Titre: Title:	On the scalability of truss geometry and topology optimization with global stability constraints via chordal decomposition
Auteurs: Authors:	Alemseged Gebrehiwot Weldeyesus, Jacek Gondzio, & Miguel F. Anjos
Date:	2024
Type:	Article de revue / Article
Référence: Citation:	Gebrehiwot Weldeyesus, A., Gondzio, J., & Anjos, M. F. (2024). On the scalability of truss geometry and topology optimization with global stability constraints via chordal decomposition. Structural and Multidisciplinary Optimization, 68, 7 (15 pages). https://doi.org/10.1007/s00158-024-03947-z

Structural and Multidisciplinary Optimization (2025) 68:7
https://doi.org/10.1007/s00158-024-03947-z

RESEARCH PAPER

On the scalability of truss geometry and topology optimization with global stability constraints via chordal decomposition

Alemseged Gebrehiwot Weldeyesus[1] · **Jacek Gondzio**[1] · **Miguel F. Anjos**[1,2]

Received: 21 March 2024 / Revised: 1 October 2024 / Accepted: 2 December 2024 / Published online: 24 December 2024

Abstract
Geometry optimization was recently introduced to existing truss topology optimization with global stability constraints. The resulting problems are formulated as highly nonlinear semidefinite programming problems that demand extensive computational effort to solve and have been solved only for small problem instances. The main challenge for effective computation is the positive semidefinite constraints which involve large sparse matrices. In this paper, we apply several techniques to tackle the challenge. First, we use the well-known chordal decomposition approach to replace each positive semidefinite constraint on a large sparse matrix by several positive semidefinite constraints on smaller submatrices together with suitable linking constraints. Moreover, we further improve the efficiency of the decomposition by applying a graph-based clique merging strategy to combine submatrices with significant overlap. Next, we couple these techniques with an optimization algorithm that fully exploits the structure of the smaller submatrices. As a result, we can solve much larger problems, which allows us to extend the existing single-load case to the multiple-load case, and to provide a computationally tractable approach for the latter case. Finally, we employ adaptive strategies from previous studies to solve successive problem instances, enabling the joints to navigate larger regions, and ultimately obtain further improved designs. The efficiency of the overall approach is demonstrated via computational experiments on large problem instances.

Keywords Geometry and topology optimization · Global stability · Nonlinear semidefinite programming · Interior-point methods · Chordal decomposition

1 Introduction

We consider truss geometry and topology optimization problems where the design variables are both the cross-sectional areas of the potential bars and the coordinates of the joints. These problems are frequently formulated as nonlinear and non-convex optimization problems (Dobbs and Felton 1969; Kirsch 1990b; Ben-Tal et al. 1993; Bendsøe et al. 1994; Pedersen 1972; Sergeyev and Pedersen 1996; Achtziger 1998; Tejani et al. 2018; Miguel and Miguel 2012), and are known to yield interesting designs in practice. This is unlike the other formulations that restrict the joints to be fixed, which can be solved for larger numbers of potential bars (Bendsøe and Sigmund 2003; Jarre et al. 1998; Gilbert and Tyas 2003; Sokół and Rozvany 2013; Weldeyesus and Gondzio 2018) but the resulting designs require significant post-processing efforts to obtain successful designs. Note there are many alternative approaches for shape (geometry) and topology optimization for various structures and applications (Bendsøe and Sigmund 2003; Allaire et al. 2021; Fujioka et al. 2021; Torisaki et al. 2023; Benaissa et al. 2024) .

To solve truss geometry and topology optimization problems, some studies such as Achtziger (2007), Ali Ahrari and Deb (2015), and Weldeyesus et al. (2020) simultaneously optimize all the design variables, i.e. the cross-sectional areas of the potential bars and the coordinates of the joints and obtain (locally) optimal solutions. Other studies, e.g. (Ringerts 1985), Kočvara and Zowe (1996), optimize the design variables alternately by fixing one set of the design variables and optimizing the other, and then proceeding

Responsible editor: Mehmet Polat Saka

✉ Alemseged Gebrehiwot Weldeyesus
a.weldeyesus@ed.ac.uk

[1] School of Mathematics and Maxwell Institute for Mathematical Sciences, The University of Edinburgh, Peter Guthrie Tait Road, Edinburgh EH9 3FD, UK

[2] GERAD, 3000, chemin de la Côte-Sainte-Catherine, Montréal, QC H3T 2A7, Canada

vice-versa. This approach is known to produce sub-optimal designs. There are studies that consider extended techniques (Achtziger 2007; He and Gilbert 2015) that are applied adaptively to further improve the solutions.

Truss design problems are extended in many studies to include other types of constraints such as stress constraints, local buckling constraints (Kirsch 1990a; Guo et al. 2001a; Stolpe and Svanberg 2001, 2003; Zhou 1996; Achtziger 1999; Rozvany 1996; Guo et al. 2001b, 2005; Mela 2014), nodal stability constraints Tyas et al. 2006; Descamps and Coelho 2014), and global stability constraints (e.g. Ben-Tal et al. 2000; Stingl 2006; Evgrafov 2005; Tugilimana et al. 2018; Kočvara 2002; Weldeyesus et al. 2019, 2020; Poulsen et al. 2020) to address the causes of failures in the structures.

Recently, geometry optimization was introduced in Weldeyesus et al. (2020) to truss topology problems that consider global stability constraints, initially proposed in Kočvara (2002). The problems in Weldeyesus et al. (2020) involve highly nonlinear semidefinite programming and are very challenging to solve Yamashita and Yabe (2015). For this reason, the new formulation in Weldeyesus et al. (2020) was solved only for the single-load case and only for small instances, using a second-order interior-point-based optimization method. Note that these problems are nonlinear and non-convex, and therefore, no guarantee can be provided that the obtained solutions are global optima.

In this paper, we make use of a technique known as chordal decomposition (Fukuda et al. 2001; Kim et al. 2011; Vandenberghe and Andersen 2015; Zheng et al. 2021) that was also applied recently to topology optimization in Kočvara (2020) to the nonlinear semidefinite programming formulation in Weldeyesus et al. (2020) in order to tackle the computational challenges arising from the large but sparse matrix inequalities. The method exploits the sparsity pattern of the large matrices and decomposes the large matrix inequality constraints into inequalities on several small submatrices accompanied by the resulting linking constraints. Moreover, we apply a clique graph-based merging technique (Garstka et al. 2020) to merge the submatrices with significant overlap. Suitable clique mergers improve the efficiency of the decomposition, see e.g. Sliwak et al. (2021) for more details on such merging strategies. In addition to the decomposition and merging techniques, we adopt a procedure used in Weldeyesus and Stolpe (2015) in the optimization algorithm to exploit the small submatrix inequalities and reduce the system of linear equations arising from applying Newton method. The proposed combination of chordal decomposition, clique merging, and the specialized optimization algorithm results in dramatic computational savings, and therefore truss geometry and topology optimization problem formulation that takes into account global stability constraints becomes computationally tractable, and can be solved for larger instances relevant in practice. Moreover, this computational progress leads us to extend the single-load problem formulation and model instances in Weldeyesus et al. (2020) to multiple-load cases, and to solve them.

When the coordinates of the joints are allowed to move in large regions in geometry optimization, it often causes numerical instabilities. To address such instabilities, limits on moves are usually imposed on the joints at the cost of making the optimal designs becoming more dependent on the initial configuration of the joints. To minimize this dependence, an adaptive strategy was utilized in Weldeyesus et al. (2020) based on He and Gilbert (2015). In this paper, we adopt the strategy to iteratively solve successive problems by progressively updating the regions where joints are allowed to move, removing inactive nodes, merging joints that are too close, and melting collinear joints. This allows the joints to navigate much larger regions in the design space and results in improved designs with reduced weight and better layout.

To summarize, we consider the formulation in Weldeyesus et al. (2020) and make the following contributions:

- We introduce chordal decomposition to write the large but sparse matrix inequalities as a combination of several small submatrix inequalities and the associated linking constraints.
- We apply a merging strategy to combine the submatrices with significant overlaps.
- We customize an optimization algorithm to utilize the small submatrices and eliminate the considerably larger blocks present in the Newton equations.

These contributions considerably improve the scalability of the problems. As a result,

- we additionally extend the formulations from the single-load case to the multiple-load case and solve such problems for large instances.

This paper is organized as follows. In Sect. 2, the essential mathematical background for geometry optimization of trusses, and the underlying (nonlinear semidefinite programming) multiple-load truss geometry and topology optimization with global stability constraints are presented. The general framework of the chordal decomposition and merging strategy, and their application to the truss design problems addressed in this paper are described in Sect. 3. A short summary of the optimization method and the iterative adaptive strategy is presented in Sect. 4. The computational experiments and relevant parameters are presented in Sect. 5, and the conclusions are stated in Sect. 6.

2 Problem formulation

For completeness, ease of readability, and to present the extended multiple-load case truss geometry and topology optimization with global stability constraints problem formulation, we closely adopt Sect. 2 of Weldeyesus et al. (2020).

We follow the ground structure approach (Dorn et al. 1964). In a given $N \in \{2,3\}$-dimensional design space, we distribute a certain number, say d, of joints that are connected by m potential bars with cross-sectional areas $a_i,\ i = 1,\dots,m$. Let $\bar{v}_j, j = 1,\dots,d$ be the coordinates of the joints. Then, we have $\bar{v}_j = (x_j, y_j)$ if $N = 2$, and $\bar{v}_j = (x_j, y_j, z_j)$ if $N = 3$. In this paper, these coordinates will be considered as the initial positions of the joints since some of the joints are allowed to move in geometry optimization. We assume that there is a subset of joints that are loaded or supported as fixed ones, and the remaining joints, say $d_0 < d$, are allowed to move within prescribed move limits defined by the neighbourhood

$$\mathscr{V} = \mathscr{V}_1 \cap \mathscr{V}_2, \tag{1}$$

Such that

- $\mathscr{V}_1$ is a region defined by balls of radii r around the joints and is given by

$$\mathscr{V}_1 = \{v \in \mathbb{R}^{d_0 N} |\ ||v_j - \bar{v}_j||^2 \le r_j^2,\ j = 1,\dots,d_0\}, \tag{2}$$

  with $||\cdot||$ the Euclidean norm.
- $\mathscr{V}_2$ a region described by a set of box constraints defined by

$$\mathscr{V}_2 = \{v \in \mathbb{R}^{d_0 N} | \bar{v}_{j,k} - v_{j,k}^{min} \le v_{j,k} \le \bar{v}_{j,k} + v_{j,k}^{max}, \quad j = 1,\dots,d_0,\ k = 1,\dots,N\}. \tag{3}$$

For a given bar $i,\ i = 1,\cdots,m$, let $v_i^{(2)}$ and $v_i^{(1)}$ be the coordinates of the start and end joints of the bar. To avoid singularity (or non-differentiability, for example when $v_i^{(2)} = v_i^{(1)}$ in (5)), we set the radius r_j of each of the balls in (2) to satisfy

$$0 < r_j = \frac{1}{2}\min\{||\bar{v}_j - \bar{v}_p||,\ p \in I\} - \varepsilon, \tag{4}$$

where I is the set of indices of the joints connected to joint j, and $\varepsilon > 0$.

We compute the length $l_i(v)$ of every bar $i,\ i = 1,\cdots,m$, as

$$l_i(v) = ||v_i^{(2)} - v_i^{(1)}||. \tag{5}$$

The corresponding vector of direction cosines $\gamma_i^e(v)$ is given by

$$\gamma_i^e(v) = \frac{1}{l_i(v)}(v_i^{(2)} - v_i^{(1)})^T. \tag{6}$$

Defining $n = Nd - n_0$, where n_0 is the number of fixed degrees of freedom, we construct the corresponding global vector $\gamma_i(v) \in \mathbb{R}^n$ appropriately by embedding $(-\gamma_i^e(v), \gamma_i^e(v))^T$ and setting all of the rest of the entries to zero.

For a given external load $f_\ell \in \mathbb{R}^n, \ell \in \mathscr{L} = \{1,\cdots,n_L\}$, we assume the associated displacement $u_\ell \in \mathbb{R}^n$ to satisfy the (linear) elastic equilibrium equation

$$K(a,v)u_\ell = f_\ell,\ \forall \ell \in \mathscr{L}. \tag{7}$$

The global stiffness matrix $K(a, v)$ is given by

$$K(a,v) = \sum_{i=1}^{m} a_i \frac{E}{l_i(v)}\gamma_i(v)\gamma_i^T(v), \tag{8}$$

where E is the Young's modulus of the material. We write the global geometrical stiffness matrix $G(a, v, u_\ell), \ell \in \mathscr{L}$ as

$$G(a,v,u_\ell) = \sum_{i=1}^{m} \frac{a_i E \gamma_i(v)^T u_\ell}{l_i^2(v)}(\delta_i(v)\delta_i(v)^T + \eta_i(v)\eta_i(v)^T). \tag{9}$$

The vectors $\delta_i(v)$ and $\eta_i(v)$ need to be computed in such a way that the vectors $\gamma_i(v)$, $\delta_i(v)$, and $\eta_i(v)$ are mutually orthogonal. The method proposed in this paper computes these vectors for both of the two- and three-dimensional problems, i.e,. $N \in \{2,3\}$ as follows.

- For $N = 2$, we have only $\delta_i^e(v)$. Defining the rotation matrix $A_{2\times 2}$ as

$$A_{2\times 2} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \tag{10}$$

  we compute the vector $\delta_i^e(v)$ as

$$\delta_i^e(v) = A_{2\times 2}\gamma_i^e(v). \tag{11}$$

- For $N = 3$, we define the following three rotation matrices

$$A_{3\times 3}^{(1)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix},\ A_{3\times 3}^{(2)} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix},\ A_{3\times 3}^{(3)} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{12}$$

  Then, we first determine $\delta_i(v)$ as

$$\delta_i^e(v) = \frac{A_{3\times 3}^{(j)}\gamma_i^e(v)}{||A_{3\times 3}^{(j)}\gamma_i^e(v)||}, \tag{13}$$

where j is the index of $|(\gamma_i^e(v))_j|$ with the smallest magnitude, and then compute $\eta_i^e(v)$ using the vector product

$$\eta_i^e(v) = \gamma_i^e(v) \times \delta_i^e(v). \tag{14}$$

We now present the nonlinear semidefinite programming multiple-load minimum weight truss geometry and topology optimization with global stability constraints:

$$\begin{aligned} \underset{a,v,u_\ell}{\text{minimize}}\ & l(v)^T a \\ \text{subject to}\ & f^T u_\ell \le \zeta,\ \forall \ell \in \mathscr{L} \\ & K(a,v)u_\ell = f_\ell,\ \forall \ell \in \mathscr{L} \\ & K(a,v) + \tau_\ell G(a,v,u_\ell) \succeq 0,\ \forall \ell \in \mathscr{L} \\ & v \in \mathscr{V} \\ & a \ge 0, \end{aligned} \tag{15}$$

where $\varsigma > 0$ is a compliance bound. The matrix inequality constraints account for global stability assuming a linearised buckling analysis. The design load factor is set to a value $\tau_\ell \ge 1$, $\ell \in \mathscr{L}$. Then, the resulting optimal design will be stable for each load $\tau_\ell f_\ell$, $\ell \in \mathscr{L}$.

3 Chordal decomposition and merging strategy

In this section, we describe (i) the chordal decomposition technique that decomposes the large matrix inequality constraints into several inequalities on small submatrices, and (ii) the merging strategy that combines the submatrices with significant overlaps, in the context of the truss design problems addressed in this paper.

3.1 Chordal decomposition

In order to describe the decomposition technique, first we present the relevant basic concepts and definitions from graph theory.

Remark 1 The notations $\mathscr{V}$ and v in the next paragraph apply only to the content of the paragraph, and not to the rest of the paper. This is to be consistent with the notations that are used in the literature on graph theory. Moreover, the coordinates of the joints v in Sect. 2 tend to represent (not identically) the vertices of graphs that are defined below.

An undirected graph $G(\mathscr{V}, \mathscr{E})$ is characterized by two sets, namely set of vertices $\mathscr{V} = \{1, \cdots, n\}$ and set of edges $\mathscr{E} \subseteq \mathscr{V} \times \mathscr{V}$, where $(v_i, v_j) \in \mathscr{E}$ implies $(v_j, v_i) \in \mathscr{E}$. A path in a graph G is a sequence of vertices $\{v_1, v_2, \cdots v_k\} \subseteq \mathscr{V}$ such that the edges $(v_1, v_2), (v_2, v_3), \cdots, (v_{k-1}, v_k)$ are elements in the set of edges $\mathscr{E}$ of the graph. The path $\{v_1, v_2, \cdots v_k\}$, $k \ge 3$ is a cycle if $(v_k, v_1) \in \mathscr{E}$, i.e. if the first vertex corresponds to the last vertex. A chord in a cycle is an edge joining nonconsecutive vertices. An undirected graph is chordal if every cycle of length $k \ge 4$ has a chord. A clique $\mathscr{C}$ is a subset of the set of vertices $\mathscr{V}$ where every two distinct vertices in the sub-graph induced by $\mathscr{C}$ are connected. A clique is maximal if it is not a subset of another clique.

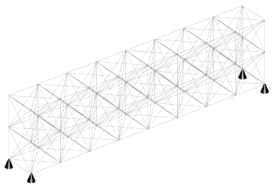
Next, we describe how we can apply the chordal decomposition technique to the underlying nonlinear semidefinite programming problem (15). Introducing the $n \times n$ slack positive semidefinite matrix variable Z_ℓ, $\ell \in \mathscr{L}$ to the matrix inequality, the formulation (15) can be rewritten as

$$\begin{aligned} \underset{a,v,u_\ell}{\text{minimize}}\ & l(v)^T a \\ \text{subject to}\ & f^T u_\ell \le \zeta,\ \forall \ell \in \mathscr{L} \\ & K(a,v)u_\ell = f_\ell,\ \forall \ell \in \mathscr{L} \\ & K(a,v) + \tau_\ell G(a,v,u_\ell) - Z_\ell = 0,\ \forall \ell \in \mathscr{L} \\ & Z_\ell \succeq 0,\ \forall \ell \in \mathscr{L} \\ & v \in \mathscr{V} \\ & a \ge 0. \end{aligned} \tag{16}$$

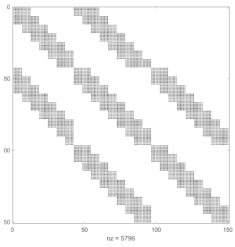
The matrix Z_ℓ inherits the aggregated sparsity pattern of the global stiffness matrix $K(a, v)$ and the global geometrical stiffness matrix $G(a, v, u_\ell)$. In this paper, we consider the bars that connect (semi) adjacent joints as the potential bars. Hence, the matrix Z_ℓ becomes a sparse matrix, for e.g. see Fig. 1b, which makes it suitable for applying chordal decomposition.

To apply chordal decomposition, the undirected graph induced by the non-zero entries of the large positive semidefinite matrix Z_ℓ, $\ell \in \mathscr{L}$ must be chordal. If the initial design and configuration is of the form shown in Fig. 4a, then the sparsity graph of Z_ℓ is chordal. However, the sparsity graph is often not chordal when we consider finer grids, for example, that of Fig. 4b. In that case, we find its chordal extension Z_ℓ^H, that is, we consider additional non-zeros entries of Z_ℓ (equivalent to adding edges to the graph). This can be done in several heuristic ways since finding the minimum chordal extension is NP-complete (Yannakakis 1981).

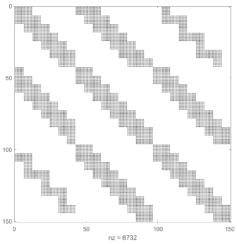
In our case, the matrix Z_ℓ is not only sparse but also well structured, and banded (or semi-arrow type if nodes are melted or merged in the adaptive strategy). Hence, a customized chordal extension technique, for example dense block completion, that closely corresponds to domain decomposition can be used. However, and for the sake of generic usefulness, we follow one of the approaches that require only the information about the sparsity pattern of Z_ℓ to get its chordal extension Z_ℓ^H. The approach is based on Cholesky factorization with zero fill-in and we follow



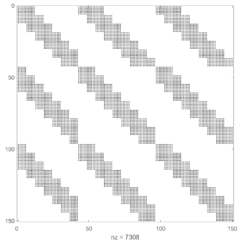
(a) Initial design domain (and supports) for the problem in Figure 4b



(b) Sparsity pattern of the resulting 150×150 matrix Z



(c) Sparsity pattern of the chordal extension Z^H via Cholesky factorization with zero fill. The matrix decomposes into 16 submatrices of sizes between 24 and 30 shown in Figure 2a.



(d) Sparsity pattern of the matrix aggregated from the 8 merged submatrices of sizes 30 and 36 shown in Figure 2b

Fig. 1 Sparsity pattern of the stiffness matrices and their chordal extension for the problem in Fig. 4b

the steps used in Constante et al. (2021) to find Z_ℓ. It is described in Algorithm 1 and Remark 2.

Algorithm 1 Chordal extension and decomposition. See also Remark 2.

Input The sparsity structure of matrix Z_ℓ. This is generated by replacing the values of the Young's Moduls E the length of the bars, l_i. all the non-zeros values of the vectors $\gamma_i(v)$ by 1 in the stiffness matrix $K(a,v)$ in (7) by 1 for each $i = 1, \cdots, m$.
Output Maximal cliques $\mathscr{C}_1, \cdots, \mathscr{C}_s$.

1: Generate the graph for the sparsity structure of the matrix Z_ℓ.
2: Compute the Laplacian matrix A_ℓ for the graph.
3: Update the Laplacian matrix as $A_\ell \leftarrow A_\ell + I_n$, where I_n is $n \times n$ identity matrix.
4: Compute the Cholesky factorization of A_ℓ such that,

$$PA_\ell P^T = LL^T,$$

where P is re-ordering (permutation) of matrix of A_ℓ obtained by approximate minimum degree algorithm.
5: Compute the chordal extension Z_ℓ^H as

$$Z_\ell^H = P^T(L + L^T)P.$$

6: Generate the maximal cliques $\mathscr{C}_1, \cdots, \mathscr{C}_s$.

Remark 2 We implemented Algorithm 1 in MATLAB. The generation of the graph and the Laplacian matrix, and the computation of the Cholesky factorization with re-ordering are performed by calling MATLAB built-in functions. The maximal cliques are generated using a MATLAB implementation based on Eppstein et al. (2010). The algorithm is demonstrated with an example in Sect. 3.2.

Now, let us assume that the large positive semidefinite matrix Z_ℓ^H has the sparsity chordal graph $G(\mathscr{V}, \mathscr{E})$ with maximal (or merged, see Sect. 3.2) cliques $\mathscr{C}_1, \cdots, \mathscr{C}_s$ of cardinalities $|\mathscr{C}_1|, \cdots, |\mathscr{C}_s|$. Then, decomposing Z_ℓ^H into smaller submatrices, we can rewrite formulation (16) as

$$\begin{aligned}
&\underset{a,v,u_\ell}{\text{minimize}}\ l(v)^T a \\
&\text{subject to} f^T u_\ell \le \zeta,\ \forall \ell \in \mathscr{L} \\
&\qquad K(a,v)u_\ell = f_\ell,\ \forall \ell \in \mathscr{L} \\
&\qquad K(a,v) + \tau_\ell G(a,v,u_\ell) - Z_\ell^H = 0,\ \forall \ell \in \mathscr{L} \\
&\qquad Z_\ell^H = \sum_{k=1}^{s} E_{\mathscr{C}_k}^T Z_{\ell,k} E_{\mathscr{C}_k},\ \forall \ell \in \mathscr{L} \\
&\qquad Z_{\ell,k} \succeq 0,\ \forall \ell \in \mathscr{L},\ k = 1, \cdots, s \\
&\qquad v \in \mathscr{V} \\
&\qquad a \ge 0,
\end{aligned} \tag{17}$$

where $Z_{\ell,k}$ are $|\mathscr{C}_k| \times |\mathscr{C}_k|$ matrices, and $E_{\mathscr{C}_k}$ are $|\mathscr{C}_k| \times n$ entry-selector matrices defined as

$$(E_{\mathscr{C}_k})_{ij} = \begin{cases} 0, & \text{if } \mathscr{C}_k(i) = j \\ 1, & \text{otherwise,} \end{cases} \tag{18}$$

with $\mathscr{C}_k(i)$ the i-th vertex.

In the implementation, we eliminate Z_ℓ^H and solve the problem

$$\begin{aligned}
&\underset{a,v,u_\ell}{\text{minimize}}\ l(v)^T a \\
&\text{subject to} f^T u_\ell \le \zeta,\ \forall \ell \in \mathscr{L} \\
&\qquad K(a,v)u_\ell = f_\ell,\ \forall \ell \in \mathscr{L} \\
&\qquad K(a,v) + \tau_\ell G(a,v,u_\ell) - \sum_{k=1}^{s} E_{\mathscr{C}_k}^T Z_{\ell,k} E_{\mathscr{C}_k} = 0, \\
&\qquad \forall \ell \in \mathscr{L} \\
&\qquad Z_{\ell,k} \succeq 0,\ \forall \ell \in \mathscr{L},\ k = 1, \cdots, s \\
&\qquad v \in \mathscr{V} \\
&\qquad a \ge 0.
\end{aligned} \tag{19}$$

Remark 3 In formulation (19),

– The linking constraints that arise from the chordal decomposition are embedded in the third constraint, hence there are no additional constraints.

– The new fill-in appearing in the chordal extension Z^H is extracted from the third constraint, and modelled as linear equality constraints in the implementation.

3.2 Clique merging

The chordal extension method described above (based on Cholesky factorization with zero fill-in) produces a matrix Z_ℓ^H that decomposes into many small submatrices but with significant overlaps. For example, for the 150×150 matrix Z in Fig. 1b, the approach produces the extended matrix Z^H in Fig. 1c that decomposes into 16 submatrices of sizes between 24 and 30 with many overlaps, see Fig. 2a. Therefore, to further improve the efficiency of the chordal decomposition, some of the small matrices could be merged to produce fewer submatrices with marginally larger sizes.

In this paper, we apply a clique graph-based merging technique (Garstka et al. 2020) to merge the submatrices with significant overlaps. The merging strategy first computes the pairwise weights w_{ij} as

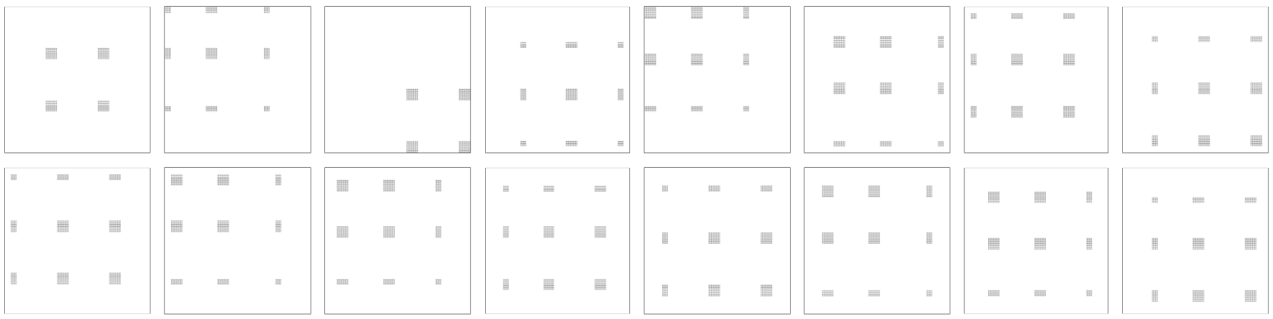
$$w_{ij} = |\mathscr{C}_i|^3 + |\mathscr{C}_j|^3 - |\mathscr{C}_i \cup \mathscr{C}_j|^3, \quad i,j \in \{1,\cdots,s\}$$

and merges the cliques that correspond to $\max\{w_{ij} > 0 | i,j \in \{1,\cdots,s\}\}$. It then updates the set of cliques accordingly and computes the associated new weights. The iterates of the strategy continue until all of the positive weights vanish. The approach reduces the submatrices obtained via Cholesky factorization with zero fill-in by about 50% for the problems addressed in this paper.

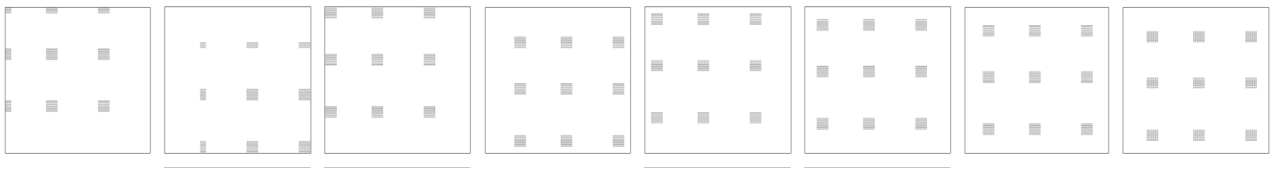
For example, if we apply the merging strategy to the 16 submatrices of sizes between 24 and 30 in Fig. 2a, we get the 8 submatrices displayed in Fig. 2b with slightly higher sizes, i.e. between 30 and 36, but reducing the overall number of matrix entries by 1854 (i.e. about 27%). The sparsity of the matrix that was aggregated from the merged cliques is shown in Fig. 1d. This has 288 more non-zero entries, compared to that of Fig. 1c. These additional entries will be enforced to take on zero value by adding linear equality constraints in the problem formulation. This demonstrates the benefit of the merging strategy to hugely reduce the overall number of matrix entries at the expense of having marginally larger submatrices and a few additional linear equality constraints. Note that the number of the entries of the matrix are that of the triangular matrices, and the savings become more pronounced for larger problem instances, see Table 5.

4 Optimization method and iterative adaptive strategies

We solve problem (19) using an interior-point method (IPM) implemented to solve generic nonlinear semidefinite programming (Weldeyesus et al. 2020) and customised based on a technique adopted from Weldeyesus and Stolpe (2015) to have the capability of exploiting the several but small submatrix inequalities arising from the chordal decomposition. The efficiency improvement originates from the utilization of the small sizes of the submatrices to eliminate the considerably larger blocks present in the Newton equations



(a) The 16 submatrices (of sizes between 24 and 32) of the matrix Z^H in Figure 1c



(b) The 8 merged submatrics using clique graph-based merging technique. The matrices are of sizes that are marginally higher, i.e., between 30 and 36.

Fig. 2 Decomposition and merging

from the augmented system of equations. See Fig. 3 for the comparison of the linear systems with and without the elimination of these block matrices.

In order to minimize the dependence of the optimal design on the initial configuration of the joints, we apply an iterative adaptive strategy to solve successive problems that are obtained by progressively updating the regions where the joints are allowed to move, removing inactive nodes, merging joints that are too close, and melting collinear joints. The details of the procedure and the relevant parameters of the iterative adaptive strategy are described in Section 5, and one of the outcomes of the strategy is shown in Fig. 8b.

5 Computational results

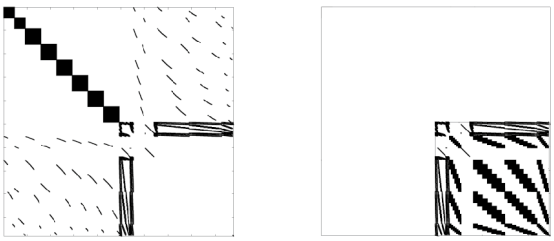
The models and the interior-point optimization method were implemented in MATLAB (R2023a). The computational experiments were performed on a PC equipped with a 13th Gen Intel(R) Core(TM) i5-1345U CPU running at 1.60 GHz with 32 GB RAM. In all of the examples, the input data are without units. The Young's modulus is set to $E = 1$. We use a value of the design loading factor $\tau = 1$.

In the plots that display the optimal designs, we show only the bars with cross-sectional area $\geq 0.001 a_{\max}$. The balls show the active joints connecting these bars.

The move limit radius r (see equation (2)) for the initial configuration is set to the same value for all joints. Its value is provided in the respective examples. For the other configurations that were solved in the iterations of the adaptive strategy, the move limit radius is set to be joint dependent, i.e. for a given joint j, the move limit radius r_j is determined as

$$r_j = \min\{k \min\{||\bar{v}_j - \bar{v}_p||,\ p \in I\}, 0.3\}, \tag{20}$$

where $\bar{v}_j$ and $\bar{v}_p$ are the coordinates of the joints, and I is the set of indices of the joints connected to joint j. The value of k is provided for each example. Moreover,



(a) Without elimination (b) With elimination

Fig. 3 Comparison (and sparsity) of the size of the coefficient matrices of the linear systems arising in the IPM with and without elimination of the large blocks for the problem in Fig. 4b

- Joints that are connected only to negligibly thin bars, i.e. bars with cross-sectional area $< 0.001 a_{\max}$ are removed.
- Joints connecting only two collinear bars of cross-sectional area $\geq 0.001 a_{\max}$ are vanished. The collinear bars are merged to a single bar.
- Joints that are too close to each other, i.e. with intra-distance ≤ 0.25, are merged into a single node. The coordinate of the new node is computed by taking the average coordinates of the merging nodes. But if any of the merging nodes is either supported or loaded, then that will be the coordinate of the new node.
- The iterative adaptive optimization procedure stops when there is no significant change in the volume of the optimal designs of successive iterations.

Next, we present several set of examples on large problem instances to demonstrate the contribution of the paper in Sects. 5.1 and 5.2. We choose the loading conditions so that, unless stability constraints are considered, we can find parallel planar truss designs that lack connectivity and result in unstable designs. Similar examples are discussed in many studies, for example, in Kočvara (2002), Weldeyesus et al. (2019) to demonstrate the need for such constraints.

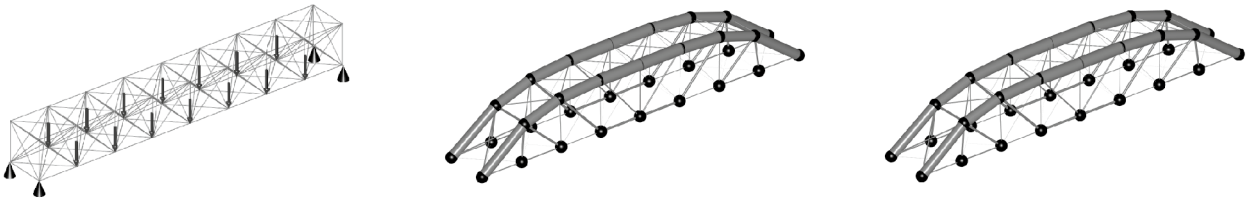
5.1 Efficiency of the chordal decomposition

In this section, we demonstrate the efficiency of the chordal decomposition by comparing the computational times on problem instances that can also be solved without chordal decomposition.

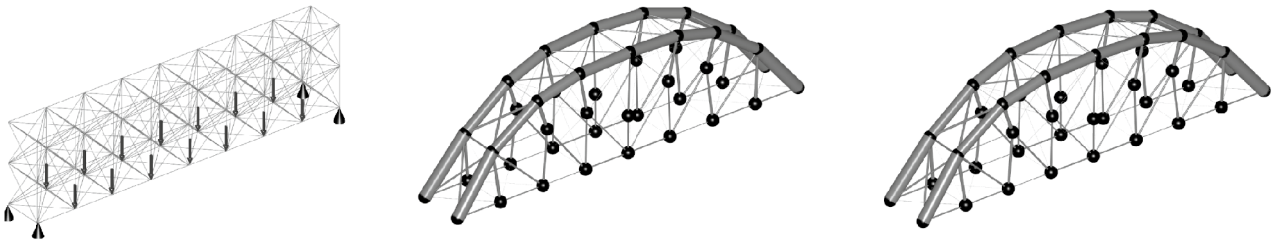
Example 1 We consider the instances shown in Fig. 4. The dimensions of the problem in Fig. 4a are $8 \times 1 \times 1$ and that of Fig. 4b are $8 \times 1 \times 2$. In both cases, the nodal loads are applied simultaneously and have magnitude $(0, 0, -0.001)$. The move limit radius is set to $r = 0.2$ and the bound on the compliance is set to $\eta = 0.003$. The computational statistics are presented in Table 1. Looking at the 4th and 7th columns, when we apply the chordal decomposition then the solution time is 6 times lower for the smaller problem in Fig. 4a and 31 times lower for the larger problem in Fig. 4b. This demonstrates the efficiency of the decomposition. Of course, there are differences in the IPM iterations for the larger problem, but the reduction in solution time (and memory requirement) is mainly driven by the chordal decomposition which can be demonstrated by comparing solution time per IPM iteration. For the plot of the convergence history of the objective function and optimality condition for solving the problems with and without applying chordal decomposition, see Fig. 5. The plots show that there is no significant difference in the behaviour of the convergence rate for both cases.

Table 1 Example 1 - Computational results showing the comparison of solution times with and without chordal decomposition

Problem	Without chordal decomposition			With chordal decomposition		
	Volume	IPM iter	CPU (s)	Volume	IPM iter	CPU (s)
Problem in Fig. 4a	6.9670	46	144	6.9670	45	25
Problem in Fig. 4b	4.3351	68	1660	4.3351	52	53

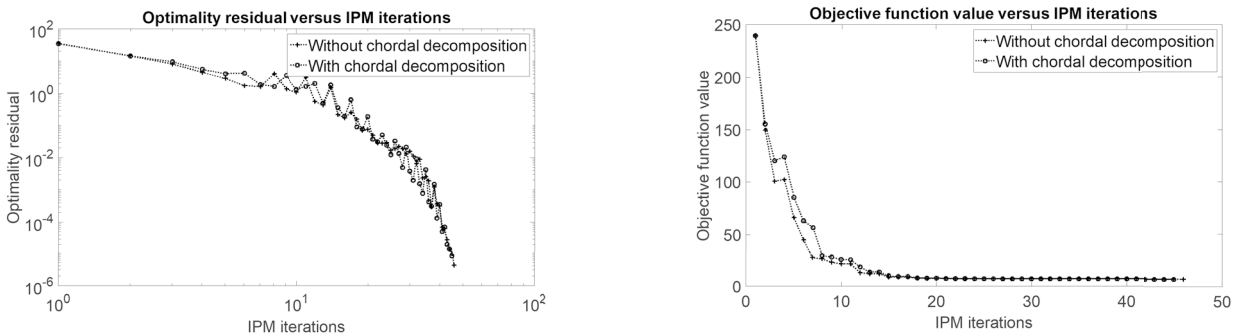


(a) Left: Design domain and loading condition, Middle: Optimal design without chordal decomposition, Right: Optimal design with chordal decomposition

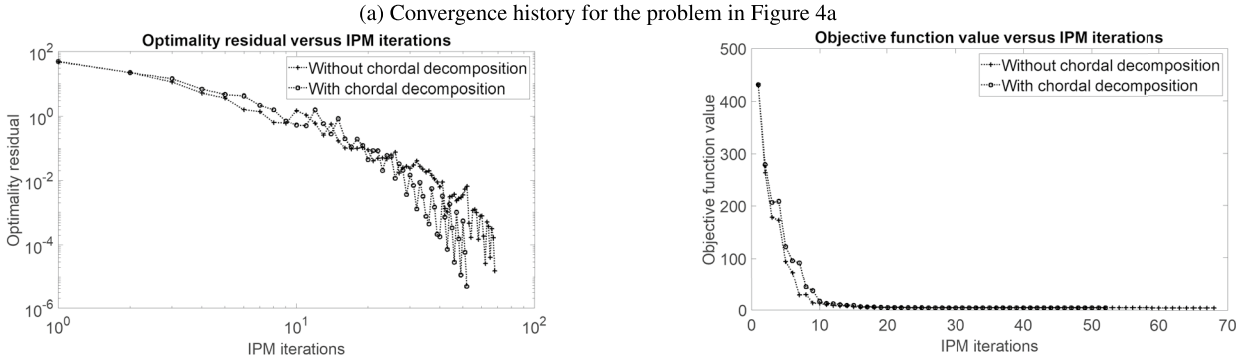


(b) Left: Design domain and loading condition, Middle: Optimal design without chordal decomposition, Right: Optimal design with chordal decomposition

Fig. 4 Example 1 Problem instances and designs obtained with and without chordal decomposition



(a) Convergence history for the problem in Figure 4a



(b) Convergence history for the problem in Figure 4b

Fig. 5 Comparison of convergence history for the problems in Example 1 (or Fig. 4) when solved with and without chordal decomposition

5.2 Large-scale problem instances

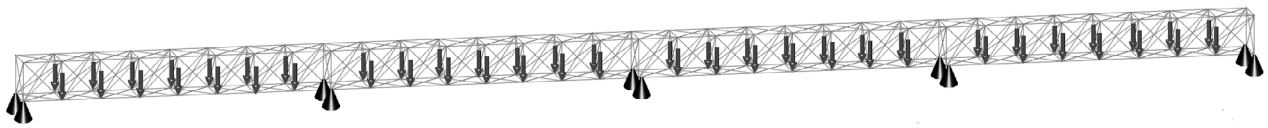
In this section, we present examples with large dataset that would have been prohibitively expensive to solve without using chordal decomposition. Moreover, we apply the iterative adaptive strategy to improve the optimal designs.

Example 2 We solve the bridge problems shown in Fig. 6. The design domains in Fig. 6a–c have dimensions $32 \times 1 \times 1$, $32 \times 1 \times 2$, and $32 \times 1 \times 2$, respectively. The potential bars in the first two designs are the bars that connect only the adjacent nodes. In the third design shown in Fig. 6c, we additionally consider the diagonal bars of the vertical (narrow) rectangles on both sides of the bridge. In all cases, (i) the nodal loads are applied simultaneously to the base plane and have magnitude $(0, 0, -0.00025)$, (ii) the bound on the compliance is $\eta = 0.003$, and (iii) the move limit radius for the initial configuration is $r = 0.1$. The initial move limit radius is fairly small (about 10% of the length of the bars) but larger initial move limit radii hindered the convergence of the optimization algorithm, particularly for the finer grids such as the design in Fig. 6b. However, further move limit radii for the iterates of the adaptative strategy are larger. This is given in Table 2 represented with the values of k in (20).

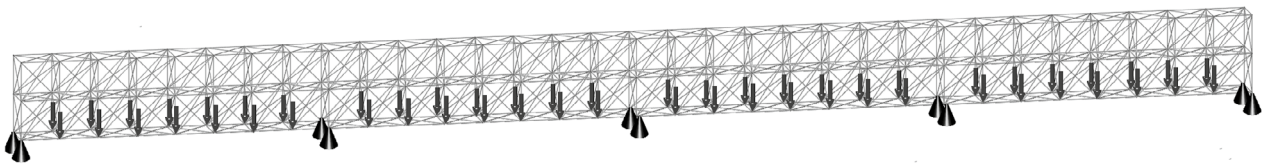
The optimal designs for the initial designs in Fig. 6a–c and the corresponding final solutions of the further adaptive iterations are shown in Figs. 7, 8, and 9, respectively. The computational statistics are presented in Table 2. We use optimality tolerance about $1e-4$ and feasibility of the order $1e-7$. Clearly, the weight of the optimal designs reduces as we perform more iterates of the adaptive strategy. The table shows a reduction of 33% for the problem in Fig. 6a, 15% for the problem in Fig. 6b, and 10% for the problem in Fig. 6c. The reduction is higher for the problem with the coarse grid in Fig. 6a as the joints had to make significant moves to achieve the final optimal design. Another outcome of the adaptive procedure is that, the layouts of the optimal design are improved, where we can see the smooth semicircle-like structures spanning between and above the supports. For similar arches, see (Li and Xie 2021)

In response to the global constraints, we can see the dominant planar structures are connected by some inter-planar bars mostly on the outer surfaces in all designs shown in Figs. 7b, 8b, and 9b. In the top views of the final designs, we can also see diamond-like shapes towards both of the end sides of the bridge.

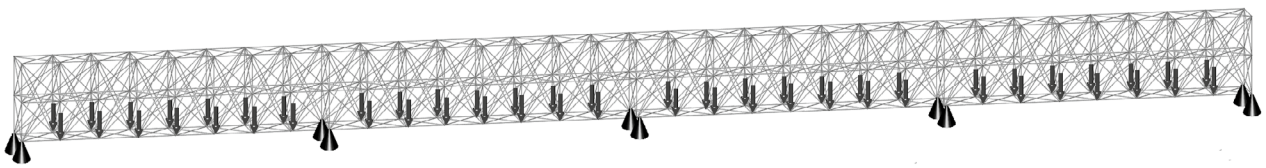
Comparing the weight of final designs in Figs. 7b, 8b, and 9b, the first design is the heaviest and the last (third) design is the lightest. This is an expected outcome since having more potential bars results in lighter designs. Interestingly, the solution time of the problem with more bars in Fig. 6c is not different from that of the problem in Fig. 6b which has the same number of joints. This is mainly because of (by design) the similarity of the chordal decompositions of both problems.



(a) The potential bars are the bars that are connecting adjacent nodes.

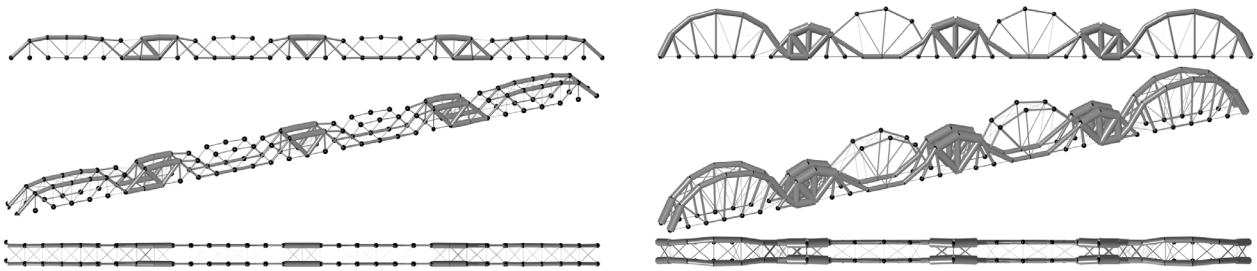


(b) The potential bars are the bars that are connecting adjacent nodes.



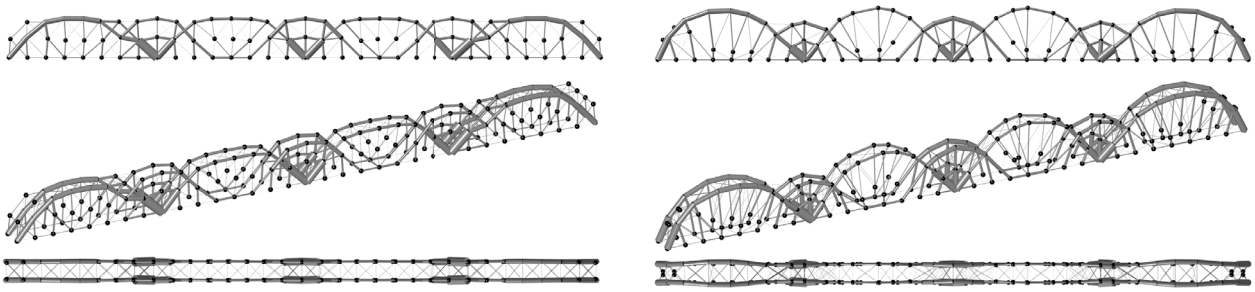
(c) The potential bars are (i) the bars that are connecting adjacent nodes and (ii) the diagonal bars of the vertical (narrow) rectangles on both sides of the bridge.

Fig. 6 Example 2 - Design domain, boundary conditions, and loads



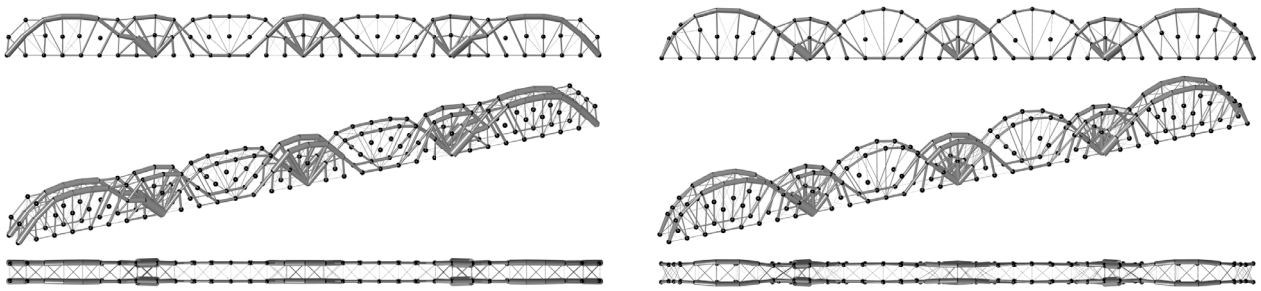
(a) vol = 6.3403. Optimal design for the initial design in Figure 6a (b) vol = 4.2744. Optimal design. 7th iteration of the adaptive strateg.

Fig. 7 Optimal design for the initial and final configuration problem in Fig. 6a of Example 2. From top to bottom: side, 3D, and top views in all figures



(a) vol = 4.2940. Optimal design for the initial design in Figure 6b (b) vol =3.6675. Optimal design. 5th iteration of the adaptive strategy

Fig. 8 Optimal design for the initial and final configuration for the problem in Fig. 6b of Example 2. From top to bottom: side, 3D, and top views in all figures



(a) vol = 4.019 Optimal design for the initial design in Figure 6c (b) vol = 3.6278. Optimal design. 4th iteration of the adaptive strategy

Fig. 9 Optimal design for the initial and final configuration problem in Fig. 6c of Example 2. From top to bottom: side, 3D, and top views in all figures

Example 3 We solve the two-load case problem of dimensions $10 \times 1 \times 3$ which is shown in Fig. 10a. The largest nodal loads (represented by the long arrows) have magnitude $(0, 0, -0.001)$, and the smallest ones (represented by the short arrows) have magnitude $(0, 0, -0.00025)$. The bound on the compliance is set to $\eta = 0.003$ and the move limit radius is set to $r = 0.2$. The solution of the problem for the initial configuration is shown in Fig. 10b. The optimal design is mainly composed of two planar arch structures connected by inter-planar bars. The weight of the initial optimal design is 1.7780. After performing 2 more iterations of the adaptive strategy, we obtained the design shown Fig. 10c of weight 1.7058, reduced by 4% compared to the weight of the initial optimal design. The reduction is smaller

Table 2 Example 2 - Computational results

Stage number	Problem in Fig. 6a				Problem in Fig. 6b				Problem in Fig. 6c			
	Volume	IPM iter	CPU	k	Volume	IPM iter	CPU	k	Volume	IPM iter	CPU	k
1	6.3403	47	52	–	4.2940	72	200	–	4.0419	48	140	–
2	5.3029	44	48	1/5	3.7992	55	157	1/5	3.7181	52	153	1/5
3	4.7470	45	48	1/4	3.7036	53	151	1/5	3.6494	51	141	1/6
4	4.4492	47	59	1/3	3.6769	69	195	1/5	3.6278	61	164	1/5
5	4.3312	53	61	1/3	3.6675	84	242	1/5				
6	4.2876	57	64	1/4								
7	4.2744	85	88	1/5								

Table 3 Example 3- Computational results

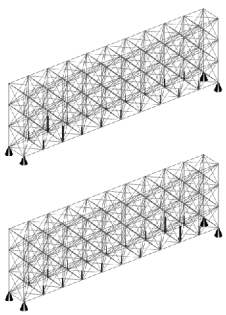
Stage number	Volume	IPM iter	CPU (s)	k
1	1.7780	63	298	–
2	1.7175	58	261	1/5
3	1.7058	63	294	1/6

compared to those in Example 2 for two reasons. Firstly, the initial move limit radius is larger in this example. Secondly, the joints are all over the design space resulting in a very small change of the coordinates of the joints throughout the iterations of the adaptive procedure. In the final design, the arches are smoother and the inner configuration of the joints has been improved inline with the arch curvature. Similar arch curvatures can be seen in Golecki et al. (2023) for two-dimensional problems. The computational results are reported in Table 3.

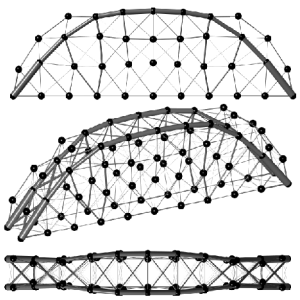
Example 4 In this example, we solve the six-load case of dimensions $18 \times 1 \times 5$ on a slightly more complex design domain shown in Fig. 11a. The nodal loads have magnitude $(0, 0, -0.00025)$, the bound on the compliance is set to $\eta = 0.002$, and the move limit radius for the initial problem is $r = 0.1$. The solutions to the problems for the initial configuration and the final stage of the adaptive strategy are shown in Fig. 11b and c, respectively. The numerical statistics are given in Table 4. Looking at the weights of the optimal designs, the optimal weight of the problem on the initial configuration 10.3818 is reduced by 35% to 6.6700 after applying 7 iterations of the adaptive strategy. The designs consist of two planar arches extending from the supports that are interconnected due to the presence of the stability constraints. Such designs (arches) can be also found in Xie (2022).

5.3 Efficiency of the clique merging strategy

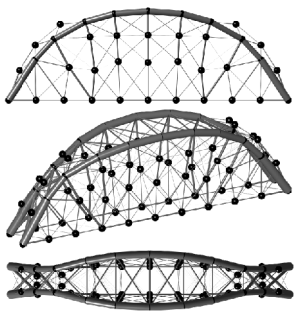
In this section, we demonstrate the efficiency of the clique merging strategy described in Sect. 3.2 that combines the submatrices with significant overlaps to reduce the overall number of matrix entries. The comparison is made for the



(a) Design domain, boundary conditions, and loads. Top: Load case 1, Bottom: Load case 2

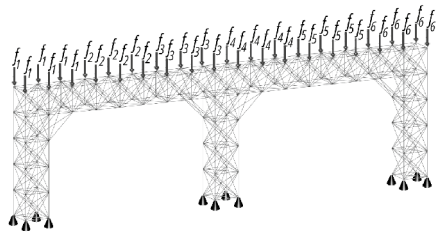


(b) vol = 1.7780. Optimal design for the initial design. From top to bottom: side, 3D, and top views in all figures

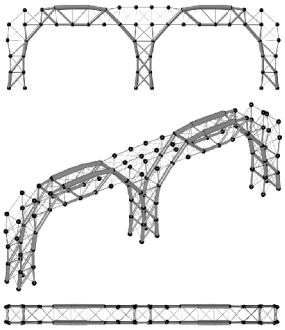


(c) vol = 1.7058. Optimal design. 3rd iteration of the adaptive strategy. From top to bottom: side, 3D, and top views in all figures

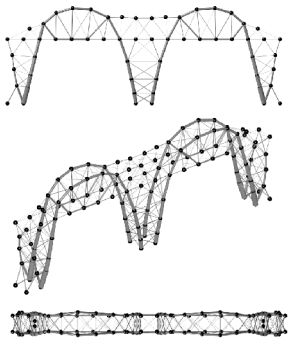
Fig. 10 Example 3 - Loading conditions and optimal designs



(a) Design domain, boundary conditions, and loads



(b) vol = 10.3818. Optimal design for the initial design. From top to bottom: side, 3D, and top views in all figures



(c) vol = 6.6700. Optimal design. 7th iteration of the adaptive strategy. From top to bottom: side, 3D, and top views in all figures

Fig. 11 Example 4 - Loading conditions and optimal designs

Table 4 Example 4- computational result

Stage number	Volume	IPM iter	CPU (s)	k
1	10.3818	57	353	-
2	8.3058	67	412	1/6
3	7.4309	68	413	1/6
4	7.0727	66	383	1/6
5	6.8879	65	390	1/7
6	6.7438	70	387	1/7
7	6.6770	71	435	1/7
8	6.6700	74	417	1/7

problem in Fig. 6c by solving it with and without applying the merging strategy. The result is reported in Table 5. The Columns 2 and 7 show that the merging strategy decreases the number of submatrices by roughly 50% with only a marginal increase in the sizes of the merged submatrices. This results in a significant reduction of the number of matrix entries which can be seen when comparing Columns 3 and 8. Note that the number of the matrix entries reported in the table is only those of the triangular matrices to take symmetry into account. This contributes to reducing the amount of memory requirement. However, despite this huge reduction, the savings in the CPU times are not that pronounced. This is due to (i) the submatrices are in general of small sizes which makes the cost of computing singular value decompositions in the optimization algorithm less dominant, (ii) as described in Sect. 4, the optimization algorithm exploits the small submatrices to eliminate the larger blocks that are present in the Newton equation systems from the augmented system of equations, and therefore, there is no considerable difference in the sizes of the linear systems with and without applying the merging strategy. However, this will not be the case if the problems were solved by a general purpose nonlinear semidefinite programming implementation where larger augmented linear systems would be involved. In this case, the merging strategy will have a considerable role to lower the solution times. For example, we have solved the large augmented system in Fig. 3a for the initial configuration of the problem in Fig. 6c, and the CPU time without applying the merging strategy was 1322 sec. This was reduced to 713sec (or from 22sec to 15sec per iteration) after applying the merging strategy. Note that these

Table 5 Computational results showing the efficiency of the merging strategy for the Problem in Fig. 6c

Stage number	Without merging					With merging				
	# of submatrices, min and max sizes	# entries of the triangular matrix	Volume	IPM iter	CPU	# of submatrices, min and max sizes	# entries of the triangular matrix	Volume	IPM iter	CPU
1	61, (27×27, 33×33)	32,103	4.0427	51	162	32, (30×30, 36×36)	19,704	4.0419	48	140
2	59, (24×24, 33×33)	31,191	3.7190	52	161	32, (24×24, 36×36)	18,570	3.7181	52	153
3	59, (24×24, 33×33)	29,727	3.6503	55	162	32, (24×24, 36×36)	18,570	3.6494	51	141
4	59, (24×24, 33×33)	29,727	-	-	-	32, (24×24, 36×36)	18,570	3.6278	61	164

longer runs additionally demonstrate the efficiency of the customized optimization method, see Column 12 of Table 2. We have also observed that the number of IPM iterations tends to be smaller and the optimization algorithm to be more stable (for e.g. the optimization algorithm failed to converge for the case without merging in stage number 4 in Table 5) when the clique merging strategy is applied.

5.4 General discussion

- The problems solved in the paper are highly nonlinear due to the coordinates of the joints also being design variables. This makes the computation of the Hessian of the Lagrangian (see Sect. 4 of Weldeyesus et al. (2020)) very expensive. Moreover, we create and solve (large) system of linear equations that arise from applying Newton method at each iteration of the interior-point method. Looking at the computational profiles of the problem instances, the results were consistent, i.e. most of the CPU time was spent to determine the Hessian of the Lagrangian and to formulate and solve the linear systems. For example, in solving the problem in Example 3 of the initial configuration, the CPU times spent to compute Hessian of the Lagrangian, to perform additional computations and create the coefficient matrix of the Newton system, and to solve the linear system were 32%, 13%, and 25% respectively, of the total CPU time.
- We observed that the choice of the move limit radius r_i in (2), the bound on the compliance η, the coarseness (fineness) of the joints, the barrier parameter update strategy, initial point for the cross-sectional areas, and steplengths controls (see Sect. 4 of Weldeyesus et al. (2020)) in the interior-point method all affect the convergence of the optimization algorithm and the feasibility of the problems. As we consider the finer grids, the problem becomes more challenging to solve. Hence, in that case, considering a reasonable smaller move limit radius or tightening the bound on the compliance to enhance the stiffness of the structures makes the problem feasible and less challenging to solve.

6 Conclusions

Recently proposed truss geometry and topology optimization with global stability constraints, formulated as nonlinear semidefinite programming problems, have been solved only for small problem instances. This is due to the higher computational effort associated with the positive semidefinite constraints on large sparse matrices. In this paper, we have employed several techniques to address this challenge.

We have introduced chordal decomposition technique that is accompanied with merging strategy to efficiently exploit the sparsity pattern of the matrices involved in the constraints. Furthermore, we have coupled these with an optimization method that utilizes the several but small submatrices that arise from the decomposition to eliminate the considerably large blocks present in the Newton equations appearing at each iteration of the optimization method. As a result, the scalability of the solution method has been improved and allows to model and solve larger instances. Moreover, because of these computational savings, the formulation was further extended to solve multiple-load instances. We also applied adaptive strategies and demonstrated that the optimal designs can be further improved by iteratively solving the problems and enabling the joints to explore larger space in the design domain.

The formulations addressed in the paper exclude local bulking stability analysis which is essential for practical designs. Hence, it will be interesting to extend the formulation to consider local buckling constraints based on the Euler or other buckling criterion, and to use our techniques to solve these problems. Moreover, the practicality of the results will be greater if the design variables for the cross-sectional areas of the bars can be discrete instead of continuous.

Acknowledgements We thank EPSRC grant EP/N019652/1 for partially funding the project.

Declarations

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

Replication of results The parameters, input and output data for all instances are clearly described in Sect. 5.

References

Achtziger W (1998) Multiple-load truss topology and sizing optimization: some properties of minimax compliance. J Optim Theory Appl 98(2):255–280

Achtziger W (1999) Local stability of trusses in the context of topology optimization part ii: a numerical approach. Struct Optim 17(4):247–258

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

Ali Ahrari AAA, 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
Allaire G, Dapogny C, Jouve F (2021) Chapter 1 - shape and topology optimization. In: Bonito A, Nochetto RH (eds) Geometric partial differential equations - part II, handbook of numerical analysis, vol 22. Elsevier, pp 1–132
Ben-Tal A, Kočvara M, Zowe J (1993) Two nonsmooth approaches to simultaneous geometry and topology design of trusses. Springer, Dordrecht, pp 31–42
Ben-Tal A, Jarre F, Kočvara M, Nemirovski A, Zowe J (2000) Optimal design of trusses under a nonconvex global buckling constraint. Optim Eng 1(2):189–213
Benaissa B, Kobayashi M, Al Ali M, Khatir S, Shimoda M (2024) A novel exploration strategy for the Yuki algorithm for topology optimization with metaheuristic structural binary distribution. Eng Optim. https://doi.org/10.1080/0305215X.2024.2349104
Bendsøe M, Sigmund O (2003) Topology optimization: theory, methods and applications. Springer
Bendsøe MP, Ben-Tal A, Zowe J (1994) Optimization methods for truss geometry and topology design. Struct Optim 7(3):141–159
Constante FSG, López JC, Rider MJ (2021) Optimal reactive power dispatch with discrete controllers using a branch−and−bound algorithm: a semidefinite relaxation approach. IEEE Trans Power Syst 36(5):4539–4550
Descamps B, Coelho RF (2014) The nominal force method for truss geometry and topology optimization incorporating stability considerations. Int J Solids Struct 51(13):2390–2399
Dobbs MW, Felton LP (1969) Optimization of truss geometry. ASCE J Struct Di 95:2015–2115
Dorn W, Gomory R, Greenberg H (1964) Automatic design of optimal structures. J de Mécanique 3:25–52
Eppstein D, Löffler M, Strash D (2010) Listing all maximal cliques in sparse graphs in near-optimal time. In: Cheong O, Chwa KY, Park K (eds) Algorithms and computation. Springer, Heidelberg, pp 403–414
Evgrafov A (2005) On globally stable singular truss topologies. Struct Multidisc Optim 29(3):170–177
Fujioka M, Shimoda M, Al Ali M (2021) Shape optimization of periodic-microstructures for stiffness maximization of a macrostructure. Compos Struct 268:113873
Fukuda M, Kojima M, Murota K, Nakata K (2001) Exploiting sparsity in semidefinite programming via matrix completion i: general framework. SIAM J Optim 11(3):647–674
Garstka M, Cannon M, Goulart P (2020) A clique graph based merging strategy for decomposable SDPs. IFAC-PapersOnLine 53(2):7355–7361
Gilbert M, Tyas A (2003) Layout optimization of large-scale pin-jointed frames. Eng Comput 20(8):1044–1064
Golecki T, Gomez F, Carrion J, Spencer BF (2023) Bridge topology optimization considering stochastic moving traffic. Eng Struct 292:116498
Guo X, Cheng G, Yamazaki K (2001) A new approach for the solution of singular optima in truss topology optimization with stress and local buckling constraints. Struct Multidisc Optim 22:364–372
Guo X, Cheng G, Yamazaki K (2001) A new approach for the solution of singular optima in truss topology optimization with stress and local buckling constraints. Struct Multidisc Optim 22(5):364–373
Guo X, Cheng G, Olhoff N (2005) Optimum design of truss topology under buckling constraints. Struct Multidisc Optim 30(3):169–180
He L, Gilbert M (2015) Rationalization of trusses generated via layout optimization. Struct Multidisc Optim 52(4):677–694
Jarre F, Kočvara M, Zowe J (1998) Optimal truss design by interior-point methods. SIAM J Optim 8(4):1084–1107
Kim S, Kojima M, Mevissen M, Yamashita M (2011) Exploiting sparsity in linear and nonlinear matrix inequalities via positive semidefinite matrix completion. Math Program 129(1):33–68
Kirsch U (1990) On singular topologies in optimum structural design. Struct Optim 2:133–142
Kirsch U (1990) On the relationship between optimum structural topologies and geometries. Struct Optim 2(1):39–45
Kočvara M (2002) On the modelling and solving of the truss design problem with global stability constraints. Struct Multidisc Optim 23(3):189–203
Kočvara M (2020) Decomposition of arrow type positive semidefinite matrices with application to topology optimization. Math Program 190:105–134
Kočvara M, Zowe J (1996) How mathematics can help in design of mechanical structures. Pitman Research Notes in Mathematics Series pp 76–93
Li Y, Xie YM (2021) Evolutionary topology optimization for structures made of multiple materials with different properties in tension and compression. Compos Struct 259:113497
Mela K (2014) Resolving issues with member buckling in truss topology optimization using a mixed variable approach. Struct Multidisc Optim 50(6):1037–1049
Miguel LFF, Miguel LFF (2012) Shape and size optimization of truss structures considering dynamic constraints through modern metaheuristic algorithms. Expert Syst Appl 39:9458–9467
Pedersen P (1972) On the optimal layout of multi-purpose trusses. Comput Struct 2(2):695–712
Poulsen P, Olesen J, Baandrup M (2020) Truss optimization applying finite element limit analysis including global and local stability. Struct Multidisc Optim 62:41–54
Ringerts U (1985) On topology optimization for trusses. Eng Optim 9(3):209–218
Rozvany GIN (1996) Difficulties in truss topology optimization with stress, local buckling and system stability constraints. Struct Optim 11(3):213–217
Sergeyev O, Pedersen P (1996) On design of joint positions for minimum mass 3D frames. Struct Optim 11(2):95–101
Sliwak J, Andersen ED, Anjos MF, Létocart L, Traversi E (2021) A clique merging algorithm to solve semidefinite relaxations of optimal power flow problems. IEEE Trans Power Syst 36(2):1641–1644
Sokół T, Rozvany GIN (2013) On the adaptive ground structure approach for multi-load truss topology optimization. In: 10th World congresses of structural and multidisciplinary optimization
Stingl M (2006) On the solution of nonlinear semidefinite programs by augmented Lagrangian method. PhD thesis, Institute of Applied Mathematics II, Friedrich-Alexander University of Erlangen-Nuremberg
Stolpe M, Svanberg K (2001) On the trajectories of the epsilon-relation relaxation approach for stress-constrained truss topology optimization. Struct Multidisc Optim 21:140–151
Stolpe M, Svanberg K (2003) A note on tress-based truss topology optimization. Struct Multidisc Optim 25:62–64
Tejani GG, Savsani VJ, Patel VK, Savsani PV (2018) Size, shape, and topology optimization of planar and space trusses using mutation-based improved metaheuristics. J Comput Design Eng 5(2):198–214
Torisaki M, Shimoda M, Al Ali M (2023) Shape optimization method for strength design problem of microstructures in a multiscale structure. Int J Numer Meth Eng 124(8):1748–1772
Tugilimana A, Filomeno Coelho R, Thrall AP (2018) Including global stability in truss layout optimization for the conceptual design of large-scale applications. Struct Multidisc Optim 57(3):1213
Tyas A, Gilbert M, Pritchard T (2006) Practical plastic layout optimization of trusses incorporating stability considerations. Comput Struct 84(3–4):115–126

Vandenberghe L, Andersen MS (2015) Chordal graphs and semidefinite optimization. Found Trends Optim 1(4):241–433

Weldeyesus A, Gondzio J, He L, Gilbert M, Shepherd P, Tyas A (2020) Truss geometry and topology optimization with global stability constraints. Struct Multidisc Optim 62:1721–1737

Weldeyesus AG, Gondzio J (2018) A specialized primal-dual interior point method for the plastic truss layout optimization. Comput Optim Appl 71(3):613–640

Weldeyesus AG, Stolpe M (2015) A primal-dual interior point method for large-scale free material optimization. Comput Optim Appl 61(2):409–435

Weldeyesus AG, Gondzio J, He L, Gilbert M, Shepherd P, Tyas A (2019) Adaptive solution of truss layout optimization problems with global stability constraints. Struct Multidisc Optim 60:2093–2111

Xie Y (2022) Generalized topology optimization for architectural design. Arch Intell 1:2

Yamashita H, Yabe H (2015) A survey of numerical methods for nonlinear semidefinite programming. J Oper Res Soc Jpn 58(1):24–60

Yannakakis M (1981) Computing the minimum fill-in is np¬complete. SIAM J Algebraic Discrete Methods 2(1):77–79

Zheng Y, Fantuzzi G, Papachristodoulou A (2021) Chordal and factor-width decompositions for scalable semidefinite and polynomial optimization. Annu Rev Control 52:243–279

Zhou M (1996) Difficulties in truss topology optimization with stress and local buckling constraints. Struct Optim 11(2):134–136