems in Equation 1 and Equation 5. The inputs to Algorithm a are the joint positions, $\mathbf{p}$, and the functional specification such as the external forces, **LOAD**, and the supporting points, **SUPP**, and the outputs are the force densities of bars, $\mathbf{w}$. The algorithm a is written as $[\mathbf{w}, V] = \mathbf{ALGa(p, LOAD, SUPP)}$, where V is the total volume of materials. The inputs of Algorithm b are the initial force densities, $\mathbf{w}$, the initial joint positions, $\mathbf{p}$, and the same functional specification. The outputs are the changing values of joint positions and force densities. Then, Algorithm b is written as $[\mathbf{u}, \Delta\mathbf{w}] = \mathbf{ALGb(p, w, LOAD, SUPP)}$. In the whole algorithm, we organize them in an alternating way as shown in Algorithm 1. $N_{\max}$ is the maximum iteration number and $S_{\max}$ is the maximum line search step.

Algorithm 1 Alternating LP for truss geometry optimization

procedure ALTERNATING LP
  Initial joint positions $\mathbf{p}$; **LOAD** and **SUPP**;
  $[\mathbf{w}, V] = \mathbf{ALGa(p, LOAD, SUPP)}$;
  Flag $\leftarrow True$; $N \leftarrow 0$;
  while Flag do
    $[\mathbf{u}, \Delta\mathbf{w}] = \mathbf{ALGb(p, w, LOAD, SUPP)}$;
    procedure LINE SEARCH
      for $j = 0$ to $S_{\max}$ **do**
        $s \leftarrow 2^{-j}$; $\hat{\mathbf{p}} \leftarrow \mathbf{p} + s\mathbf{u}$;
        $[\hat{\mathbf{w}}, \hat{V}] = \mathbf{ALGa}(\hat{\mathbf{p}}, \mathbf{LOAD}, \mathbf{SUPP})$;
        if $\hat{V} < V$ **then**
          $V \leftarrow \hat{V}$; $\mathbf{p} \leftarrow \hat{\mathbf{p}}$; **Break**;
        else
          if $j == S_{\max}$ **then Flag** $= False$;
    endprocedure
    $N \leftarrow N + 1$;
    if $N > N_{\max}$ **then Flag** $= False$;
  endwhile

In Figure 11, we show the effectiveness of ALP for truss optimization with three sets of load specifications involving torques.

5.4. Multiple Load Specifications

For the input specification with multiple sets of external forces, the ALP algorithm is adjusted accordingly. Static equilibrium is required for each set of external loads with generally different internal forces. Thus, assuming trusses are required to withstand K sets of external forces $\mathbf{f}^1, \ldots, \mathbf{f}^K$, the formulation of Algorithm a is rewritten as:

$$
\begin{aligned}
\underset{a_i, s_i^k}{\text{minimize}} \quad & \sum_{i=1}^{|E|} l_i a_i, & \\
\text{subject to} \quad & \mathbf{B}^T \mathbf{s}^1 = -\mathbf{f}^1, & \\
& a_i + s_i^1 \ge 0, & i = 1, \ldots, |E|\\
& a_i - s_i^1 \ge 0, & i = 1, \ldots, |E|\\
& \ldots & \\
& \mathbf{B}^T \mathbf{s}^K = -\mathbf{f}^K, & \\
& a_i + s_i^K \ge 0, & i = 1, \ldots, |E|\\
& a_i - s_i^K \ge 0, & i = 1, \ldots, |E|
\end{aligned}
$$

where $k = 1, \ldots, K$. Force equilibrium constraints similar to Equation 1 are required for each set of external forces. It is also worth noting that this set of equations is sufficient to guarantee force equilibrium in response to linear interpolation of the sets of specified external forces. As each bar needs to support the maximal axial forces from each set of the reaction forces, the cross-section of the i-th bar, $a_i = max\{|{s_i}^1|, \ldots, |{s_i}^K|\}$. We denote $m_i \in \{1, \ldots, k\}$ as the set index for which the i-th bar attains its maximal axial force, $|s_i^{m_i}| = max\{|{s_i}^1|, \ldots, |{s_i}^K|\}$. As those indices, m_i, could be easily found from the result of Algorithm a, corresponding cross sections follow directly, $a_i = |s_i^{m_i}|$. Similarly, by defining $w_i^{m_i} = s_i^{m_i}/l_i$, the formulation of Algorithm b for the multiple-load case can be written as:

$$
\begin{aligned}
\underset{u_i, \Delta {w_i}^k}{\text{minimize}} \quad & \sum_{i=1}^{|E|} \mathbf{sgn}(w_i^{m_i})(2 l_i w_i^{m_i} \mathbf{d}_i \cdot (\mathbf{u}_{i2} - \mathbf{u}_{i1}) + l_i^2 \Delta w_i^{m_i}),\\
\text{subject to} \quad & \mathbf{C}^T \Delta \mathbf{w}^1 + \Delta \mathbf{C}^T \mathbf{w}^1 = \mathbf{0},\\
& \ldots\\
& \mathbf{C}^T \Delta \mathbf{w}^K + \Delta \mathbf{C}^T \mathbf{w}^K = \mathbf{0},\\
& -\delta_i \le \Delta w_i^k \le \delta_i,\\
& -\lambda_j \le u_{jx}, u_{jy}, u_{jz} \le \lambda_j,
\end{aligned}
$$

where $i = 1, \ldots, |E|$, $k = 1, \ldots, K$, and $j = 1, \ldots, |V|$. Here, only the nodal variations, u_i, and the change of force densities, Δw_i^k, are variables, others are known from Algorithm a. Using the same alternating scheme in Section 5.3 by adjusting the Algorithm a and b accordingly, our method can tackle cases of multiple-load input specifications. See Figure 12 for an example.

6. Results

In this section, we illustrate truss designs using our framework for different types of input specifications and compare the results with state-of-the-art methods on selected benchmark design problems.

6.1. Example Designs

We show the results of our method for different types of functional specifications. We present 2D truss designs with a parallel equilibrium force system in Figure 20, a concurrent equilibrium force system in Figure 21, and a non-concurrent equilibrium force system in Figure 22. We illustrate examples of designs for the same input external forces and supporting joints but different

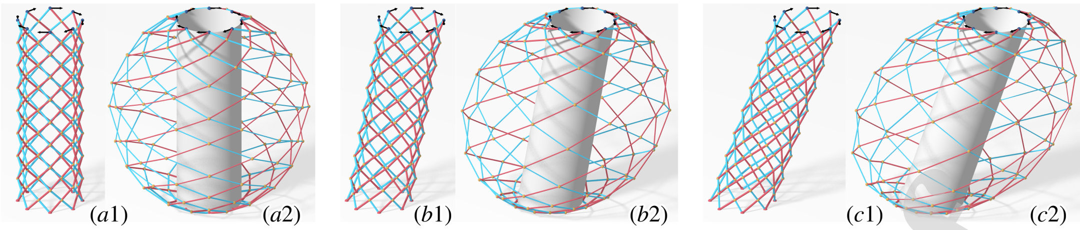


Figure 11: Three trusses optimized through ALP for different input specifications, each with eight supporting joints and eight external loads along circles. From left to right, (a1), (b1), and (c1) are the initial trusses with a total volume of material consumption being 103.93, 105.34 and 109.58 respectively. (a2), (b2), and (c2) are the optimal trusses using ALP with a total volumes of 69.88, 68.53 and 69.21, respectively.

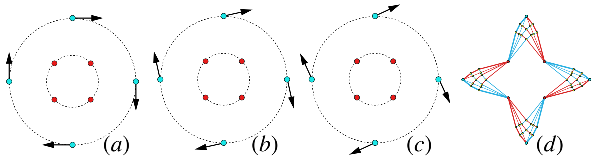


Figure 12: A truss designed to support three sets of different external loads. Here, red joints are fixed and blue joints have external loads applied. With three sets of external loads shown in (a)-(c), our method creates an optimal truss that supports all of them, as shown in (d).

design regions in Figure 19. For 3D trusses, we demonstrate the results for input of a parallel equilibrium force system in Figure 24, a concurrent equilibrium force system in Figure 25, and a non-concurrent equilibrium force system in Figure 26. In addition, we show examples based on real functional requirements such as a 2D bike frame in Figure 23, a 3D cantilever in Figure 27, and a 3D bridge in Figure 1.

6.2. Quantitative Evaluation

Our framework is implemented in Matlab R2016 on a workstation with an Intel Xeon X5550 2.67 GHz processor. We use Mosek [36] as the solver for linear programming. For the ALP algorithm, we set the maximum iteration number $N_{\max}$=500 and the maximum line search step $S_{\max}$=10. For the number of grid points in the initialization, n, we use the default value, the number of joints specified in the functional specification. We set the number of max iteration of Phase 1, $P_{\max 1} = 5$. The number of subdivisions in Phase 2, $P_{\max 2}$, controls the trade off between number of bars and truss weight and is set depending on the context. We set the thresholds ϵ_1 and ϵ_2 in Section 4.3 as $\epsilon_1 = 0.002\bar{a}$, $\epsilon_2 = 0.01\bar{d}$, where $\bar{a}$ is the average cross-section, and $\bar{d}$ is the average value of distances between joints specified in the functional specification. In Table 1, for each optimized truss, we report parameters of trusses such as the number of bars, the total volume of material, and the computation time of coarse truss optimization and different levels of structure refinement.

6.3. Evaluation of the Initialization

To test the sensitivity of our framework to the initialization, we show optimized trusses starting from initializations with different numbers of intermediate joints and bars (before subdivision operations are conducted). Figure 13 shows that adding different intermediate joints and bars results in different trusses. However, these trusses have similar structure and they all function as robust discrete approximations of the optimal truss for further subdivision.

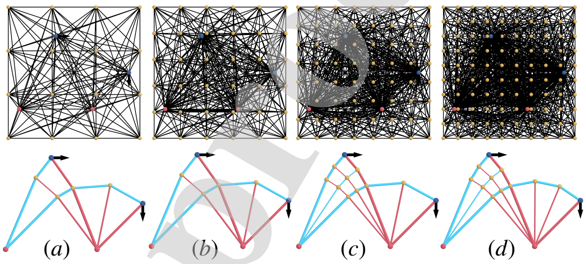


Figure 13: Comparing different initializations. From left to right we initialize with 4×4, 6×6, 8×8, and 10×10 intermediate joints and 124, 364, 732, and 1228 intermediate bars.

6.4. Comparisons

6.4.1. Comparisons with Previous Numerical Methods

We compare the performance of our method with several previous methods using the functional specifications, solutions, and running times provided in their papers. We denote the methods in [3], [5], [34], and [6] as D2013, G2003, H2015 and S2017, respectively. In Figure 1 and Figure 14, we compare our method with [3] for a 2D and a 3D bridge design problem. In Figure 15 and 16, we compare our method with [34] and [5] on a benchmark problem—Hemp cantilever design. For 3D truss optimization, we compare our method with [6] on a simple 3D model in Figure 17. The comparison is shown in Table 2. We can observe that our method is orders of magnitude faster, even though we can achieve a lower volume than previous work. Unfortunately, these methods do not have code or executables publicly available, so we cannot test them on the same machine. However, it seems unlikely that this significant difference in running time can be overcome by slightly faster hardware.

6.4.2. Comparisons with Analytical Solutions

For the cantilever design problems in Figure 15 and 16, the total volumes of analytical solutions are 4.498115 and 4.232168. The volumes of our discrete designs for these two cases are 4.498635 and 4.3223, which are closer to the analytical solutions than the previous work, [5] and [34]. For the three force problem, we compare our result with the analytical solution presented in [2]. In Figure 18, we show the computed discrete truss (130 bars, volume: 6.838) on the right which is visually similar to the analytical solution on the left (volume: 6.831).

Discussion and Limitations.. Compared with previous work, the geometry optimization in terms of both the axial forces and joint positions through two linear programming problems alternatively

Fig.	Initial Truss		Optimal Coarse Truss			Truss Subdivision 1			Truss Subdivision 2			Truss Subdivision 3		
	Bars	Volume GSM	Bars	Volume	Time(s)	Bars	Volume	Time(s)	Bars	Volume	Time(s)	Bars	Volume	Time(s)
20	800	6.516	14	5.830	3.76	26	5.732	0.31	50	5.709	1.41	98	5.703	2.97
21	404	3.125	22	2.967	5.86	46	2.913	5.85	118	2.909	8.86	358	2.907	15.16
22	124	3.872	16	3.365	0.64	36	3.248	0.88	100	3.203	5.46	324	3.188	17.28
19(a1)	835	3.375	19	3.210	5.02	45	3.183	1.43	133	3.170	8.79	453	3.163	29.56
19(b1)	835	3.335	25	3.114	5.42	67	3.041	1.81	211	3.021	7.77	739	3.016	17.84
23	466	4.340	25	4.294	4.92	67	4.289	3.22	211	4.288	9.68	739	4.287	35.63
24	2061	19.510	9	19.081	4.28	18	18.700	0.46	37	18.610	2.39	76	18.587	5.56
25	1118	13.891	24	11.742	5.87	60	11.724	7.73	180	11.718	19.09	612	11.713	57.41
26	3674	15.429	10	14.628	1.61	16	14.416	1.53	28	14.325	1.97	52	14.316	2.11
27	3674	30.850	24	29.049	8.70	58	28.507	1.96	174	28.285	14.75	598	28.217	64.01

Table 1: Statistics of results presented in this paper: For each truss design, we report the number of bars, the total volume of material consumption, as well as the computational time for different stages. Please note that the results from optimal coarse trusses are already better than the results from the ground structure method without introducing additional joints and bars. Topological refinement through subdivision provides additional reduction in material consumption generally in less than one minute in our current implementation which still has a significant room for further improvement.

Fig.	Method	Bars (initial)	Bars (final)	Time	Volume
1	D2013	258	96	1376s	408.807
	Ours	258	201	**7.7s**	**333.395**
14	D2013	31	19	n/a	34.977
	Ours	804	25	**3.4s**	**34.593**
15	G2003	>1 billion	n/a	>6h	4.4998
	Ours	105	2178	**30s**	**4.4986**
16	H2015	12,456,601	4244	4875s	4.3228
	Ours	105	2178	**30s**	**4.3223**
17	S2017	>7 billion	40	>1h	n/a
	Ours	1118	24	**13.6s**	**11.742**

Table 2: Comparisons with previous work. Compared with previous approaches, our framework consistently creates truss designs with smaller volumes with significantly shorter computational times.

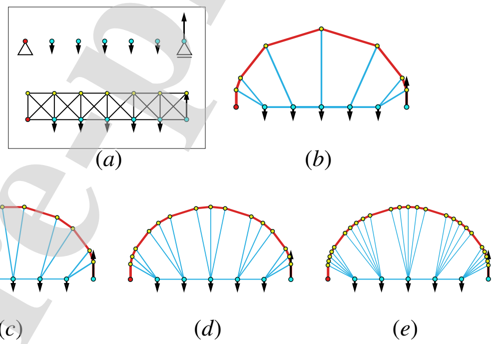


Figure 14: A comparison with the method in [3] on a 2D bridge model. (a) The input functional specification and initial truss for the method in [3]. (b) Optimal truss in [3] (number of bars: 19, volume: 34.977). (c) Our optimal coarse truss (number of bars: 25, volume: 34.593). Refined structures after 1 and 2 rounds of subdivisions are shown in (d) and (e).

applied in ALP provides more degrees of freedom compared with the original formulation of the ground structure method which solves a single linear programming problem. Splitting a highly nonlinear programming problem into two linear ones also attains better efficiency compared with the original nonlinear formulation. Moreover, the two categories of topological operations, local and global, complementarily allow both flexible yet stable topology changes and manipulation. Most importantly, the subdivision approach, which has been overlooked by previous work, is a natural choice for topology refinement from coarse to fine, which creates valid topologies at different levels both efficiently and robustly. Despite the efficiency and efficacy, our optimization framework cannot guarantee a global optimum. However, in simple special cases where the analytical optimum is known, we observe that the method almost reaches the known global optimum. As our method is based on subdivision of edge-networks on surfaces, the optimal trusses in 3D also need to constitute sheets of surfaces. For general 3D specifications, the subdivision approach requires an initial surface-like structure, or a structure that consists of multiple sheets, which is a challenging problem for future work. The proposed ALP algorithm will work for general 3D structures, however.

7. Conclusions and Future Work

We present a method for the design of optimal trusses satisfying functional specifications with minimized material consumption. The core components of the proposed approach include an alternating linear programming formulation for geometry optimization and two sets of topological operations. The subdivision scheme inspired by Michell's theoretical studies utilized in the global topology refinement step plays a crucial role for the efficiency and efficacy of the proposed approach. The performance of our framework is validated by comparisons with multiple previous studies in different scenarios, which indicate that our method creates trusses with smaller volumes and is faster in terms of computational speed. For future work, it would be exciting to study dynamic structures created by trusses with movable parts that have multiple configurations, e.g., robotic and mechatronic systems with trusses. Moreover, a better theoretical understanding of optimal trusses in 3D would be inspiring for the entire field.

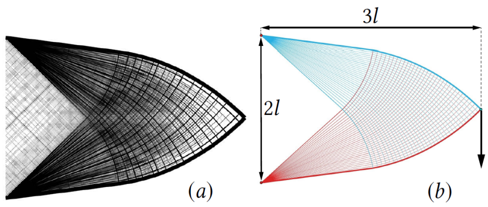


Figure 15: A comparison with the method in [5] on a benchmark design problem (Hemp cantilever). Compared with the best result presented in [5], obtained with 6h50m of computation, shown in (a), which used an initial truss of 116,288,875 bars and obtain a total volume of 4.499827, we obtain a result, shown in (b), with 2178 bars in 30s, which achieves a total volume of 4.498635. For this problem, an analytical solution with an infinite amount of bars exists, which has a total volume of 4.498115.

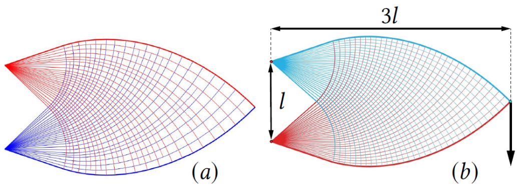


Figure 16: A comparison with the method in [34] on a benchmark design problem (Hemp cantilever). (a) The optimal truss of [34] (number of bars: 4244, running time: 4875s, volume: 4.3228). (b) Our optimized truss (number of bars: 2178, running time: 30s, volume: 4.3223). The analytical solution of optimal volume is 4.3217.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank the anonymous reviewers for their insightful comments and suggestions for improving the paper. This research was supported by KAUST baseline funding, the National Natural Science Foundation of China (62072422), the NSF of Anhui Province of China (2008085MF195) and the Zhejiang Lab (2019NB0AB03).

References

[1] H. Chan, Minimum weight cantilever frames with specified reactions. university of oxford, department of engineering science, eng. laboratory, parks road (1966).

[2] T. Sokół, T. Lewiński, On the solution of the three forces problem and its application in optimal designing of a class of symmetric plane frameworks of least weight, Structural and Multidisciplinary Optimization 42 (6) (2010) 835–853.

[3] B. Descamps, R. F. Coelho, A lower-bound formulation for the geometry and topology optimization of truss structures under multiple loading, Structural and multidisciplinary optimization 48 (1) (2013) 49–58.

[4] W. S. Dorn, Automatic design of optimal structures, Journal de mecanique 3.

[5] M. Gilbert, A. Tyas, Layout optimization of large-scale pin-jointed frames, Engineering Computations 20 (8) (2003) 1044–1064.

[6] T. Sokół, On the numerical approximation of michell trusses and the improved ground structure method, in: World Congress of Structural and Multidisciplinary Optimisation, Springer, 2017, pp. 1411–1417.

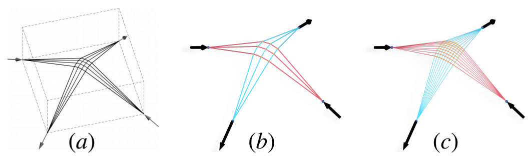


Figure 17: A comparison with the method in [6] on a 3D model. (a) is the optimal structure in [6]. To get this topology, they used a 3D grid of 50X50X50 joints and 7,318,049,198 bars, the running time is close to 2 hours. (b) and (c) are results of our method. To get the initial structure in (b), we use 1118 bars and running time is less than 10s. To get the optimization result in (c) our running time is less than 20s.

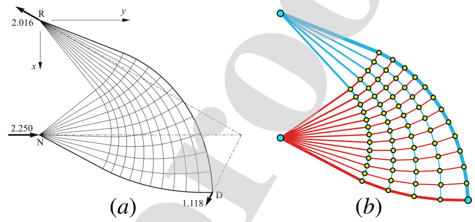


Figure 18: A comparison with an analytic solution in [2]. The volume of the analytic solution with an infinite amount of bars is 6.831 and the volume of our optimized discrete truss (130 bars) is 6.838.

[7] P. Martinez, P. Marti, O. Querin, Growth method for size, topology, and geometry optimization of truss structures, Structural and Multidisciplinary Optimization 33 (1) (2007) 13–26.

[8] A. G. M. Michell, The limits of economy of material in frame-structures, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 8.

[9] J. Smith, J. Hodgins, I. Oppenheim, A. Witkin, Creating models of truss structures with optimization, ACM Trans. Graph. 21.

[10] W. Wang, T. Y. Wang, Z. Yang, L. Liu, X. Tong, W. Tong, J. Deng, F. Chen, X. Liu, Cost-effective printing of 3d objects with skin-frame structures, ACM Trans. Graph. 32 (6).

[11] Q. Zhou, J. Panetta, D. Zorin, Worst-case structural analysis, ACM Trans. Graph. 32 (4) (2013) 137:1–137:12.

[12] J. Martínez, J. Dumas, S. Lefebvre, L.-Y. Wei, Structure and appearance optimization for controllable shape design, ACM Transactions on Graphics (TOG) 34 (6) (2015) 229.

[13] P. Block, J. Ochsendorf, Thrust network analysis: A new methodology for three-dimensional equilibrium, J. Int. Assoc. Shell and Spatial Structures 48 (3) (2007) 167–173.

[14] E. Vouga, M. Höbinger, J. Wallner, H. Pottmann, Design of self-supporting surfaces, ACM Trans. Graph. 31 (4).

[15] Y. Liu, H. Pan, J. Snyder, W. Wang, B. Guo, Computing self-supporting surfaces by regular triangulation, ACM Trans. Graph. 32.

[16] D. Panozzo, P. Block, O. Sorkine-Hornung, Designing unreinforced masonry models, ACM Trans. Graph. 32.

[17] F. de Goes, P. Alliez, H. Owhadi, M. Desbrun, On the equilibrium of simplicial masonry structures, ACM Trans. Graph. 32.

[18] C. Tang, X. Sun, A. Gomes, J. Wallner, H. Pottmann, Form-finding with polyhedral meshes made simple, ACM Tran. Graph. 33.

[19] R. Prévost, E. Whiting, S. Lefebvre, O. Sorkine-Hornung, Make it stand: balancing shapes for 3d fabrication, ACM Transactions on Graphics (TOG) 32 (4) (2013) 81.

[20] M. Bächer, E. Whiting, B. Bickel, O. Sorkine-Hornung, Spin-it: optimizing moment of inertia for spinnable objects, ACM Transactions on Graphics (TOG) 33 (4) (2014) 96.

[21] N. Umetani, T. Igarashi, N. J. Mitra, Guided exploration of physically valid shapes for furniture design, ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012) 31 (4).

[22] J. Yao, D. M. Kaufman, Y. Gingold, M. Agrawala, Interactive design and stability analysis of decorative joinery for furniture, ACM Trans. Graph. 36 (2) (2017) 20:1–20:16. `doi:10.1145/3054740`.
URL `http://doi.acm.org/10.1145/3054740`

[23] C. Jiang, C. Tang, M. Tomičí, J. Wallner, H. Pottmann, Interactive modeling

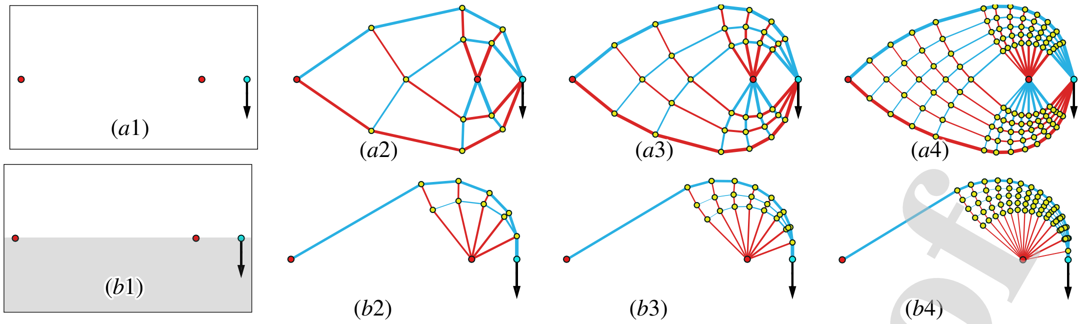


Figure 19: Truss designs for different design regions. Top: truss designs for a functional specification that the design region is the entire plane. Bottom: truss designs for the same functional specification except that the design region is the upper half plane, with the lower half plane being an obstacle to be avoided.

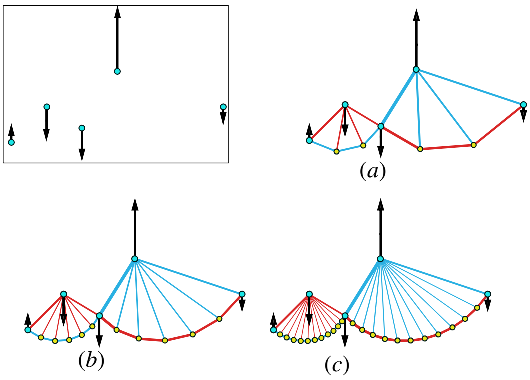


Figure 20: A truss design with the input being a 2D parallel force system.

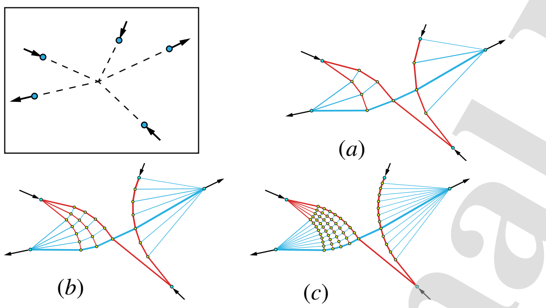


Figure 21: A truss design with the input being a 2D concurrent force system. Note that the concave quadrilateral in (a) is not subdivided.

of architectural freeform structures: Combining geometry with fabrication and statics, in: Advances in Architectural Geometry 2014, Springer, 2015, pp. 95–108.

[24] N. Pietroni, D. Tonelli, E. Puppo, M. Froli, R. Scopigno, P. Cignoni, Statics aware grid shells, in: Computer Graphics Forum, Vol. 34, Wiley Online Library, 2015, pp. 627–641.

[25] C. Jiang, C. Tang, H.-P. Seidel, P. Wonka, Design and volume optimization of space structures, ACM Transactions on Graphics 36 (4) (2017) 1–14.

[26] M. Kilian, D. Pellis, J. Wallner, H. Pottmann, Material-minimizing forms and structures, ACM Trans. Graphics 36 (6) (2017) article 173, proc. SIGGRAPH Asia. `doi:http://dx.doi.org/10.1145/3130800.3130827`.

[27] J. C. Maxwell, I.-on reciprocal figures, frames, and diagrams of forces, Earth and Environmental Science Transactions of the Royal Society of Edinburgh 26 (1) (1870) 1–40.

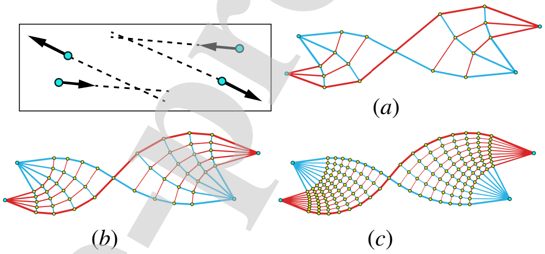


Figure 22: A truss design with the input being a 2D non-concurrent force system. Note that the orthogonality of the two families of trusses, in compression and tension, is improved with subdivision.

[28] W. Hemp, Michell's structural continua, Optimum Structures (1973) 70–101.

[29] W. Prager, G. I. Rozvany, Optimization of structural geometry, in: Dynamical systems, Elsevier, 1977, pp. 265–293.

[30] G. Rozvany, Exact analytical solutions for some popular benchmark problems in topology optimization, Structural and Multidisciplinary Optimization 15 (1) (1998) 42–48.

[31] T. Lewiński, G. Rozvany, Exact analytical solutions for some popular benchmark problems in topology optimization ii: three-sided polygonal supports, Structural and Multidisciplinary Optimization 33 (4-5) (2007) 337–349.

[32] T. Lewiński, G. Rozvany, Exact analytical solutions for some popular benchmark problems in topology optimization iii: L-shaped domains, Structural and Multidisciplinary Optimization 35 (2) (2008) 165–174.

[33] T. Lewiński, G. Rozvany, Analytical benchmarks for topological optimization iv: square-shaped line support, Structural and Multidisciplinary Optimization 36 (2) (2008) 143–158.

[34] L. He, M. Gilbert, Rationalization of trusses generated via layout optimization, Structural and Multidisciplinary Optimization 52 (4) (2015) 677–694.

[35] T. Zegard, G. H. Paulino, GRAND-Ground structure based topology optimization for arbitrary 2D domains using MATLAB, Structural and Multidisciplinary Optimization 50.

[36] M. ApS, The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28). (2015).
URL `http://docs.mosek.com/7.1/toolbox/index.html`

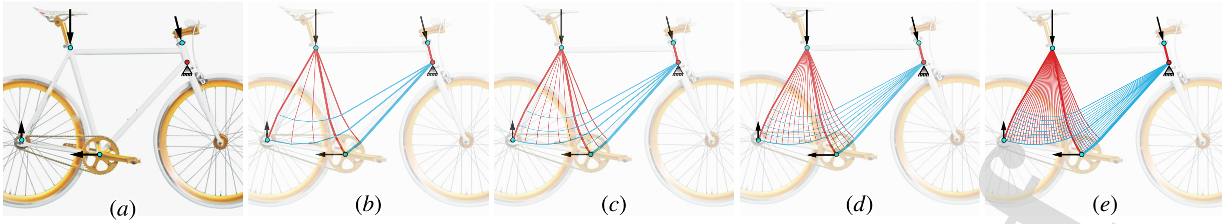


Figure 23: A bike frame design.

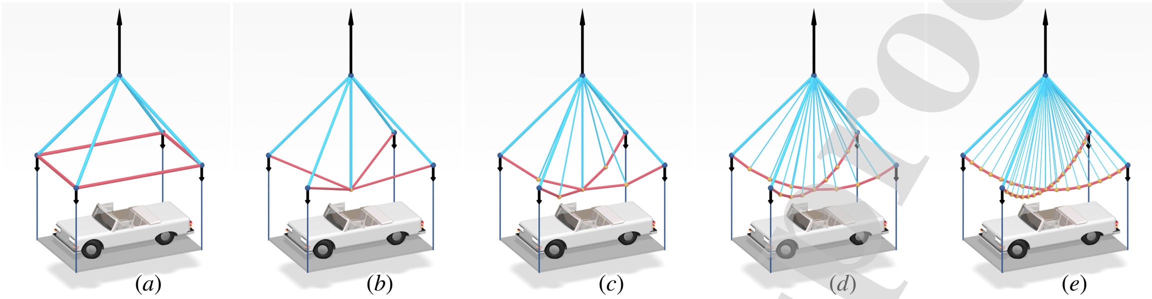


Figure 24: A truss design with the input being a 3D parallel force system.

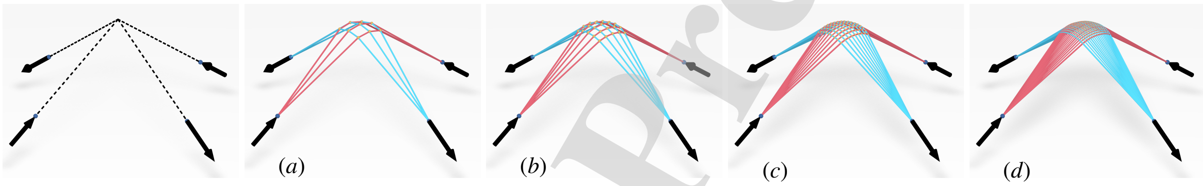


Figure 25: A truss design with the input being a 3D concurrent force system.

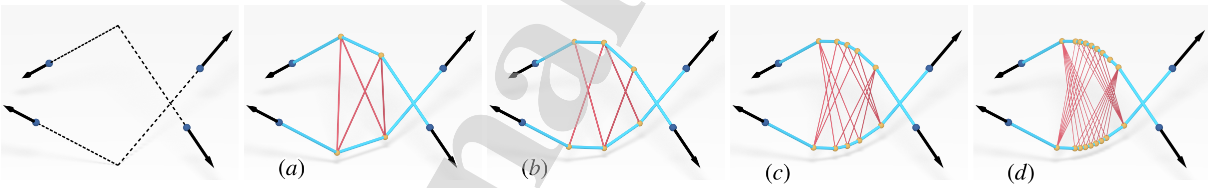


Figure 26: A truss design with the input being a 3D non-concurrent force system.

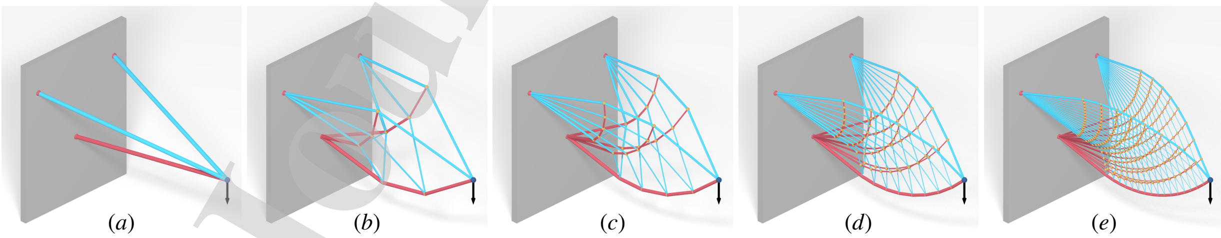


Figure 27: A 3D cantilever design.

Highlights of this manuscript:

- We propose two categories of complementary topology operations, local and global. While local operations have been used in previous work, our global operations based on subdivision are our original contribution.
- We introduce a novel algorithm for geometry optimization based on alternating linear programming (ALP) that jointly optimizes joint positions and bar cross sections.
- Based on these two technical contributions, we build a framework for lightweight truss design, a longstanding and important problem in structural engineering, architecture, graphics, and design. Compared with recent state-of-the art approaches, our method creates trusses with smaller volumes, can handle more complex functional specifications, and is over two orders of magnitude faster.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: