

Alternating direction method of multipliers for truss topology optimization with limited number of nodes: a cardinality-constrained second-order cone programming approach

Yoshihiro Kanno¹ · Shinnosuke Fujita²

Received: 22 March 2017 / Revised: 5 October 2017 / Accepted: 22 December 2017 /
Published online: 16 February 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract This paper addresses the compliance minimization of a truss, where the number of available nodes is limited. It is shown that this optimization problem can be recast as a second-order cone programming with a cardinality constraint. We propose a simple heuristic based on the alternative direction method of multipliers. The efficiency of the proposed method is compared with a global optimization approach based on mixed-integer second-order cone programming. Numerical experiments demonstrate that the proposed method often finds a solution having a good objective value with small computational cost.

Keywords Topology optimization · Truss optimization · Manufacturability · Alternating direction method of multipliers · Cardinality-constrained second-order cone programming · Mixed-integer second-order cone programming

✉ Yoshihiro Kanno
kanno@mi.tokyo-u.ac.jp
Shinnosuke Fujita
fujita.s.ag@m.titech.ac.jp

¹ Mathematics and Informatics Center, The University of Tokyo, Hongo 7-3-1, Tokyo 113-8656, Japan

² Laboratory for Future Interdisciplinary Research of Science and Technology, Institute of Innovative Research, Tokyo Institute of Technology, Nagatsuta 4259, Yokohama 226-8503, Japan

1 Introduction

It is common to use the ground structure method (Topping 1983; Kirsch 1989; Bendsøe et al. 1994) for truss topology optimization, where the cross-sectional areas of the truss members are treated as design variables to be optimized. Particularly, compliance minimization with continuous design variables is a convex problem (Bendsøe et al. 1994; Ohsaki 2011), and can be solved efficiently. An optimal solution of this problem often consists of too many members (including ones that are too thin) connected by many nodes¹ and, hence, is regarded as too complex a design from the viewpoint of practical manufacturability. Also, the fabrication cost of a truss usually increases as the number of nodes increases. To obtain a practically acceptable truss design, Asadpoure et al. (2015) proposed to minimize the weighted sum of the structural weight and the fabrication cost related to the number of members. In this method, the number of members is approximated by using a regularized Heaviside step function. Torii et al. (2016) used the same approach to take into account the number of nodes. In this paper, we consider the compliance minimization of a truss subject to an explicit upper bound constraint on the number of nodes.

The number of nodes in structural optimization has also been discussed in the layout optimization of trusses. In the classical layout optimization, we minimize the total weight of the members when the allowable stress is specified. When the potential locations of nodes of a truss are not limited, the optimal solution becomes a so-called *truss-like continuum* with infinitely many nodes (Hegemier and Prager 1969; Michell 1904). Prager (1978, 1977) showed that, by adding the weight of the nodes to the objective function, the problem has an optimal solution with a finite number of nodes. To avoid complex truss design, Parkes (1975) proposed to introduce modification of member lengths such that, at each node, a constant is added to the length of each member connected to the node. As a post-processing step for this method, He and Gilbert (2015) proposed to make use of geometry optimization. Similarly, Mazurek et al. (2011) defined a so-called performance index, by using member lengths and axial forces, to assess the cost of a structure; see also Mazurek (2012). The number of nodes in a truss is not specified explicitly in the methods in the literature (Asadpoure et al. 2015; Torii et al. 2016; Parkes 1975; He and Gilbert 2015; Mazurek 2012; Mazurek et al. 2011) cited above.

In this paper, based on the ground structure method we deal with the compliance minimization problem of a truss subjected to an upper bound constraint on the number of nodes (i.e., a cardinality constraint on the set of nodes). This design optimization problem essentially consists of two decisions: We first select a set of nodes, satisfying the cardinality constraint, among the candidate nodes in a ground structure, and then find the optimal cross-sectional areas of the members connected to the selected nodes. The first decision gives a combinatorial attribute to the design optimization problem. In this paper, we show that this optimization problem can be recast as a *mixed-integer second-order cone programming* (MISOCP) problem; see Sect. 3.2. Since an SOCP problem can be solved efficiently with a primal-dual

¹ See, for example, Fig. 6 in Sect. 6.

interior-point method (Anjos and Lasserre 2012; Ben-Tal and Nemirovski 2001), we can compute a globally optimal solution of an MISOCP problem with, e.g., a branch-and-bound method. Several software packages are available for this purpose (Andersen et al. 2003; Gurobi Optimization, Inc. 2016). However, due to its large computational cost, the MISOCP approach can be applied only to small- to medium-size truss optimization problems. The reader may refer to Bertsimas and Shioda (2009) and Miyashiro and Takano (2015) for applications of MISOCP to variable selection in statistics, and to Kanno (2013, 2016a, b) and Kočvara (2017) for applications in structural optimization.

We refer to the number of nonzero components of a real vector as the ℓ_0 -norm of the vector.² We refer to an upper bound constraint on the ℓ_0 -norm of a vector, i.e., an ℓ_0 -norm constraint, as a cardinality constraint (i.e., an upper bound constraint on the cardinality of the support of the vector). Cardinality constraints, as well as ℓ_0 -norm minimization, frequently appear in diverse fields including variable selection in statistics, image processing, compressed sensing, and portfolio selection (Natarajan 1995; Chartrand 2007; Bruckstein et al. 2009; Candès et al. 2008; Gotoh et al. 2018; Le Thi et al. 2015; Zheng et al. 2014; Burdakov et al. 2016; Bertsimas and Shioda 2009; Cui et al. 2013). An application of ℓ_0 -minimization to structural design generating link mechanisms can be found in Ohsaki et al. (2014). In this paper, we show that truss topology optimization with a limited number of nodes can be formulated as cardinality-constrained SOCP; see Sect. 3.1.

The *alternating direction method of multipliers* (ADMM) is an algorithm for convex optimization (Boyd et al. 2010). For various nonconvex optimization problems, it is known that ADMM can often serve as a simple but powerful heuristic (Takapoui et al. 2018; Kanamori and Takeda 2014; Chartrand and Wohlberg 2013; Magnússon et al. 2016; Chartrand 2012; Diamond et al. 2018). This motivates us to develop a simple heuristic based on ADMM, to find approximate solutions to the problem of truss topology optimization with a limited number of nodes. The proposed method is expected to find a local optimal solution having a reasonable objective value with small computational cost. In control theory, ADMM has been used for various sparsity-promoting optimal control methods, including the design of sparse feedback gains (Lin et al. 2013), sparse output feedback (Arastoo et al. 2015), and sparse gain matrices for the extended Kalman filter (Masazade et al. 2012).

The paper is organized as follows: Sect. 2 provides an overview of necessary background for the ADMM. Sect. 3 formulates the truss topology optimization problem with a limited number of nodes as cardinality-constrained SOCP, and recasts it as MISOCP. Section 4 presents a heuristic based on ADMM to solve the cardinality-constrained SOCP formulation. Section 5 is devoted to a discussion on the treatment of overlapping members in a ground structure. Section 6 reports results of numerical experiments. Some conclusions are drawn in Sect. 7.

² Although this number is not a norm, it is common to call it the ℓ_0 -norm (Bruckstein et al. 2009; Burdakov et al. 2016; Candès et al. 2008; Chartrand 2007; Gotoh et al. 2018; Le Thi et al. 2015; Zheng et al. 2014).

In our notation, \top denotes the transpose of a vector or a matrix. We use $\mathbf{1} = (1, 1, \dots, 1)^\top$ to denote the all-ones vector. For vectors $\mathbf{x} = (x_i) \in \mathbb{R}^n$ and $\mathbf{y} = (y_i) \in \mathbb{R}^n$, we write $\mathbf{x} \geq \mathbf{y}$ if $x_i \geq y_i$ ($i = 1, \dots, n$). We use $\|\mathbf{x}\|$ to denote the Euclidean norm (or the ℓ_2 -norm) of \mathbf{x} , i.e., $\|\mathbf{x}\| = \sqrt{\mathbf{x}^\top \mathbf{x}}$. We denote by $\|\mathbf{x}\|_0$ the number of nonzero components of \mathbf{x} , which is the so-called ℓ_0 -norm of \mathbf{x} . For a finite set T , let $|T|$ denote the cardinality of T , i.e., the number of elements in T . If we define $\text{supp}(\mathbf{x}) \subseteq \{1, \dots, n\}$ by $\text{supp}(\mathbf{x}) = \{i \in \{1, \dots, n\} \mid x_i \neq 0\}$, then $\|\mathbf{x}\|_0 = |\text{supp}(\mathbf{x})|$. For a set $S \subseteq \mathbb{R}^n$, we denote by $\delta_S : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ the indicator function of S , which is defined by

$$\delta_S(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in S, \\ +\infty & \text{if } \mathbf{x} \notin S. \end{cases}$$

For a closed set $S \subseteq \mathbb{R}^n$, the projection of a point $\mathbf{z} \in \mathbb{R}^n$ onto S , denoted $\Pi_S(\mathbf{z}) \in \mathbb{R}^n$, is defined by

$$\Pi_S(\mathbf{z}) \in S, \quad \|\mathbf{z} - \Pi_S(\mathbf{z})\| = \min\{\|\mathbf{z} - \mathbf{x}\| \mid \mathbf{x} \in S\}.$$

If S is closed and convex, then $\Pi_S(\mathbf{z})$ exists uniquely for any point $\mathbf{z} \in \mathbb{R}^n$. The n -dimensional second-order cone, denoted \mathcal{L}^n , is defined by

$$\mathcal{L}^n = \{(s_0, \mathbf{s}_1) \in \mathbb{R} \times \mathbb{R}^{n-1} \mid \|\mathbf{s}_1\| \leq s_0\}.$$

The n -dimensional rotated second-order cone, denoted \mathcal{K}^n , is defined by

$$\mathcal{K}^n = \{(\mathbf{x}, y, z) \in \mathbb{R}^{n-2} \times \mathbb{R} \times \mathbb{R} \mid \mathbf{x}^\top \mathbf{x} \leq yz, y \geq 0, z \geq 0\}.$$

We recall that $(\mathbf{x}, y, z) \in \mathcal{K}^n$ if and only if $(y + z, y - z, 2\mathbf{x}) \in \mathcal{L}^n$. We use $\mathcal{U}(a, b)$ to denote the continuous uniform distribution on the interval $(a, b) \subset \mathbb{R}$.

2 Fundamentals of alternating direction method of multipliers

In this section, we briefly outline the *alternating direction method of multipliers* (ADMM) for solving convex optimization problems; see Boyd et al. (2010) for more accounts.

Let $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ and $g : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$ be closed proper convex functions. Consider the following convex optimization problem in variables $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{z} \in \mathbb{R}^m$:

$$\underset{\mathbf{x}, \mathbf{z}}{\text{Minimize}} \quad f(\mathbf{x}) + g(\mathbf{z}) \tag{1a}$$

$$\text{subject to} \quad A\mathbf{x} + B\mathbf{z} = \mathbf{c}. \tag{1b}$$

Here, $A \in \mathbb{R}^{l \times n}$ and $B \in \mathbb{R}^{l \times m}$ are constant matrices, and $\mathbf{c} \in \mathbb{R}^l$ is a constant vector. The augmented Lagrangian of problem (1) is defined as

$$L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^\top (\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\|^2, \quad (2)$$

where $\rho > 0$ is the penalty parameter, and $\mathbf{y} \in \mathbb{R}^l$ is the Lagrange multiplier (also called the dual variable). At each iteration of ADMM, we update \mathbf{x}^k , \mathbf{z}^k , and \mathbf{y}^k as

$$\mathbf{x}^{k+1} := \operatorname{argmin}_{\mathbf{x}} L_\rho(\mathbf{x}, \mathbf{z}^k, \mathbf{y}^k), \quad (3)$$

$$\mathbf{z}^{k+1} := \operatorname{argmin}_{\mathbf{z}} L_\rho(\mathbf{x}^{k+1}, \mathbf{z}, \mathbf{y}^k), \quad (4)$$

$$\mathbf{y}^{k+1} := \mathbf{y}^k + \rho(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{c}). \quad (5)$$

The so-called *scaled form* of ADMM is defined below. Letting $\mathbf{v} = \mathbf{y}/\rho$, we see that (2) is reduced to

$$\tilde{L}_\rho(\mathbf{x}, \mathbf{z}, \mathbf{v}) = f(\mathbf{x}) + g(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c} + \mathbf{v}\|^2 - \frac{\rho}{2} \|\mathbf{v}\|^2. \quad (6)$$

By using \tilde{L}_ρ in (6), the iteration of ADMM given by (3), (4), and (5) is written as

$$\mathbf{x}^{k+1} := \operatorname{argmin}_{\mathbf{x}} \tilde{L}_\rho(\mathbf{x}, \mathbf{z}^k, \mathbf{v}^k), \quad (7)$$

$$\mathbf{z}^{k+1} := \operatorname{argmin}_{\mathbf{z}} \tilde{L}_\rho(\mathbf{x}^{k+1}, \mathbf{z}, \mathbf{v}^k), \quad (8)$$

$$\mathbf{v}^{k+1} := \mathbf{v}^k + \mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{c}. \quad (9)$$

We refer to the form given in (7), (8), and (9) as the scaled form of ADMM, and we refer to \mathbf{v} as the scaled dual variable.

Primarily, ADMM is an algorithm for solving convex optimization problems. It is known that ADMM can often serve as an efficient heuristic for diverse nonconvex optimization problems; see, e.g., Takapoui et al. (2018), Kanamori and Takeda (2014), Chartrand and Wohlberg (2013), Magnússon et al. (2016), Chartrand (2012), and Boyd et al. (2010, Sect. 9). For nonconvex problems, ADMM does not necessarily converge. Also, when it converges, the obtained solution is not necessarily a global optimum. Furthermore, the obtained solution can depend on the penalty parameter and the initial point. Nevertheless, ADMM can be a simple algorithm, and can be efficient in the sense that it often converges to a solution with a good objective value.

3 Design optimization with limited number of nodes

In Sect. 3.1, we define truss topology optimization subject to an upper bound constraint on the number of nodes. In Sect. 3.2, we show that this problem can be recast as an MISOCP problem.

3.1 Problem setting

Following the ground structure approach, consider an initial truss consisting of many candidate members that are connected by nodes with given locations. The cross-sectional areas of the members are treated as the design variables to be optimized. It is worth noting that the ground structure may involve some overlapping members, an example is shown in Fig. 1. The necessity, as well as the treatment, of overlapping members in a ground structure is thoroughly discussed in Sect. 5. We use m , l , and d to denote the number of members, the number of nodes, and the number of degrees of freedom of the nodal displacements, respectively.

Let x_i ($i = 1, \dots, m$) denote the member cross-sectional areas. We use $K(\mathbf{x}) \in \mathbb{R}^{d \times d}$ to denote the stiffness matrix, which can be written as

$$K(\mathbf{x}) = \sum_{i=1}^m \frac{E}{c_i} x_i \mathbf{b}_i \mathbf{b}_i^T.$$

Here, c_i is the undeformed member length, E is the Young modulus, and $\mathbf{b}_i \in \mathbb{R}^d$ is a constant vector reflecting the member connectivity and the direction cosine of member i . For a given external load vector $\mathbf{p} \in \mathbb{R}^d$, the compliance of the truss, denoted $\pi(\mathbf{x})$, is defined by

$$\pi(\mathbf{x}) = \sup\{2\mathbf{p}^T \mathbf{u} - \mathbf{u}^T K(\mathbf{x}) \mathbf{u} \mid \mathbf{u} \in \mathbb{R}^d\}. \tag{10}$$

Let $V (> 0)$ denote the specified upper bound on the structural volume. The conventional compliance minimization problem is formulated as follows:

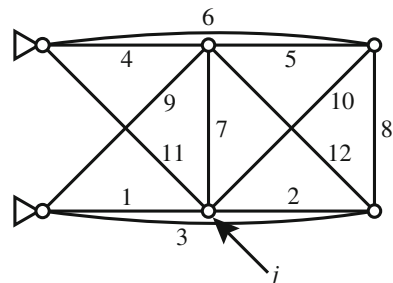
$$\text{Minimize}_{\mathbf{x}} \quad \pi(\mathbf{x}) \tag{11a}$$

$$\text{subject to} \quad \mathbf{x} \geq \mathbf{0}, \tag{11b}$$

$$\mathbf{c}^T \mathbf{x} \leq V. \tag{11c}$$

This is a convex problem, and can be recast as follows (Ben-Tal and Nemirovski 2001, Sect. 3.4.3):

Fig. 1 An example of ground structure with overlapping members



$$\text{Minimize}_{\mathbf{x}, \mathbf{q}, \mathbf{w}} \quad \sum_{i=1}^m w_i \quad (12a)$$

$$\text{subject to} \quad w_i x_i \geq \frac{c_i}{E} q_i^2, \quad i = 1, \dots, m, \quad (12b)$$

$$\mathbf{x} \geq \mathbf{0}, \quad (12c)$$

$$\sum_{i=1}^m q_i \mathbf{b}_i = \mathbf{p}, \quad (12d)$$

$$\mathbf{c}^\top \mathbf{x} \leq V. \quad (12e)$$

Constraints (12b) and (12c) can be rewritten equivalently as the rotated second-order cone constraints

$$(\sqrt{c_i/E} q_i, w_i, x_i) \in \mathcal{K}^3, \quad i = 1, \dots, m.$$

These constraints also can be rewritten equivalently as the second-order cone constraints

$$w_i + x_i \geq \left\| \begin{bmatrix} w_i - x_i \\ 2\sqrt{c_i/E} q_i \end{bmatrix} \right\|, \quad i = 1, \dots, m.$$

Thus, the conventional compliance minimization, (12), can be recast as an SOCP problem (Ben-Tal and Nemirovski 2001, Sect. 3.4.3); see also Kanno (2016a) and Kočvara (2017).

We are now in a position to consider an upper bound constraint on the number of nodes in a truss design. Let n denote the specified upper bound. For the j th node ($j = 1, \dots, l$), define $I(j) \subseteq \{1, \dots, m\}$ as the set of indices of the members connected to node j . For example, in the case of Fig. 1, we have $I(j) = \{1, 2, 7, 10, 11\}$. Defining z_j ($j = 1, \dots, l$) by

$$z_j = \sum_{i \in I(j)} x_i, \quad (13)$$

we see that the number of nodes is equal to $\|\mathbf{z}\|_0$. For notational simplicity, we write (13) as

$$\mathbf{z} = \mathbf{Z}\mathbf{x}$$

with a constant matrix $\mathbf{Z} \in \mathbb{R}^{l \times m}$. The upshot is that compliance minimization subject to an upper bound constraint on the number of existing nodes is formulated as follows:

$$\underset{\mathbf{x}, \mathbf{z}}{\text{Minimize}} \quad \pi(\mathbf{x}) \quad (14a)$$

$$\text{subject to} \quad \mathbf{x} \geq \mathbf{0}, \quad (14b)$$

$$\mathbf{c}^\top \mathbf{x} \leq V, \quad (14c)$$

$$\mathbf{z} = \mathbf{Z}\mathbf{x}, \quad (14d)$$

$$\|\mathbf{z}\|_0 \leq n. \quad (14e)$$

As mentioned above, the conventional compliance minimization in (11) can be recast as an SOCP problem. Therefore, problem (14) can be reduced to a cardinality-constrained SOCP problem. In Sect. 3.2, we present its MISOCP reformulation.

3.2 MISOCP formulation

In this section, we show that problem (14) can be recast as an MISOCP problem.

For node j ($j = 1, \dots, l$), we introduce a new variable, $s_j \in \{0, 1\}$, to indicate whether the node vanishes ($s_j = 0$) or exists ($s_j = 1$). The relationship between s_j and z_j can be given as

$$0 \leq z_j \leq Ms_j,$$

where $M > 0$ is a sufficiently large constant. The upper bound constraint on the number of existing nodes is written in terms of s_1, \dots, s_l as

$$\sum_{j=1}^l s_j \leq n.$$

This observation, in conjunction with the SOCP reformulation of problem (11), allows us to conclude that problem (14) is reduced to the following MISOCP problem:

$$\text{Minimize}_{\mathbf{x}, \mathbf{q}, \mathbf{w}, \mathbf{z}, \mathbf{s}} \sum_{i=1}^m w_i \tag{15a}$$

$$\text{subject to } w_i + x_i \geq \left\| \begin{bmatrix} w_i - x_i \\ 2\sqrt{c_i/Eq_i} \end{bmatrix} \right\|, \quad i = 1, \dots, m, \tag{15b}$$

$$\sum_{i=1}^m q_i \mathbf{b}_i = \mathbf{p}, \tag{15c}$$

$$\mathbf{c}^\top \mathbf{x} \leq V, \tag{15d}$$

$$\mathbf{z} = \mathbf{Z}\mathbf{x}, \tag{15e}$$

$$\mathbf{z} \leq \mathbf{M}\mathbf{s}, \tag{15f}$$

$$\sum_{j=1}^l s_j \leq n, \tag{15g}$$

$$\mathbf{s} \in \{0, 1\}^l. \tag{15h}$$

Here, the optimization variables are $\mathbf{x} \in \mathbb{R}^m$, $\mathbf{q} \in \mathbb{R}^m$, $\mathbf{w} \in \mathbb{R}^m$, $\mathbf{z} \in \mathbb{R}^l$, and $\mathbf{s} \in \mathbb{R}^l$. Although problem (15) is a fairly straightforward extension of the existing SOCP formulation for problem (11), it cannot be found in literature to the best of the authors' knowledge.

4 Simple heuristic based on alternating direction method of multipliers

In this section, we present an ADMM as a heuristic for solving problem (14).

For notational simplicity, define $F \subseteq \mathbb{R}^m$ and $G \subseteq \mathbb{R}^l$ by

$$F = \{\mathbf{x} \in \mathbb{R}^m \mid \mathbf{x} \geq \mathbf{0}, \mathbf{c}^\top \mathbf{x} \leq V\},$$

$$G = \{\mathbf{z} \in \mathbb{R}^l \mid \|\mathbf{z}\|_0 \leq n\}.$$

We see that problem (14) can be written as follows:

$$\text{Minimize } \pi(\mathbf{x}) + \delta_F(\mathbf{x}) + \delta_G(\mathbf{z}) \tag{16a}$$

$$\text{subject to } \mathbf{Z}\mathbf{x} - \mathbf{z} = \mathbf{0}. \tag{16b}$$

The augmented Lagrangian for problem (16) is formulated as

$$L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = \pi(\mathbf{x}) + \delta_F(\mathbf{x}) + \delta_G(\mathbf{z}) + \mathbf{y}^\top (\mathbf{Z}\mathbf{x} - \mathbf{z}) + \frac{\rho}{2} \|\mathbf{Z}\mathbf{x} - \mathbf{z}\|^2, \tag{17}$$

where $\rho > 0$ is the penalty parameter, and $\mathbf{y} \in \mathbb{R}^l$ is the Lagrange multiplier. Letting $\mathbf{v} = \mathbf{y}/\rho$, we see that (17) is reduced to

$$\tilde{L}_\rho(\mathbf{x}, \mathbf{z}, \mathbf{v}) = \pi(\mathbf{x}) + \delta_F(\mathbf{x}) + \delta_G(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{Z}\mathbf{x} - \mathbf{z} + \mathbf{v}\|^2 - \frac{\rho}{2} \|\mathbf{v}\|^2.$$

Using \tilde{L}_ρ , we can write the iterations of ADMM in scaled form as

$$\mathbf{x}^{k+1} := \operatorname{argmin}_x \left\{ \pi(\mathbf{x}) + \delta_F(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{Z}\mathbf{x} - \mathbf{z}^k + \mathbf{v}^k\|^2 \right\}, \tag{18}$$

$$\mathbf{z}^{k+1} := \operatorname{argmin}_z \left\{ \delta_G(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{Z}\mathbf{x}^{k+1} - \mathbf{z} + \mathbf{v}^k\|^2 \right\}, \tag{19}$$

$$\mathbf{v}^{k+1} := \mathbf{v}^k + \mathbf{Z}\mathbf{x}^{k+1} - \mathbf{z}^{k+1}. \tag{20}$$

The first step of ADMM in (18) means that \mathbf{x}^{k+1} is an optimal solution of the following convex optimization problem:

$$\text{Minimize } \pi(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{Z}\mathbf{x} - \mathbf{z}^k + \mathbf{v}^k\|^2 \tag{21a}$$

$$\text{subject to } \mathbf{x} \geq \mathbf{0}, \tag{21b}$$

$$\mathbf{c}^\top \mathbf{x} \leq V. \tag{21c}$$

This problem can be recast as an SOCP problem. To see this, using the SOCP formulation of problem (11), we rewrite problem (21) as follows:

$$\text{Minimize } \sum_{i=1}^m w_i + \frac{\rho}{2} t \tag{22a}$$

$$\text{subject to } t \geq \|\mathbf{Z}\mathbf{x} - \mathbf{z}^k + \mathbf{v}^k\|^2, \tag{22b}$$

$$w_i + x_i \geq \left\| \begin{bmatrix} w_i - x_i \\ 2\sqrt{c_i/Eq_i} \end{bmatrix} \right\|, \quad i = 1, \dots, m, \tag{22c}$$

$$\sum_{i=1}^m q_i \mathbf{b}_i = \mathbf{p}, \tag{22d}$$

$$\mathbf{c}^\top \mathbf{x} \leq V, \tag{22e}$$

where $t \in \mathbb{R}$ is an auxiliary variable. Since constraint (22b) is a rotated second-order cone constraint³

³ It can also be rewritten as

$$t + 1 \geq \left\| \begin{bmatrix} t - 1 \\ 2(\mathbf{Z}\mathbf{x} - \mathbf{z}^k + \mathbf{v}^k) \end{bmatrix} \right\|,$$

which is a second-order cone constraint.

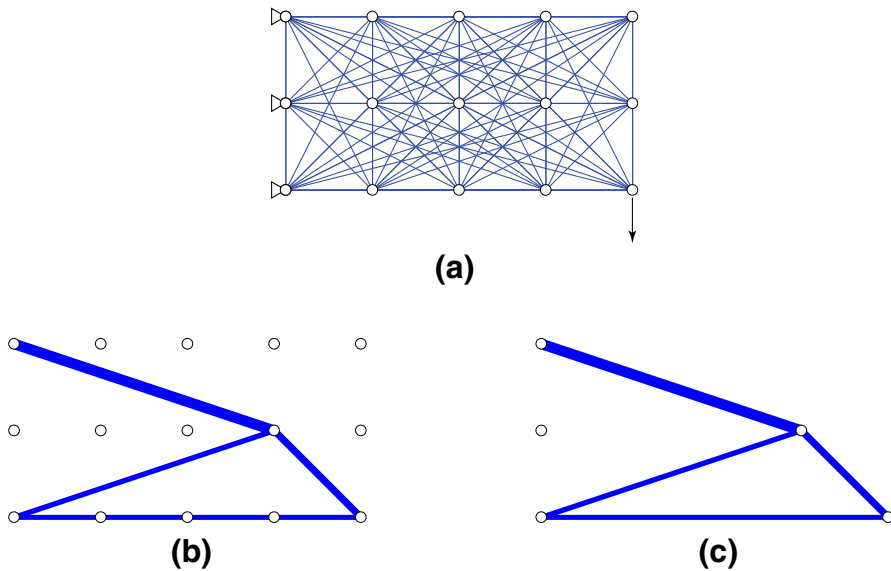


Fig. 2 An example of truss topology optimization and hinge cancellation. **a** The problem setting; **b** the optimal solution; and **c** the final design after hinge cancellation

$$(Zx - z^k + v^k, t, 1) \in \mathcal{K}^{l+2},$$

problem (22) is an SOCP problem. We adopt a primal-dual interior-point method for solving this problem. Next, the second step of ADMM in (19) can be written as

$$z^{k+1} \in \Pi_G(Zx^{k+1} + v^k), \tag{23}$$

where Π_G is the projection onto G .⁴ We can compute (23) easily (Boyd et al. 2010, Chap. 9); for a point $z \in \mathbb{R}^l$, $\Pi_G(z)$ keeps the n largest magnitude components of z and zeros out the other components. In this way, each step of ADMM in (18), (19), and (20) can be carried out very easily.

5 On overlapping members

Unlike the conventional compliance minimization of a truss, overlapping members in a ground structure are not redundant for the optimization problem considered in this paper. This section explains the treatment of overlapping members.

We begin by reviewing that overlapping members in a ground structure is redundant for the conventional compliance minimization problem of a truss. For example, consider the ground structure shown in Fig. 2a. Here, any two nodes are connected by a member, but overlapping of members is avoided by removing the longer member when two members overlap. The leftmost nodes are pin-supported.

⁴ Since G is nonconvex, the projection of a point onto G is not necessarily unique.

The vertical external force is applied to the bottom rightmost node. Figure 2b shows the optimal solution of the compliance minimization problem, i.e., problem (11). This solution has four horizontal consecutive members that are connected by nodes supported only in the direction of those members. A sequence of such members is sometimes called a *chain* (Achtziger 1999). In this example, without changing the objective value, we can remove three intermediate nodes to replace the chain with a single longer member. This procedure is called *hinge cancellation* (Achtziger 1999; Rozvany 1996). As a result of hinge cancellation, we obtain the final truss design shown in Fig. 2c. Thus, longer overlapping members, like the horizontal member in Fig. 2c, are unnecessary to a ground structure. In contrast, when we consider a constraint on a number of nodes, the optimal solution depends on the existence of overlapping members in a ground structure. For example, the truss in Fig. 2b has five free nodes, while the one in Fig. 2c has two free nodes. Thus, hinge cancellation can possibly change the feasibility of the cardinality constraint and, hence, overlapping members in a ground structure are not redundant.⁵

When we consider a ground structure with some overlapping members, the existence of overlapping members in an obtained solution is not allowed from a practical point of view. The method proposed in Sect. 4 does not consider explicitly the constraint prohibiting presence of overlapping members. Nevertheless, in practice, a solution obtained by the proposed method often has no overlapping members, as illustrated through numerical experiments in Sect. 6.

Within the framework of MISOCP, we can explicitly incorporate the constraints prohibiting the presence of mutually overlapping members in a truss design. To do this, besides $s \in \{0, 1\}^l$ in Sect. 3.2, we use additional binary variables $t \in \{0, 1\}^m$ to indicate whether each member vanishes or exists. Namely, $t_i = 0$ means that member i is removed, while $t_i = 1$ means that member i exists. The relationship between t_i and x_i is given by

$$0 \leq x_i \leq M t_i$$

where $M > 0$ is a sufficiently large constant. Recall that $I(j)$ denotes the set of indices of members connected to node j ; see Sect. 4. The relationship between t_i ($i \in I(j)$) and s_j is given by

$$t_i \leq s_j, \quad \forall i \in I(j).$$

Let D denote the set of pairs of indices of members that mutually overlap. Namely, $(i_1, i_2) \in D$ means that member i_1 and member i_2 cannot exist simultaneously. This constraint is written as

$$t_{i_1} + t_{i_2} \leq 1, \quad \forall (i_1, i_2) \in D.$$

The upshot is that the truss topology optimization problem can be formulated as the following MISOCP problem:

⁵ Such non-redundancy of overlapping members is also known for truss topology optimization considering, e.g., the self-weight load (Bendsøe et al. 1994; Kanno and Yamada 2017) and the member buckling constraints (Mela 2014; Guo et al. 2005).

$$\text{Minimize } \sum_{i=1}^m w_i \quad (24a)$$

$$\text{subject to } w_i + x_i \geq \left\| \begin{bmatrix} w_i - x_i \\ 2\sqrt{c_i/Eq_i} \end{bmatrix} \right\|, \quad i = 1, \dots, m, \quad (24b)$$

$$\sum_{i=1}^m q_i \mathbf{b}_i = \mathbf{p}, \quad (24c)$$

$$\mathbf{c}^\top \mathbf{x} \leq V, \quad (24d)$$

$$\mathbf{x} \leq M\mathbf{t}, \quad (24e)$$

$$t_i \leq s_j \quad (\forall i \in I(j)), \quad j = 1, \dots, l, \quad (24f)$$

$$\mathbf{s} \geq \mathbf{0}, \quad (24g)$$

$$\sum_{j=1}^l s_j \leq n, \quad (24h)$$

$$t_{i_1} + t_{i_2} \leq 1, \quad \forall (i_1, i_2) \in D, \quad (24i)$$

$$\mathbf{t} \in \{0, 1\}^m. \quad (24j)$$

It is worth noting that the binary constraints on s_1, \dots, s_k can be omitted.

6 Numerical experiments

In this section, we report on numerical experiments on the method presented in Sect. 4. In Sect. 6.1, we describe implementation details of the algorithm and the problem settings of the numerical experiments. The computational results of the proposed ADMM approach, together with a comparison with the MISOCP approach, are presented in Sects. 6.2, 6.3, and 6.4. Empirical evidences of our stopping criterion and selection of initial points are presented in Sects. 6.5 and 6.6, respectively. Section 6.7 presents an application of the proposed method to robust truss optimization, which is recast as a mixed-integer semidefinite programming problem.

6.1 Implementation and problem settings

At each iteration of the proposed method, we solved problem (22) by using CVX ver. 2.1, a MATLAB package for specifying and solving convex optimization problems (Grant and Boyd 2008, 2016). As a solver, we used SDPT3 ver. 4.0

(Tütüncü et al. 2003) on MATLAB ver. 9.1.0. The `cvx_precision` of CVX is set to `best`, which means that the solver continues as long as it can make progress (Grant and Boyd 2016). For comparison, we solved the MISOCP problem in (15) with a global optimization approach. The value of M in constraint (15f) is set to 1.0×10^5 in m.⁶ We used PICOS ver. 1.1.2, a Python interface to diverse optimization solvers (Sagnol 2017). MOSEK ver. 8.0.1 (Andersen et al. 2003) was used as the solver. Computation was carried out on two 3.2 GHz Intel Xeon E5-2667 v4 processors with 256 GB RAM.

In practice, we slightly modify the original version of ADMM introduced in Sect. 2 so that the penalty parameter in the augmented Lagrangian is gradually increased. Specifically, ρ in subproblem (22) is given by

$$\rho_{k+1} := \min\{\mu\rho_k, \rho_{\max}\},$$

where $\mu (> 1)$ and $\rho_{\max} (> \rho_0)$ are constants. In the following, we set $\mu = 1.5$, $\rho_0 = 1$, and $\rho_{\max} = 10^6$. Define $J_0^k \subseteq \{1, \dots, l\}$ by

$$J_0^k = \{j \in \{1, \dots, l\} \mid z_j^k \leq \epsilon\},$$

where we set $\epsilon = 0.1 \text{ mm}^2$. We terminate the ADMM when

$$l - |J_0^k| \leq n$$

is satisfied. Then we solve problem (11) with the additional constraints

$$\sum_{i \in I(j)} x_i = 0, \quad \forall j \in J_0^k$$

to generate the final output. As for the initial point for the ADMM, we examine two cases:

- Initial point (A): $z^0 := Zx^0$ and $v^0 := \mathbf{0}$, where x^0 is an optimal solution of problem (11).
- Initial point (B): $z^0 := Zx^0$ and $v^0 := \mathbf{0}$ with $x^0 := (V/c^\top \mathbf{1})\mathbf{1}$.

It should be clear that only z^0 and v^0 are used as input data of the ADMM; x^0 is not required as input.

Consider the problem setting shown in Fig. 3. The nodes are aligned on a $1 \text{ m} \times 1 \text{ m}$ grid. We vary the values of N_X and N_Y to generate problem instances with diverse sizes. The number of free nodes in this ground structure is $N_X(N_Y + 1)$. The members in a ground structure are generated as follows: We first consider all possible members such that any two nodes are connected by a member. Then, we remove members that are longer than a specified value, 5 m in Sects. 6.2 and 6.3 and 7 m in Sect. 6.4. It is worth noting that the ground structure retains overlapping members.

⁶ Through our preliminary numerical experiments it was found that the computational cost required by MOSEK does not change drastically depending on the value of M .

Fig. 3 The problem setting for numerical experiments with $(N_X, N_Y) = (5, 2)$

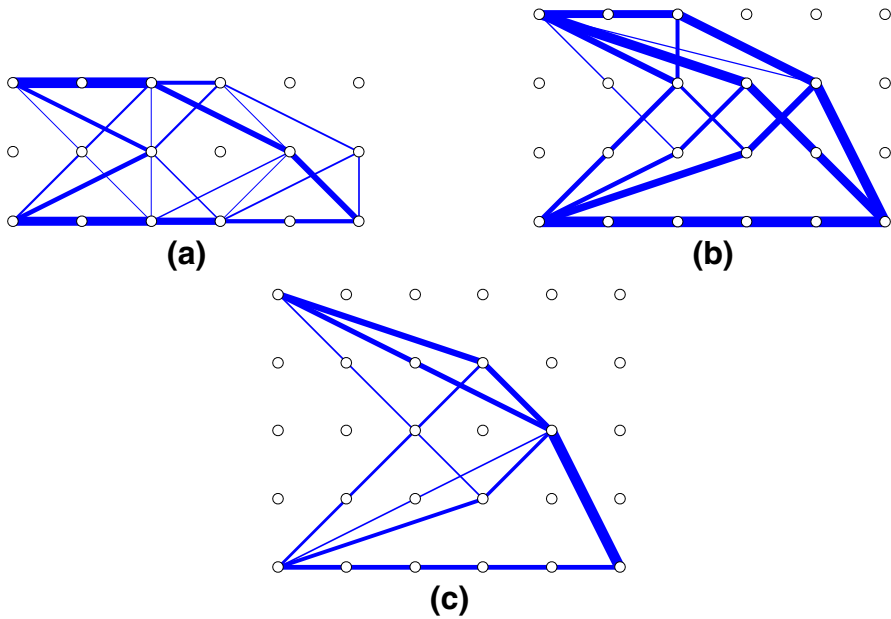
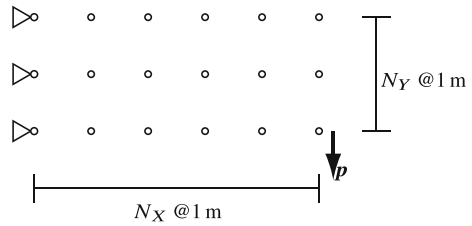


Fig. 4 Example (I). The optimal solutions of the compliance minimization problem (without the cardinality constraint). **a** $(N_X, N_Y) = (5, 2)$; **b** $(N_X, N_Y) = (5, 3)$; and **c** $(N_X, N_Y) = (5, 4)$

In the following examples, the Young modulus is $E = 20\text{ GPa}$, and the specified upper bound for the structural volume is $V = 2N_X N_Y \times 10^5\text{ mm}^3$. As for p , the external vertical force of 100 kN is applied to the bottom rightmost node. Let n be the upper bound on the number of free nodes. In other words, the number of supports is not restricted in the following examples, and l in the previous sections denotes the number of free nodes of a ground structure.

6.2 Example (I)

In this section, we set the upper bound for the existing free nodes to $n = 4$. As for problem instances, we consider $(N_X, N_Y) = (5, 2)$, $(5, 3)$, and $(5, 4)$ in Fig. 3. Figure 4 shows the optimal solutions⁷ of the conventional compliance minimization without the constraint on the number of nodes, i.e., problem (11), where the width of

⁷ To obtain these solutions, we used the ground structures without overlapping members.

Table 1 Characteristics of the problem instances for the numerical experiments

(N_X, N_Y)	m	d	\hat{w} (J)	#Free nodes
(5,2)	147	30	12100.00	9
(5,3)	264	40	5007.41	7
(5,4)	411	50	2812.50	5
(8,2)	273	48	34515.63	15
(9,2)	315	54	45125.00	8
(8,4)	750	80	6937.81	10
(9,4)	863	90	8900.28	10
(8,6)	1296	112	3080.39	10
(9,6)	1489	126	3847.25	12

each member is proportional to its cross-sectional area. Table 1 reports the optimal values, denoted \hat{w} , the number of members (m), and the number of degrees of freedom of the nodal displacements (d). As mentioned in Sect. 5, the intermediate nodes on a chain in Fig. 4 can be removed without changing the objective value. After this hinge cancellation procedure, the numbers of free nodes in Fig. 4a–c become 9, 7, and 5, respectively, as listed in Table 1.

Figure 5 shows the solutions obtained by the proposed ADMM for the problem with the limited number of free nodes. In Fig. 5b we see that the number of free nodes is 3 ($< n$). It should be clear that \mathbf{x}^0 used to generate initial point (A) for the ADMM is in general different from the one in Fig. 4, because \mathbf{x}^0 is computed from the ground structure involving the overlapping members. Indeed, \mathbf{x}^0 for $(N_X, N_Y) = (5, 2)$, $(5, 3)$, and $(5, 4)$ have 12, 15, and 13 free nodes, respectively. Thus, the number of nodes is decreased successfully by the proposed method. It is observed that the solutions in Fig. 4a, b have too many members from a practical point of view. In contrast, we can see in Fig. 5a, b that the number of members is decreased as a result of optimization with the limitation of the number of nodes. The computational results of the ADMM are listed in Table 2, where w^* is the objective value of the obtained solution, “#iter.” is the number of iterations required before convergence, and “time” is the computational time. As mentioned before, we examine two different values, denoted (A) and (B), for \mathbf{z}^0 and \mathbf{v}^0 . The one which yields the better objective value is indicated by “*.” It is observed in Table 2 that, for every instance, the objective value of the solution obtained by the ADMM approach is identical to the optimal value of the problem without the limitation of the number of nodes (i.e., problem (11)). Since problem (11) can be regarded as a relaxation problem, the solutions obtained by the proposed ADMM are globally optimal. This also illustrates that, in general, the compliance minimization of a truss has more than one optimal solution, and the optimal solutions may have different numbers of nodes.

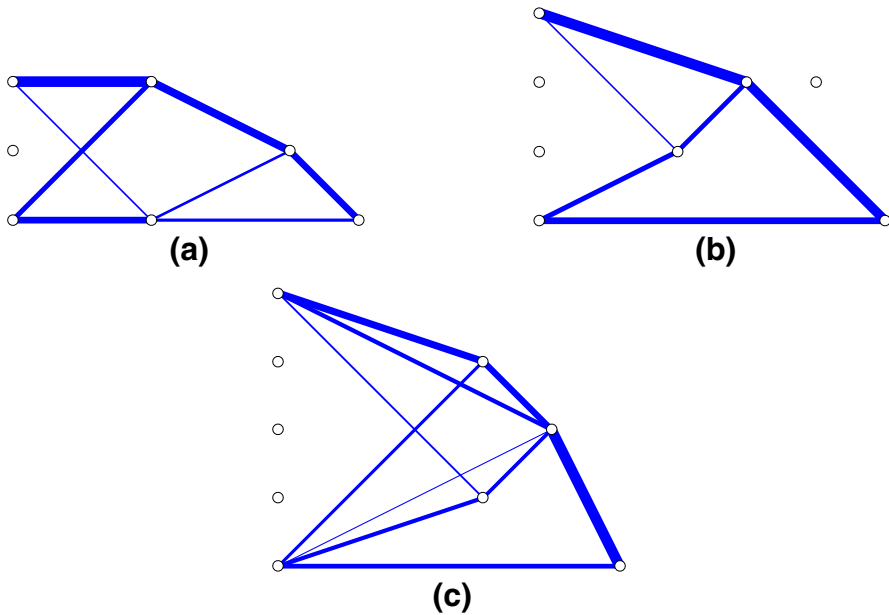


Fig. 5 Example (I). The solutions obtained by the proposed method for the compliance minimization problem with the cardinality constraint ($n = 4$). **a** $(N_X, N_Y) = (5, 2)$; **b** $(N_X, N_Y) = (5, 3)$; and **c** $(N_X, N_Y) = (5, 4)$

Table 2 Computational results of example (I)

(N_X, N_Y)	ADMM					MISOCP	
	Init. sol.	w^* (J)	w^*/\hat{w}	#iter	Time (s)	\bar{w} (J)	Time (s)
(5,2)	* (A)	12100.00	1.000	5	3.6	12100.00	0.72
	* (B)	12100.00	1.000	3	2.2		
(5,3)	* (A)	5007.41	1.000	5	3.2	5007.41	1.69
	(B)	5052.45	–	3	2.0		
(5,4)	* (A)	2812.50	1.000	2	2.6	2812.50	0.52
	* (B)	2812.50	1.000	3	3.8		

For comparison, we also solved the MISOCP problem (15) with a global optimization approach. Table 2 lists the obtained results,⁸ where \bar{w} is the objective value. The solutions obtained by the MISOCP solver are identical to the ones obtained by the ADMM approach.

⁸ It should be clear that no initial point was assigned for the MISOCP approach, although in Table 2, for convenience of presentation, the results of MISOCP are placed in the rows concerning the results of the ADMM with initial point (A).

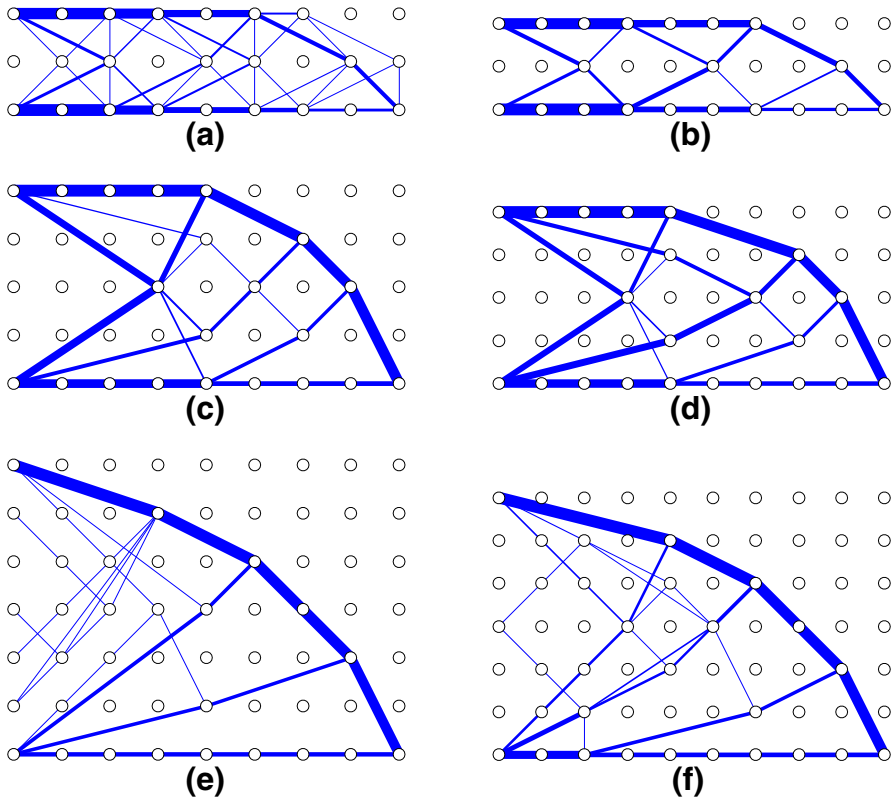


Fig. 6 Example (II). The optimal solutions of the compliance minimization (without the cardinality constraint). **a** $(N_X, N_Y) = (8, 2)$; **b** $(N_X, N_Y) = (9, 2)$; **c** $(N_X, N_Y) = (8, 4)$; **d** $(N_X, N_Y) = (9, 4)$; **e** $(N_X, N_Y) = (8, 6)$; and **f** $(N_X, N_Y) = (9, 6)$

6.3 Example (II)

As for instances with larger sizes, consider $N_X, N_Y = (8, 2), (9, 2), (8, 4), (9, 4), (8, 6),$ and $(9, 6)$. In this section, we set the upper bound for the number of free nodes to $n = 5$.

6.3.1 Results

Figure 6 collects the optimal solutions without limiting the number of nodes. The number of free nodes after applying hinge cancellation is reported in Table 1. Figure 7 shows the solutions obtained by the ADMM approach. The number of free nodes in Fig. 7e is 4 ($< n$). Two nodes can be removed from the solution in Fig. 7a, which results in a truss design with three free nodes. It is observed in Figs. 6 and 7 that the limitation of the number of nodes often yields a solution with fewer members. Also, very thin members are observed in Fig. 6, while such thin members do not appear in Fig. 7. These two features of the solutions in Fig. 7 are considered

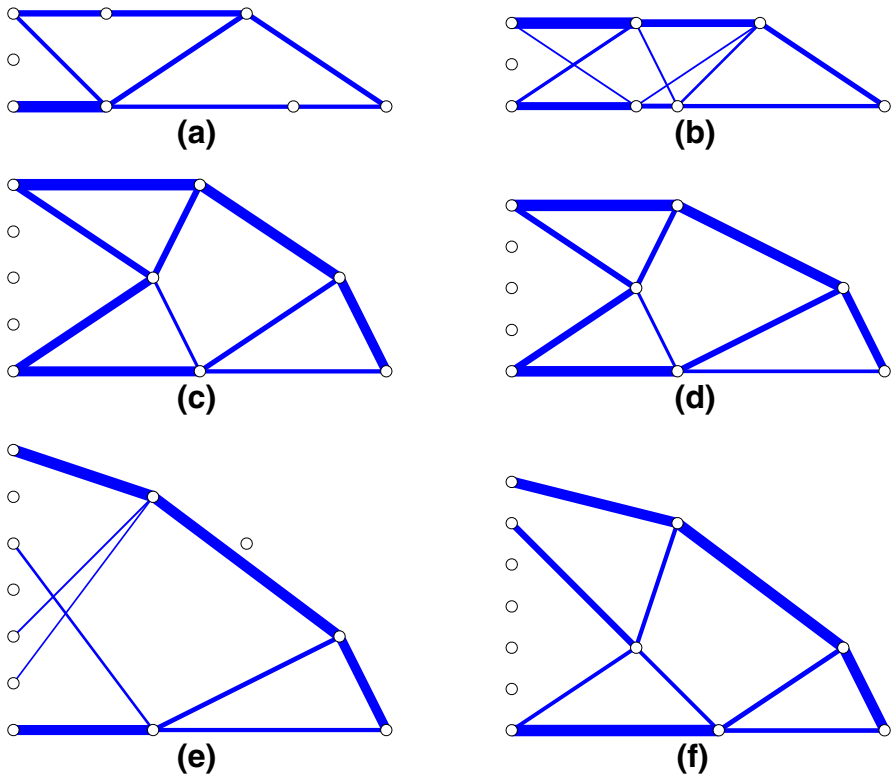


Fig. 7 Example (II). The solutions obtained by the ADMM for the compliance minimization problem with the cardinality constraint ($n = 5$). **a** $(N_X, N_Y) = (8, 2)$; **b** $(N_X, N_Y) = (9, 2)$; **c** $(N_X, N_Y) = (8, 4)$; **d** $(N_X, N_Y) = (9, 4)$; **e** $(N_X, N_Y) = (8, 6)$; and **f** $(N_X, N_Y) = (9, 6)$

practically preferable. The initial design \mathbf{x}^0 used for generating initial points (A) for the ADMM to solve $(N_X, N_Y) = (8, 2), (9, 2), \dots, (9, 6)$ have 20, 18, 19, 20, 24, and 26 free nodes, respectively.

It is observed in Table 3 that the ADMM terminates after at most 20 iterations. The increase of the objective value over the optimal value of the problem without the cardinality constraint is quite small, i.e., the increase by at most about 10%. Particularly, for $(N_X, N_Y) = (8, 4)$ and $(9, 4)$ we have only about a 1% increase. Thus, it is often the case that the number of nodes can be reduced at the expense of only a small increase of the compliance.

The computational results of the MISOCQP approach are listed in Table 3. Figure 8 collects the obtained solutions. For $(N_X, N_Y) = (8, 4)$, the solution obtained by MISOCQP is identical to the one obtained by the ADMM; i.e., the ADMM found a global optimal solution. For $(N_X, N_Y) = (9, 4)$, it is observed in Table 3 that the objective values obtained by the two methods are almost the same, but the two solutions are slightly different as seen in Fig. 7d and 8d. The largest value of w^*/\bar{w} is 1.072 in the case of $(N_X, N_Y) = (8, 2)$. It is also worth noting that, for $(N_X, N_Y) = (8, 6)$ and $(9, 6)$, although the global optimal solutions in Fig. 8e, f involve very thin

Table 3 Computational results of example (II)

(N_X, N_Y)	ADMM					MISOCP		
	Init. sol.	w^* (J)	w^*/\hat{w}	w^*/\bar{w}	#iter	Time (s)	\bar{w} (J)	Time (s)
(8,2)	* (A)	37515.63	1.087	1.072	11	9.0	35006.93	13.42
	* (B)	37515.63	1.087	1.072	13	9.8		
(9,2)	* (A)	50000.00	1.108	1.070	20	18.8	46722.20	19.92
	* (B)	50000.00	1.108	1.070	18	16.5		
(8,4)	* (A)	7031.25	1.013	1.000	11	46.4	7031.25	4.30
	* (B)	7031.25	1.013	1.000	6	24.5		
(9,4)	(A)	9167.44	–	–	13	77.2	8999.91	12.20
	* (B)	9000.00	1.011	1.000	8	46.7		
(8,6)	* (A)	3287.84	1.067	1.038	10	80.3	3168.98	76.22
	(B)	3440.05	–	–	7	54.1		
(9,6)	(A)	4353.91	–	–	9	96.5	3983.34	112.08
	* (B)	4221.65	1.097	1.060	12	122.1		

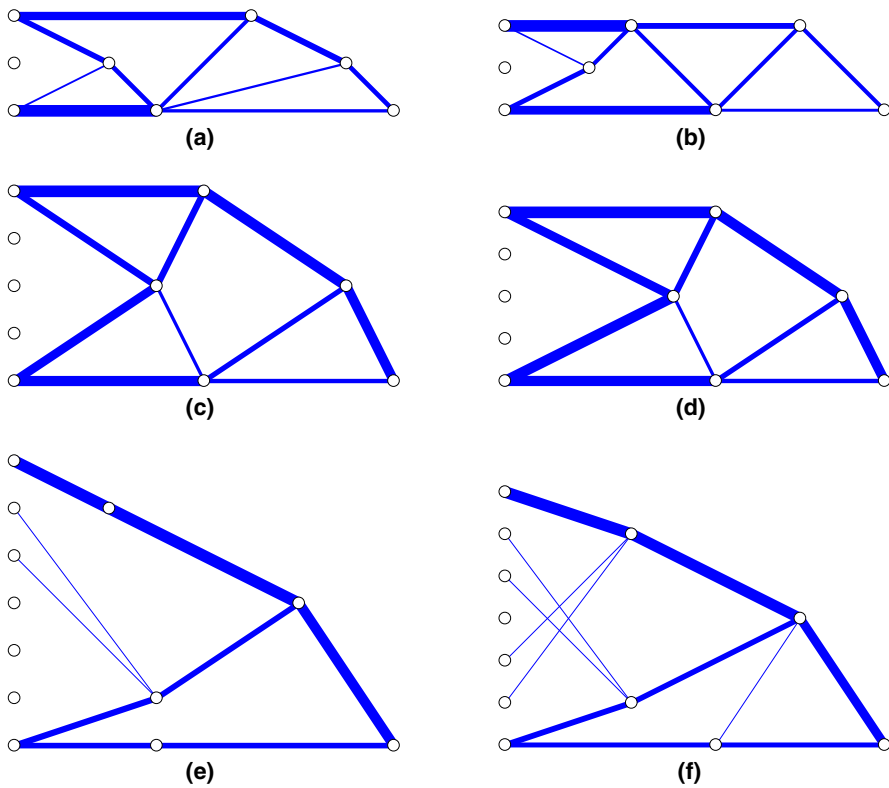


Fig. 8 Example (II). The optimal solutions obtained from the MISOCP formulation for the compliance minimization problem with the cardinality constraint ($n = 5$). **a** $(N_X, N_Y) = (8, 2)$; **b** $(N_X, N_Y) = (9, 2)$; **c** $(N_X, N_Y) = (8, 4)$; **d** $(N_X, N_Y) = (9, 4)$; **e** $(N_X, N_Y) = (8, 6)$; and **f** $(N_X, N_Y) = (9, 6)$

members, the solutions obtained by the ADMM shown in Fig. 7e, f do not have such a thin member.

It is observed from Table 3 that the proposed ADMM often converges more quickly than the MISOCP solver; exceptions are $(N_X, N_Y) = (8, 4)$ and $(9, 4)$. The computational time required by the MISOCP solver varies drastically depending on the problem instances. In contrast, the number of iterations required by the ADMM is almost independent of the problem instances. Since the computation time required for solving the SOCP subproblem of the ADMM depends on the problem size, it is possible to roughly estimate the total computational cost of the ADMM from the problem size. This might be considered one of advantages of ADMM over MISOCP.

As mentioned in Sect. 5, the proposed method does not incorporate a constraint prohibiting overlapping members. Nevertheless, for the problems with limitation of the number of nodes, all the solutions obtained in Sect. 6 do not involve overlapping members.

6.3.2 MISOCP with slenderness constraints

Constraints preventing the presence of very thin members observed in Fig. 8e, f can be handled within the framework of mixed-integer programming (MIP) (Kanno and Yamada 2017). Recall problem (24) in Sect. 5, where t_i is a binary variable indicating whether member i exists or vanishes. Let $x_{\min} > 0$ denote the specified lower bound for the member cross-sectional area. The constraint avoiding the existence of too-thin members, which we refer to as *slenderness constraints*, can be formulated as

$$x_{\min} \mathbf{t} \leq \mathbf{x} \leq M \mathbf{t}. \quad (25)$$

In problem (24), we replace constraint (24e) with (25). The constraint avoiding the presence of overlapping members, (24i), is not considered. We solve this MISOCP problem for the instances $(N_X, N_Y) = (8, 6)$ and $(9, 6)$ with $x_{\min} = 200 \text{ mm}^2$. The obtained solutions are shown in Fig. 9. Both solutions have parallel consecutive

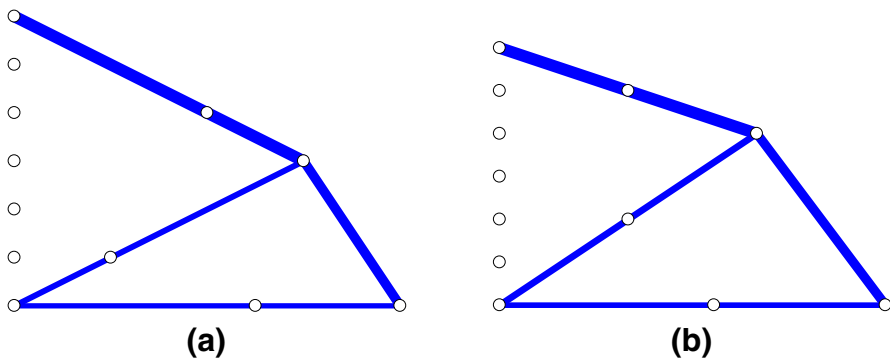


Fig. 9 The optimal solutions of example (II) with the constraints avoiding the presence of thin members. **a** $(N_X, N_Y) = (8, 6)$; and **b** $(N_X, N_Y) = (9, 6)$

members that are connected by nodes supported only in the direction of those members. The intermediate nodes can be removed without changing the optimal value. Hence, the number of free nodes of these solutions is essentially two. The objective value of the solution for $(N_X, N_Y) = (8, 6)$ is 3168.97 J, which is slightly less than that for the case without the slenderness constraints in Table 3. This is due to computational error in computing the objective value via the finite element method. In the solution shown in Fig. 7e, i.e., in the solution obtained by the ADMM without the slenderness constraints, the cross-sectional area of the thinnest member is 75.4 mm^2 . Hence, this solution is not globally optimal under the slenderness constraint. The computational time required by MOSEK is 365.0 s. The objective value of the solution for $(N_X, N_Y) = (9, 6)$ is 4028.94 J. This is larger than that for the case without the slenderness constraints as expected, and is less than that of the solution obtained by the ADMM. Since the cross-sectional area of the thinnest member of the solution shown in Fig. 7f is 146.7 mm^2 , Hence, the solution shown in Fig. 7f is not globally optimal under the slenderness constraints. The computational time required by MOSEK to find the solution in Fig. 9b was 1676.0 s, which is much larger than the computational time of the ADMM.

6.4 Example (III)

Consider problem instances $(N_X, N_Y, n) = (12, 6, 6)$, $(13, 6, 6)$, and $(14, 6, 7)$. The maximum member length in a ground structure is set to 7 m.

Figure 10 shows the solutions obtained by the proposed ADMM approach. The ADMM terminates with a solution having n free nodes. One of these nodes vanishes in post-processing. The objective value as well as the computational cost is reported in Table 4.

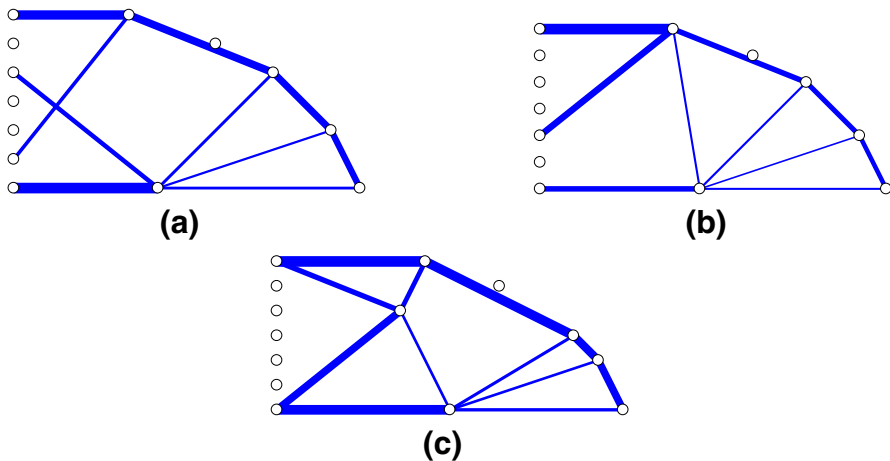


Fig. 10 Example (III). The solutions obtained by ADMM for the compliance minimization problem with the cardinality constraint. **a** $(N_X, N_Y, n) = (12, 6, 6)$; **b** $(N_X, N_Y, n) = (13, 6, 6)$; and **c** $(N_X, N_Y, n) = (14, 6, 7)$,

Table 4 Computational results of example (III)

(N_X, N_Y, n)	ADMM					MISOCP	
	Init. sol.	w^* (J)	w^*/\bar{w}	#iter	Time (s)	\bar{w} (J)	Time (s)
(12, 6, 6)	* (A)	7817.72	1.115	10	531.7	7012.95	3492.34
	(B)	9527.47		10	516.7		
(13,6,6)	* (A)	10142.88	1.228	10	642.6	8258.85	14675.17
	(B)	13631.69		11	793.7		
(14,6,7)	* (A)	9720.09	1.018	9	724.2	9550.97	14657.54
	(B)	14928.49		14	1148.6		

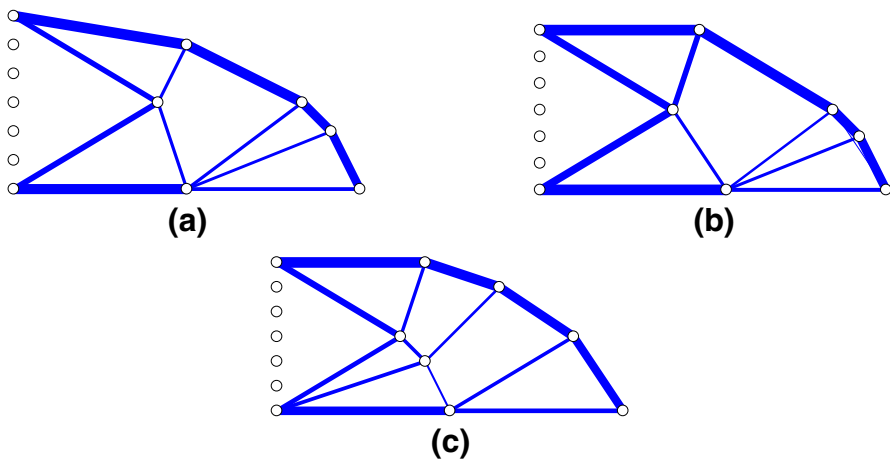
**Fig. 11** Example (III). The optimal solutions obtained by MISOCP for the compliance minimization problem with the cardinality constraint. **a** $(N_X, N_Y, n) = (12, 6, 6)$; **b** $(N_X, N_Y, n) = (13, 6, 6)$; and **c** $(N_X, N_Y, n) = (14, 6, 7)$

Figure 11 collects the optimal solutions found by the MISOCP approach. These solutions use exactly n free nodes. It is observed from Table 4 that the objective value obtained by the ADMM for the largest instance, $(N_X, N_Y, n) = (14, 6, 7)$, is very close to the optimal value. In contrast, for $(N_X, N_Y, n) = (13, 6, 6)$ the objective value obtained by the ADMM is more than 20% larger than the optimal value. However, the computational cost of the ADMM is much less than the MISOCP approach (which requires more than four hours). Thus, the quality of the solution obtained by the ADMM approach can possibly be very good, although in general it depends on problem instances. As the problem size increases, the computational cost of the ADMM approach becomes much smaller compared with the MISOCP approach.

Table 5 The computational results when the ADMM is run until convergence to evaluate effectiveness of the heuristic stopping criterion

(N_X, N_Y, n)	Init. sol.	\tilde{K}	\tilde{K}_p	K^*
(8,2,5)	(A)	35	18	11
(8,2,5)	(B)	35	18	13
(9,2,5)	(A)	100 [†]	77 [†]	20
(9,2,5)	(B)	81	58	18
(8,4,5)	(A)	27	15	11
(8,4,5)	(B)	21	15	6
(9,4,5)	(A)	29	15	13
(9,4,5)	(B)	25	15	8
(8,6,5)	(A)	26	12	10
(8,6,5)	(B)	21	15	7
(9,6,5)	(A)	24	16	9
(9,6,5)	(B)	36	25	12

6.5 On heuristic for stopping ADMM

As mentioned in Sect. 6.1, we use a heuristic criterion for stopping the ADMM. Namely, we stop the ADMM when the cardinality constraint is satisfied with ϵ tolerance. Then, as post-processing, we solve the compliance minimization problem, (11), specifying the set of vanishing nodes. This section presents some empirical justification for this procedure. Namely, it is illustrated through numerical experiments that with this heuristic procedure the number of subproblems to be solved is drastically reduced, without missing out solutions having better objective values. We use the problem instances in Sect. 6.3.

We performed the following experiment. The ADMM is run until it terminates with a small tolerance, namely, $\|\mathbf{x}^{k+1} - \mathbf{x}^k\| \leq 10^{-1}$ (in mm²) is satisfied. This requires many more iterations than the procedure described above. In the iteration history, we select every iterate that satisfies the cardinality constraint, i.e., that satisfies $l - |J_0^k| \leq n$. For every selected iterate, we run the post-processing, i.e., we solve problem (11), specifying the set of vanishing nodes.

The computational results are listed in Table 5, where \tilde{K} and \tilde{K}_p are the number of iterations required before convergence and the number of iterates that approximately satisfy the cardinality constraint, respectively. For all the \tilde{K}_p iterates, the post-processing yields the same solution as the one reported in Table 3. In the case $(N_X, N_Y, n) = (9, 2, 5)$ with initial point (A), the ADMM does not converge after 100 iterations. In this iteration history, there exist 77 iterates satisfying the cardinality constraint approximately. From all of them, the post-processing generates the solution reported in Table 3. In a nutshell, the set of vanishing nodes does not change, even if the ADMM iterations are continued after the iterate at which our heuristic stopping criterion is satisfied.

For ease of comparison, the number of iterations reported in Table 3 is listed again as K^* in Table 5. Our stopping criterion reduces the number of iterations from \tilde{K} to K^* , without changing the final output. It is worth noting that the solutions

found between the $(K^* + 1)$ th iteration and the \tilde{K} th iteration do not necessarily satisfy the cardinality constraint within an ϵ -tolerance.

6.6 On choice of initial points

Since we apply an ADMM to a nonconvex problem, the obtained solution may depend on the choice of initial points. In Sect. 6.1, we suggested using two initial points, (A) and (B), and adopt the better solution as the final output. In this section, we perform comparison with the results obtained by using randomly generated initial points to empirically justify our selection. As for two representative instances for which initial points (A) and (B) lead to different solutions, we use $(N_X, N_Y, n) = (5, 3, 4)$ in Sect. 6.1 and $(N_X, N_Y, n) = (9, 4, 5)$ in Sect. 6.2 in the following numerical experiments.

As for randomly generated initial points, we examine two cases:

- Initial point (C): $\mathbf{x}^0 := (V/c^\top \xi)\xi$, $\mathbf{z}^0 := Z\mathbf{x}^0$, and $\mathbf{v}^0 := Z\mathbf{x}^0 - \mathbf{z}^0 = \mathbf{0}$, where $\xi \in \mathbb{R}^m$ is a random vector with entries independently drawn from $\mathcal{U}(0, 1)$.
- Initial point (D): $\mathbf{x}^0 := (V/c^\top \xi)\xi$, $\mathbf{z}^0 := Z\mathbf{x}^0$, and $\mathbf{v}^0 := \max\{z_1^0, \dots, z_l^0\}\zeta$, where $\xi \in \mathbb{R}^m$ and $\zeta \in \mathbb{R}^l$ are random vectors with entries independently drawn from $\mathcal{U}(0, 1)$.

We generate 100 samples of each of these initial points, and run our ADMM approach from every sample. Table 6 reports the minimum value, maximum value, mean, and variance of the objective value.

For $(N_X, N_Y, n) = (5, 3, 4)$ with initial point (C), in all the cases the ADMM converges to the same solution. This solution is the one obtained by using initial point (B), as shown in Table 2. Therefore, using initial point (A) yielded a better solution (which is globally optimal) than using 100 samples of (C). In contrast, when initial point (D) was adopted, the global optimal solution is obtained from 7 sampled initial points, among 100 trials. From the other 93 samples, the ADMM converges to the solution obtained with initial point (B). The mean and the variance of the objective value are listed in Table 6. In this manner, it is demonstrated that the global optimal solution, easily obtained by carrying out our ADMM procedure with initial point (A), is rarely obtained from randomly generated initial points.

For $(N_X, N_Y, n) = (9, 4, 5)$ with initial point (C), the best solution is the same as the one obtained from initial point (B) in Table 3. This is not globally optimal. Among 100 trials, 67 sampled initial points yield this solution. In contrast, the objective value of the worst solution is larger than the one obtained from initial

Table 6 Computational results of ADMM from randomly generated initial points

(N_X, N_Y, n)	Init. sol.	Min. (J)	Max. (J)	Mean (J)	Var. (J ²)
(5,3,4)	(C)	5052.45	5052.45	5052.45	0.00
(5,3,4)	(D)	5007.41	5052.45	5049.74	115.56
(9,4,5)	(C)	9000.00	9143.42	9039.78	3344.71
(9,4,5)	(D)	9000.00	9768.04	9076.47	13916.47

Table 7 Computational results of the robust optimization problem with the cardinality constraint

(N_X, N_Y, n)	ADMM					MISDP		
	Init. sol.	w^* (J)	w^*/\bar{w}	#iter	Time (s)	\bar{w} (J)	#iter	Time (s)
(5,2,4)	* (A)	12156.25	1.000	3	7.8	12156.25	22	11.8
	* (B)	12156.25	1.000	3	6.7			
(5,3,4)	* (A)	5044.91	1.000	5	6.6	5044.91	9	7.6
	(B)	5089.95	–	3	4.7			
(5,4,4)	* (A)	2840.63	1.000	2	6.7	2840.63	5	10.9
	* (B)	2840.63	1.000	3	7.4			
(8,2,5)	* (A)	37605.63	1.071	11	16.5	35096.94	238	293.1
	* (B)	37605.63	1.071	13	18.2			
(9,2,5)	* (A)	50101.25	1.070	20	32.0	46823.47	1484	2396.2
	* (B)	50101.25	1.070	14	20.6			
(8,4,5)	* (A)	7076.25	1.000	11	67.0	7076.25	37	334.1
	* (B)	7076.25	1.000	6	34.3			
(9,4,5)	(A)	9218.06	–	13	110.6			
	* (B)	9050.63	1.000	6	47.0	9050.63	69	1026.2
(8,6,5)	* (A)	3317.84	1.037	10	118.8	3198.98	573	2192.7
	(B)	26377.74	–	19	358.5			
(9,6,5)	(A)	4387.66	–	9	148.6			
	* (B)	4255.40	‡	12	181.9	4387.66 [‡]	300 [‡]	16468.0 [‡]

point (A). By using initial point (D), the variance of the objective value increased, but the global optimal solution was not obtained. Thus, the ADMM with randomly generated initial points could not find a solution better than the one obtained from initial point (B).

In short, for these two problem instances, using many randomly generated initial points does not yield a better solution. Therefore, using initial points (A) and (B) might be considered a reasonable selection.

6.7 Application to robust optimization against uncertainty in external load

The ADMM approach presented in this paper can easily be extended to the case in which the external load possesses uncertainty. The set of nodes at which the external forces can possibly be applied is supposed to be specified. Then we consider the robust optimization against the uncertainty, under the upper bound constraint on the number of nodes. In this section, we examine the efficiency of the ADMM applied to this problem, as an example of optimization problems that are not handled with current mainstream MIP solvers. The computation in this section was carried out on a 2.2 GHz Intel Core i5 processor with 8 GB RAM.

As a concrete instance, consider the problem setting shown in Fig. 3. The external force is applied at the bottom right node, but this time its direction and magnitude are assumed to be uncertain. Without loss of generality, let p_1 and p_2

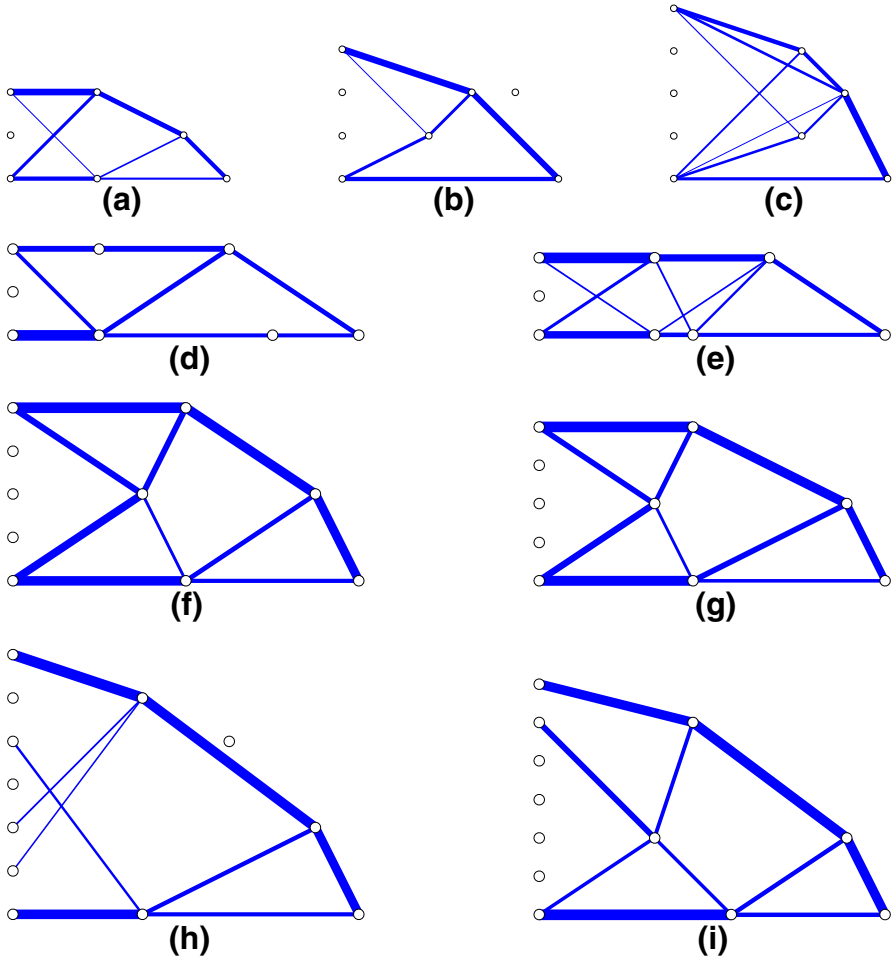


Fig. 12 The solutions obtained by the ADMM applied to the robust optimization problem. **a** $(N_X, N_Y, n) = (5, 2, 4)$; **b** $(N_X, N_Y, n) = (5, 3, 4)$; **c** $(N_X, N_Y, n) = (5, 4, 4)$; **d** $(N_X, N_Y, n) = (8, 2, 5)$; **e** $(N_X, N_Y, n) = (9, 2, 5)$; **f** $(N_X, N_Y, n) = (8, 4, 5)$; **g** $(N_X, N_Y, n) = (9, 4, 5)$; **h** $(N_X, N_Y, n) = (8, 6, 5)$; and **i** $(N_X, N_Y, n) = (9, 6, 5)$

denote the horizontal and vertical components, respectively, of this external force. The set of possible realizations of the external load is defined by

$$P = \{(p_1, p_2, 0, \dots, 0)^T \mid p_1 = p_1^0 \psi_1, p_2 = p_2^0 \psi_2, \|(\psi_1, \psi_2)\| \leq 1\},$$

where $p_1^0 = 30 \text{ kN}$ and $p_2^0 = 100 \text{ kN}$. With referring to (10), we see that the compliance in the worst case is given by

$$\hat{\pi}(x) = \sup\{2\mathbf{p}^T \mathbf{u} - \mathbf{u}^T K(x)\mathbf{u} \mid \mathbf{u} \in \mathbb{R}^d, \mathbf{p} \in P\}.$$

In the following, we consider the minimization problem of this function.

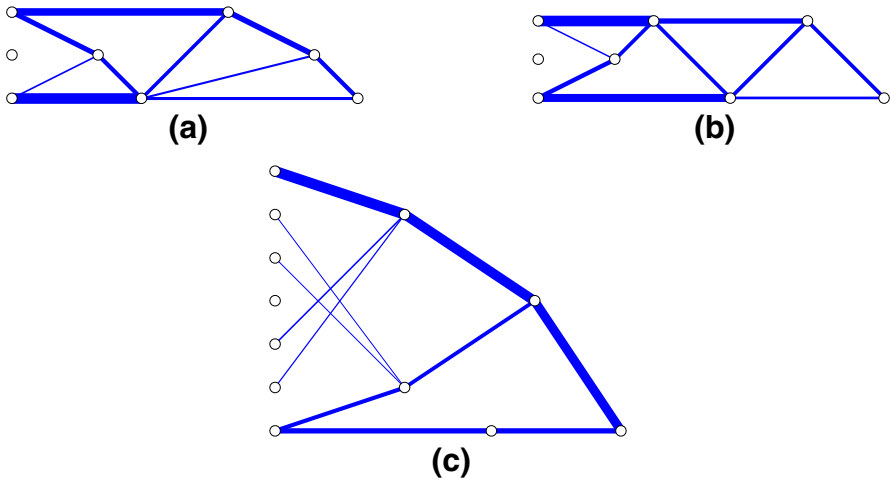


Fig. 13 The optimal solutions of the robust optimization problem obtained by YALMIP. **a** $(N_X, N_Y, n) = (8, 2, 5)$; **b** $(N_X, N_Y, n) = (9, 2, 5)$; and **c** $(N_X, N_Y, n) = (8, 6, 5)$

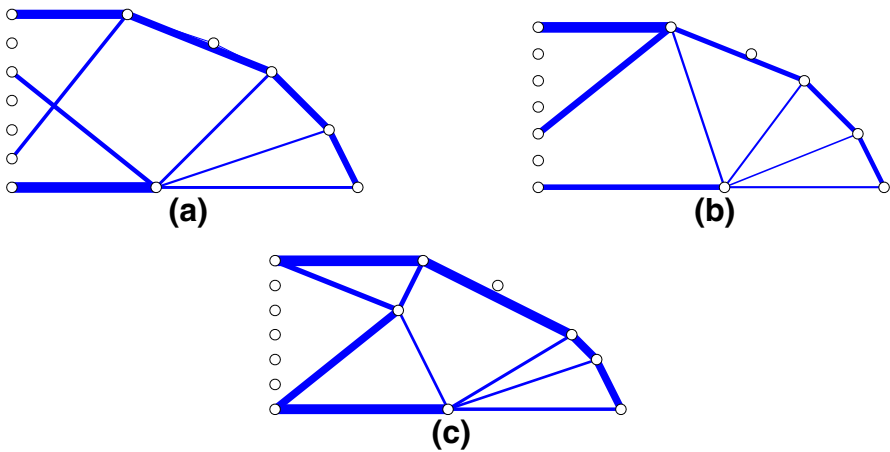


Fig. 14 The solutions obtained by ADMM for the large-scale robust optimization problems. **a** $(N_X, N_Y, n) = (12, 6, 6)$; **b** $(N_X, N_Y, n) = (13, 6, 6)$; **c** $(N_X, N_Y, n) = (14, 6, 7)$;

When the constraint on the number of nodes is not considered, it is known that this optimization problem can be recast as semidefinite programming (SDP) (Bent-Tal and Nemirovski 1997). Since the upper bound constraint on the number of nodes is treated as presented in Sect. 3.2, the optimization problem under this constraint can be recast as mixed-integer semidefinite programming (MISDP). For comparison, we solve this MISDP problem with YALMIP (Löfberg 2004), which finds a global optimal solution with a branch-and-bound method (Löfberg 2004). We used YALMIP with the default setting, in which SDP subproblems are solved with SeDuMi ver. 1.3 (Pólik 2005; Sturm 1999). Alternatively, consider the problem

Table 8 Computational results of the ADMM approach applied to the robust optimization problem with the cardinality constraint

(N_X, N_Y, n)	Init. sol.	w^* (J)	#iter	Time (s)
(12,6,6)	* (A)	7862.67	10	771.7
	(B)	9572.47	10	730.5
(13,6,6)	* (A)	10191.63	10	972.2
	(B)	13680.44	12	1108.4
(14,6,7)	* (A)	9772.59	9	1033.9
	(B)	14980.99	14	1614.1

obtained by replacing the objective function of (14) by $\hat{\pi}$. It is fairly straightforward to apply the ADMM in Sect. 4 to this optimization problem. The subproblem solved to update the variable \mathbf{x} at each iteration is formulated as SDP.

Table 7 reports the computational results. For five instances, $(N_X, N_Y, n) = (5, 2, 4)$, $(5, 3, 4)$, $(5, 4, 4)$, $(8, 4, 5)$, and $(9, 4, 5)$, the ADMM approach found the global optimal solutions. In all these cases, the computational cost of the ADMM is smaller than that of YALMIP. The difference of computational cost increases as the problem size increases. For $(N_X, N_Y, n) = (9, 6, 5)$, YALMIP did not terminate after 300 iterations. The best solution is the same as the one found by the ADMM with initial point (A), but a better solution was found by the ADMM with initial point (B). For three instances, $(N_X, N_Y, n) = (8, 2, 5)$, $(9, 2, 5)$, and $(8, 6, 5)$, the solutions found by the ADMM are not optimal. The difference between the obtained objective value and the optimal value is 7% or less, like in the cases in Sect. 6.3. The computational time required by YALMIP is more than 10 times (in some cases, more than 100 times) larger than that of the ADMM. Figure 12 collects the solutions obtained by the ADMM. The global optimal solutions that could not be obtained by the ADMM are shown in Fig. 13. The set of nodes in Fig. 12d is much different from that in Fig. 13a. The solution in Fig. 12e has only one node that is not included in the solution in Fig. 13b. Similarly, the difference between the solutions in Figs. 12h and 13c is the location of one node.

Figure 14 shows the solutions obtained by the ADMM for problem instances with larger sizes. The computational results are listed in Table 8. A global optimization method, YALMIP, cannot solve these problems within a reasonable amount of time.

7 Conclusions

In this paper, we have studied the compliance minimization of a truss with a limited number of nodes. It has been shown that this optimization problem can be formulated as a cardinality-constrained SOCP problem. We have proposed a simple and efficient heuristic based on an ADMM.

The problem considered in this paper can also be formulated as an MISOCP problem involving the so-called big-M. In the numerical experiments, we compared the proposed ADMM approach with a global optimization approach using this MISOCP formulation. For small-size problem instances, it has been confirmed that the ADMM can find a global optimal solution. For middle-size instances, the objective value of the solution obtained by the ADMM is often close to the optimal value. The number of iterations of the ADMM is almost the same for instances with different sizes. In contrast, the computational cost required by a standard MISOCP solver depends strongly on instances, even if the instances have similar sizes.

In the numerical experiments, it has also been illustrated that, for some problem instances, the compliance minimization problem of a truss has some different optimal solutions, and the number of nodes can be decreased without losing optimality. In most other cases, the number of nodes can be decreased at the expense of only a small increase of the compliance.

Acknowledgements The authors thank the referees and the editors for providing us with the detailed comments and suggestions, which were helpful in revision. The work of the first author is partially supported by JSPS KAKENHI 15KT0109 and 17K06633.

References

- Anjos MF, Lasserre JB (eds) (2012) Handbook on semidefinite, conic and polynomial optimization. Springer, New York
- Achtziger W (1999) Local stability of trusses in the context of topology optimization part I: exact modelling. *Struct Optim* 17:235–246
- Andersen ED, Roos C, Terlaky T (2003) On implementing a primal-dual interior-point method for conic quadratic optimization. *Math Program* 95:249–277
- Arastoo R, Bahavarnia M, Kothare MV, Motee N (2015) Output feedback controller sparsification via \mathcal{H}_∞ -approximation. *IFAC-PapersOnLine* 48:112–117
- Asadpoure A, Guest JK, Valdevit L (2015) Incorporating fabrication cost into topology optimization of discrete structures and lattices. *Struct Multidiscip Optim* 51:385–396
- Bendsøe MP, Ben-Tal A, Zowe J (1994) Optimization methods for truss geometry and topology design. *Struct Optim* 7:141–159
- Ben-Tal A, Nemirovski A (1997) Robust truss topology optimization via semidefinite programming. *SIAM J Optim* 7:991–1016
- Ben-Tal A, Nemirovski A (2001) Lectures on modern convex optimization: analysis, algorithms, and engineering applications. SIAM, Philadelphia
- Bertsimas D, Shioda R (2009) Algorithm for cardinality-constrained quadratic optimization. *Comput Optim Appl* 43:1–22
- Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2010) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found Trends Mach Learn* 3:1–122
- Bruckstein AM, Donoho DL, Elad M (2009) From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev* 51:34–81
- Burdakov OP, Kanzow C, Schwartz A (2016) Mathematical programs with cardinality constraints: reformulation by complementarity-type conditions and a regularization method. *SIAM J Optim* 26:397–425
- Candès EJ, Wakin MB, Boyd SP (2008) Enhancing sparsity by reweighted ℓ_1 minimization. *J Fourier Anal Appl* 14:877–905
- Chartrand R (2007) Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Process Lett* 14:707–710
- Chartrand R (2012) Nonconvex splitting for regularized low-rank + sparse decomposition. *IEEE Trans Signal Process* 60:5810–5819

- Chartrand R, Wohlberg B (2013) A nonconvex ADMM algorithm for group sparsity with sparse groups. In: 2013 IEEE international conference on acoustics, speech and signal processing, Vancouver, pp 6009–6013 (2013)
- Cui XT, Zheng XJ, Zhu SS, Sun XL (2013) Convex relaxations and MIQCQP reformulations for a class of cardinality-constrained portfolio selection problems. *J Glob Optim* 56:1409–1423
- Diamond S, Takapoui R, Boyd S (2018) A general system for heuristic minimization of convex functions over non-convex sets. *Optim Methods Softw* 33:165–193
- Gotoh J, Takeda A, Tono K (2018) DC formulations and algorithms for sparse optimization problems. *Math. Program.*, to appear. <https://doi.org/10.1007/s10107-017-1181-0>
- Grant M, Boyd S (2008) Graph implementations for nonsmooth convex programs. In: Blondel V, Boyd S, Kimura H (eds) Recent advances in learning and control (a tribute to M. Vidyasagar). Springer, Berlin, pp 95–110
- Grant M, Boyd S (2017) CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx/>. Accessed Jan 2017
- Guo X, Cheng GD, Olhoff N (2005) Optimum design of truss topology under buckling constraints. *Struct Multidiscip Optim* 30:169–180
- Gurobi Optimization, Inc.: Gurobi optimizer reference manual. <http://www.gurobi.com/>. Accessed Sept 2016
- He L, Gilbert M (2015) Rationalization of trusses generated via layout optimization. *Struct Multidiscip Optim* 52:677–694
- Hegemier GA, Prager W (1969) On Michell trusses. *Int J Mech Sci* 11:209–215
- Kanamori T, Takeda A (2014) Numerical study of learning algorithms on Stiefel manifold. *CMS* 11:319–340
- Kanno Y (2013) Damper placement optimization in a shear building model with discrete design variables: a mixed-integer second-order cone programming approach. *Earthq Eng Struct Dyn* 42:1657–1676
- Kanno Y (2016a) Global optimization of trusses with constraints on number of different cross-sections: a mixed-integer second-order cone programming approach. *Comput Optim Appl* 63:203–236
- Kanno Y (2016b) Mixed-integer second-order cone programming for global optimization of compliance of frame structure with discrete design variables. *Struct Multidiscip Optim* 54:301–316
- Kanno Y, Yamada H (2017) A note on truss topology optimization under self-weight load: mixed-integer second-order cone programming approach. *Struct Multidiscip Optim* 56:221–226
- Kirsch U (1989) Optimal topologies of structures. *Appl Mech Rev* 42:223–239
- Kočvara M (2017) Truss topology design by linear conic optimization. In: Terlaky T, Anjos MF, Ahmed S (eds) Advances and trends in optimization with engineering applications. SIAM, Philadelphia
- Le Thi HA, Dinh TP, Le HM, Vo XT (2015) DC approximation approaches for sparse optimization. *Eur J Oper Res* 244:26–46
- Lin F, Fardad M, Jovanović MR (2013) Design of optimal sparse feedback gains via the alternating direction method of multipliers. *IEEE Trans Autom Control* 58:2426–2431
- Löfberg J (2004) YALMIP: a toolbox for modeling and optimization in MATLAB. In: 2004 IEEE international conference on computer aided control system design, Taipei, pp 284–289 (2004)
- Magnússon S, Rabbat MG, Fischione C (2016) On the convergence of alternating direction Lagrangian methods for nonconvex structured optimization problems. *IEEE Trans Control Netw Syst* 3:296–309
- Masazade E, Fardad M, Varshney PK (2012) Sparsity-promoting extended Kalman filtering for target tracking in wireless sensor networks. *IEEE Signal Process Lett* 19:845–848
- Mazurek A (2012) Geometrical aspects of optimum truss like structures for three-force problem. *Struct Multidiscip Optim* 45:21–32
- Mazurek A, Baker WF, Tort C (2011) Geometrical aspects of optimum truss like structures. *Struct Multidiscip Optim* 43:231–242
- Mela K (2014) Resolving issues with member buckling in truss topology optimization using a mixed variable approach. *Struct Multidiscip Optim* 50:1037–1049
- Michell AGA (1904) The limits of economy of material in frame-structures. *Lond Edinb Dublin Philos Mag J Sci* 8:589–597
- Miyashiro R, Takano Y (2015) Mixed integer second-order cone programming formulations for variable selection in linear regression. *Eur J Oper Res* 247:721–731
- Natarajan BK (1995) Sparse approximate solutions to linear systems. *SIAM J Comput* 24:227–234
- Ohsaki M (2011) Optimization of finite dimensional structures. CRC Press, Boca Raton
- Ohsaki M, Kanno Y, Tsuda S (2014) Linear programming approach to design of spatial link mechanism with partially rigid joints. *Struct Multidiscip Optim* 50:945–956

- Parkes EW (1975) Joints in optimum frameworks. *Int J Solids Struct* 11:1017–1022
- Pólik I (2005) Addendum to the SeDuMi user guide: version 1.1. Technical Report, Advanced Optimization Laboratory, McMaster University, Hamilton (2005). <http://sedumi.ie.lehigh.edu/>
- Prager W (1977) Optimal layout of cantilever trusses. *J Optim Theory Appl* 23:111–117
- Prager W (1978) Optimal layout of trusses with finite numbers of joints. *J Mech Phys Solids* 26:241–250
- Rozvany GIN (1996) Difficulties in truss topology optimization with stress, local buckling and system stability constraints. *Struct Optim* 11:213–217
- Sagnol G (2012) PICOS: a python interface for conic optimization solvers. <http://picos.zib.de/>. Accessed Feb 2017
- Sturm JF (1999) Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim Methods Softw* 11(12):625–653
- Takapoui R, Moehle N, Boyd S, Bemporad A (2018) A simple effective heuristic for embedded mixed-integer quadratic programming. *Int J Control*, to appear. <https://doi.org/10.1080/00207179.2017.1316016>
- Topping BHV (1983) Shape optimization of skeletal structures: a review. *J Struct Eng* 109:1933–1951
- Torii AJ, Lopez RH, Miguel LFF (2016) Design complexity control in truss optimization. *Struct Multidiscip Optim* 54:289–299
- Tütüncü RH, Toh KC, Todd MJ (2003) Solving semidefinite-quadratic-linear programs using SDPT3. *Math Program B* 95:189–217
- Zheng X, Sun X, Li D, Sun J (2014) Successive convex approximations to cardinality-constrained convex programs: a piecewise-linear DC approach. *Comput Optim Appl* 59:379–397