Struct Multidisc Optim (2009) 37:377–393
DOI 10.1007/s00158-008-0237-4

RESEARCH PAPER

Topology optimization of trusses by growing ground structure method

T. Hagishita · M. Ohsaki

Received: 26 September 2007 / Revised: 6 December 2007 / Accepted: 23 January 2008 / Published online: 12 April 2008

Abstract A new method called the growing ground structure method is proposed for truss topology optimization, which effectively expands or reduces the ground structure by iteratively adding or removing bars and nodes. The method uses five growth strategies, which are based on mechanical properties, to determine the bars and nodes to be added or removed. Hence, the method can optimize the initial ground structures such that the modified, or grown, ground structures can generate the optimal solution for the given set of nodes. The structural data of trusses are manipulated using C++ standard template library and the Boost Graph Library, which help alleviate the programming efforts for implementing the method. Three kinds of topology optimization problems are considered. The first problem is a compliance minimization problem with cross-sectional areas as variables. The second problem is a minimum compliance problem with the nodal coordinates also as variables. The third problem is a minimum volume problem with stress constraints under multiple load cases. Six numerical examples corresponding to these three problems are solved to demonstrate the performance of the proposed method.

T. Hagishita (✉) · M. Ohsaki
Department of Architecture and Architectural Engineering,
Kyoto University, Kyoto-Daigaku Katsura,
Nishikyo, Kyoto, Japan
e-mail: is.hagishita@archi.kyoto-u.ac.jp

M. Ohsaki
e-mail: ohsaki@archi.kyoto-u.ac.jp

Keywords Truss · Topology optimization · Singular optimum · Potential strain · Growing ground structure · C++ template library

1 Introduction

The ground structure method proposed by Dorn et al. (1964), hereafter abbreviated as *GSM*, is widely used in topology optimization of trusses. Although *GSM* can easily find an optimal solution, the quality of the solution depends on the locations of the nodes and the connectivity of bars of the initial *ground structure*. Therefore, the initial *ground structure* should be dense enough to obtain optimum topology for the given set of nodes because *GSM* assumes only the removal of bars. However, it is laborious to prepare the densest truss with a bar for every pair of nodes. Hence, another optimization method that does not need the construction of the densest truss is necessary. It should be noted that, throughout this paper, the term *ground structure* is used to indicate the structure to which *GSM* is applied regardless of its density. Hence, a sparse truss without bars for several pairs of nodes is also called ground structure.

It is well known for the problem with stress constraints that the stress constraints suddenly disappear as the cross-sectional area approaches zero, and, accordingly, degenerated feasible regions are generated (Kirsch 1993). Therefore, it is difficult to obtain the optimum solution in the degenerated region by *GSM* because the stress constraints of some nonexisting bars of the optimal truss may be violated in the region. We call this kind of solution the *singular optimum solution* for brevity. For an optimization problem that has a

singular optimum solution, it is not necessarily true that an initial ground structure with more bars can lead to a better optimal solution. Therefore, *GSM* has a shortcoming that the solution strongly depends on the initial ground structure. Hence, another method to optimize the topology of the ground structure is needed.

In this paper, a new method called *growing ground structure method*, hereafter abbreviated as *GGSM*, is proposed, which considers addition and removal of bars and nodes based on five growth strategies to optimize the ground structure. In the method, the initial ground structure need not be dense because the structure is optimized such that a better solution can be obtained by effectively modifying the ground structure based on the optimization results obtained after applying *GSM*.

Three kinds of optimization problems under static loads are considered. The details of problem formulation are presented in Section 2. Problem 1 (P1) is to minimize the compliance under constraint on the total structural volume with the cross-sectional areas as design variables. The problem can be reduced to the equivalent linear programming (LP) problem with only the nodal displacements as design variables (Achtziger et al. 1992; Bendsøe and Sigmund 2003). The singular optimum solution does not exist for this problem, and the global optimal truss with respect to the connectivity of bars for a given set of nodes can be obtained by assuming an initial ground structure as the trusses with every pair of nodes connected by a bar. However, the number of bars increases exponentially with respect to the number of nodes. Hence, *GSM* may not be efficient for problems with large sizes considering the cost for constructing the densest trusses without the overlapping bars. Therefore, we apply the *GGSM* for this problem to effectively optimize the ground structure.

Problem 2 (P2) is to minimize the compliance under constraint on the total structural volume with the cross-sectional areas and the nodal coordinates as design variables. The problem can be reduced to a simple nonlinear programming (NLP) problem through the equivalent reformulation (Achtziger 2007). For this problem, the phenomena called *coalescent nodes*, or *melting nodes*, and/or *collinear bars* may happen (Ohsaki 1998), which are discussed in Section 3. Note that existence of *melting nodes* and *collinear bars* leads to convergence to a local optimal solution with respect to the nodal coordinates and the predefined number of nodes of an initial ground structure. The occurrence of the phenomena mentioned above may depend on the connectivity of initial ground structure. However, it is difficult to predict the emergence of the phenomena. Therefore, we apply *GGSM*, which can adaptively change the connectivity of the nodes and bars. Moreover, addition of nodes is also considered to increase the degrees of freedom that may be reduced due to emergence of the melting nodes.

Problem 3 (P3) is to minimize the total structural volume under stress constraints with the cross-sectional areas as design variables. The truss is subjected to multiple load cases. It is well-known that the optimum solution may be singular for this problem. There are some approaches for obtaining singular optimum solutions. The ε-relaxation approach proposed by Cheng and Guo (1997) is widely used and discussed intensively. The approach aims at obtaining a singular optimum solution by relaxing the stress constraints by small amount ε, thereby avoiding the degeneracy of the feasible region. However, the parameter ε should be assigned through trial-and-error. There exists an extension of the approach based on the continuation method (Cheng and Guo 1997). In the method, the cost for adjusting ε may be alleviated; however, there still remains a few parameters to be adjusted through trial-and-error. For these methods, the discontinuity of the optimal objective value with respect to ε has been reported (Stolpe and Svanberg 2001). Hence, it is inevitable for the methods to find singular optimum solution by adjusting ε or the parameters. Singular optimum solution can also be found by a branch-and-bound method (Ohsaki and Katoh 2005), which leads to substantial computational effort for large problems. Therefore, we apply *GGSM* for P3 to obtain the singular optimum solution within the practically acceptable computational cost.

There have been some topology optimization methods considering the addition of bars and nodes by heuristics (Rule 1994). Reddy and Cagan (1994) proposed the shape annealing method based on the shape grammar to generate neighborhood solutions. The neighborhood solutions are evaluated one by one, then the best solutions among them are chosen to update the current best solution. The method considers not only the reduction but also the expansion of the number of bars and nodes of truss. Hence, the number of nodes and bars are also optimized in the optimization. However, the shape grammar is not related to the mechanical properties of the trusses obtained so far. Therefore, the method is purely heuristic and requires a large computational effort to evaluate a large number of neighborhood solutions to obtain a high-quality solution. The growth method has been proposed in Martinez et al. (2007), Mckeown (1998), Rule (1994). They proposed an interesting multistage topology optimization method in which topological growth plays a vital role to expand, or to grow, the trusses. The addition of node was performed to increase the orthogonality of bars in the trusses. However, the increase of the orthogonality

may not have clear relation to the mechanical properties of the trusses. Moreover, the iterative addition of bars is not considered and the removal of bars is preformed by *GSM*. Hence, the method is applicable to a few kinds of truss optimization problems. Considering the difficulties stated above, we introduce the growth strategies based on the mechanical properties of the trusses under static loads. The details of the growth strategies are summarized in Section 3. Then, we propose the algorithms of *GGSM* that integrate the addition/removal strategies and *GSM*. The details of the algorithms are summarized in Section 4.

Kirsch (1996) proposed a two-phase method for truss topology optimization that integrates reduction and expansion processes. However, it may be laborious to implement the method using a procedural programming language, e.g., FORTRAN, because the reformulation of the problem is necessary with the change of the connectivity or the number of degrees of freedom. We use the C++ standard template library and Boost Graph Library to implement the algorithms of *GGSM*. The details are summarized in Section 5.

In Section 6, the performance of *GGSM* is examined through six numerical examples and, finally, we discuss the advantages of applying *GGSM* over the traditional *GSM*. Moreover, for P3, the efficiency of *GGSM* is evaluated by comparing the optimization results with those by the continuation method.

2 Problem formulations

Three kinds of truss topology optimization problems (P1–P3) are formulated in this section. We assume two-dimensional trusses for simple presentation of the formulations.

Let m and N denote the total numbers of bars and nodes, respectively. The numbers of fixed and free degrees of freedom are denoted by t and n, respectively; i.e. $2N = t + n$. The so-called *geometry vector* $\hat{\mathbf{b}}_i \in \mathbb{R}^{2N}$ $(i = 1, \ldots, m)$ is defined as

$$\hat{b}_{i,p} = \begin{cases} \cos\theta_i & \text{if} \quad p = 2r - 1 \\ -\cos\theta_i & \text{if} \quad p = 2s - 1 \\ \sin\theta_i & \text{if} \quad p = 2r \\ -\sin\theta_i & \text{if} \quad p = 2s \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\hat{b}_{i,p}$ is the pth component of vector $\hat{\mathbf{b}}_i$. r and s $(r < s)$ are the indices of the end nodes of bar i and θ_i is the angle between the x-coordinate axis and the ith bar directing from node r to node s. $\mathbf{b}_i \in \mathbb{R}^n$ denotes the reduced geometry vector, which is composed by removing the components corresponding to the fixed degrees of freedom from $\hat{\mathbf{b}}_i$. The equilibrium matrix $\mathbf{B} \in \mathbb{R}^{n \times m}$ is defined as $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_m]$. We denote axial force vector and external load vector as $\mathbf{q} \in \mathbb{R}^m$ and $\mathbf{f} \in \mathbb{R}^n$, respectively. Then, the equilibrium equation is written as

$$\mathbf{Bq} = \mathbf{f} \tag{2}$$

Let $\mathbf{u} \in \mathbb{R}^n$ denote the nodal displacement vector. Then, the elongation of bar i is written as $\mathbf{b}_i^T\mathbf{u}$. The global stiffness matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ is defined as

$$\mathbf{K} = \sum_{k=1}^{m} \mathbf{K}_i \tag{3}$$

$$\mathbf{K}_i = a_i \frac{E_i}{l_i} \mathbf{b}_i \mathbf{b}_i^T \tag{4}$$

where a_i, E_i and l_i are the cross-sectional area, Young's modulus, and the length of bar i, respectively.

The first problem is to minimize the compliance under constraint on the total structural volume, considering the cross-sectional areas of bars $\mathbf{a} \in \mathbb{R}^m$ as design variables, where the single load case is assumed. The problem is formulated as

$$\begin{aligned} (\mathrm{P1}') \quad & \min_{\mathbf{a},\mathbf{u}} \ \mathbf{f}^T\mathbf{u} \\ \text{s.t.} \quad & \mathbf{K}(\mathbf{a})\mathbf{u} = \mathbf{f} \\ & \sum_{i=1}^{m} a_i l_i \le V \\ & a_i \ge 0, \quad (i = 1, \ldots, m) \end{aligned} \tag{5}$$

where V is the upper bound on the total structural volume.

(P1′) can be reduced to the following LP problem (P1) with only the nodal displacements as design variables (Achtziger et al. 1992):

$$\begin{aligned} (\mathrm{P1}) \quad & \max_{\mathbf{u}} \ \mathbf{f}^T\mathbf{u} \\ \text{s.t.} \quad & -1 \le \frac{\sqrt{E_i}}{l_i} \mathbf{b}_i^T\mathbf{u} \le 1, \quad (i = 1, \ldots, m) \end{aligned} \tag{6}$$

It is well-known that the dual problem of (P1) is the classical *plastic design* with the axial forces as design variables (Bendsøe and Sigmund 2003).

The second problem is to minimize the compliance under constraint on the total structural volume with the cross-sectional areas and the nodal coordinates as

design variables, where the single-load case is assumed. The problem is formulated as

$$\text{(P2}') \quad \min_{\mathbf{y},\mathbf{a},\mathbf{u}} \ \mathbf{f}^T\mathbf{u}$$

$$\text{s.t.} \quad \mathbf{K}(\mathbf{y},\mathbf{a})\mathbf{u} = \mathbf{f}$$

$$\sum_{k=1}^{m} a_i l_i(\mathbf{y}) \le V$$

$$a_i \ge 0, \quad (i = 1, \ldots, m) \tag{7}$$

where the vector $\mathbf{y} \in \mathbb{R}^{2N}$ represents the nodal coordinates in two directions.

(P2′) can be reduced to the following simple NLP problem (P2) (Achtziger 2007):

$$\text{(P2)} \quad \min_{\mathbf{y},\boldsymbol{\lambda},\mu} \sum_{i=1}^{m} (\lambda_i + \mu_i)\mathbf{y}^T\mathbf{C}_i\mathbf{y}$$

$$\text{s.t.} \quad \sum_{i=1}^{m} (\mu_i - \lambda_i)\sqrt{E_i}\mathbf{P}\mathbf{C}_i\mathbf{y} + \mathbf{f} = 0$$

$$\lambda_i \ge 0, \quad \mu_i \ge 0, \quad (i = 1, \ldots, m) \tag{8}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ and $\boldsymbol{\mu} \in \mathbb{R}^m$ are the variables corresponding to the Lagrange multipliers for displacement formulation of (P2′). See Achtziger (2007) for definition of $\mathbf{P} \in \mathbb{R}^{2N\times 2N}$ and $\mathbf{C}_i \in \mathbb{R}^{n\times 2N}$, which can successfully incorporate, in the optimization process, the existence of melting nodes discussed in Section 3.

The third problem is to minimize the total structural volume under stress constraints considering multiple load cases. In the following, the number of load cases is restricted to two for simple presentation of the formulation. The problem is written as

$$\text{(P3)} \quad \min_{\mathbf{a},\mathbf{q}_1,\mathbf{q}_2,\mathbf{u}_1,\mathbf{u}_2} \ \sum_{i=1}^{m} a_i l_i$$

$$\text{s.t.} \quad \mathbf{B}\mathbf{q}_1 = \mathbf{f}_1$$

$$\mathbf{B}\mathbf{q}_2 = \mathbf{f}_2$$

$$q_{1,i} = a_i(E_i/l_i)\mathbf{b}_i^T\mathbf{u}_1, \quad (i = 1, \ldots, m)$$

$$q_{2,i} = a_i(E_i/l_i)\mathbf{b}_i^T\mathbf{u}_2, \quad (i = 1, \ldots, m)$$

$$-a_i\sigma_i^a \le q_{1,i} \le a_i\sigma_i^a, \quad (i = 1, \ldots, m) \tag{9}$$

$$-a_i\sigma_i^a \le q_{2,i} \le a_i\sigma_i^a, \quad (i = 1, \ldots, m) \tag{10}$$

$$a_i \ge 0, \quad (i = 1, \ldots, m) \tag{11}$$

where σ_i^a is the allowable stress for bar i, for which the absolute values of the bounds in tension and compression are assumed to be the same. $\mathbf{f}_1 \in \mathbb{R}^n$ and $\mathbf{f}_2 \in \mathbb{R}^n$ are the two nodal load vectors. $\mathbf{q}_1 \in \mathbb{R}^m$, $\mathbf{u}_1 \in \mathbb{R}^n$ and $\mathbf{q}_2 \in \mathbb{R}^m$, $\mathbf{u}_2 \in \mathbb{R}^n$ are the axial force vectors and nodal displacement vectors for $\mathbf{f}_1$ and $\mathbf{f}_2$, respectively. Note that there may exist a singular optimum solution for this problem. We exclude a single load case because a singular optimum solution for single load case can be found by solving an equivalent LP problem, which is the dual problem of P1 (Stolpe and Svanberg 2004).

ε-relaxation approach formulates P3 by relaxing constraints (9), (10) and (11) as (12), (13) and (14), respectively as

$$-a_i\sigma_i^a - \varepsilon \le q_{1,i} \le a_i\sigma_i^a + \varepsilon, \quad (i = 1, \ldots, m) \tag{12}$$

$$-a_i\sigma_i^a - \varepsilon \le q_{2,i} \le a_i\sigma_i^a + \varepsilon, \quad (i = 1, \ldots, m) \tag{13}$$

$$a_i \ge \varepsilon^2, \quad (i = 1, \ldots, m) \tag{14}$$

Then, the value for ε is determined through trial-and-error to obtain a singular optimum solution. We call the ε-relaxed P3 as (P3$_\varepsilon$).

On the other hand, there exists an extension of the ε-relaxation approach based on the continuation method (Cheng and Guo 1997), in which the value for ε is decreased at a specified rate starting from a relatively large value for the initial ε. If we denote the epsilon, the solution vector at kth iteration, and the reduction rate by ε_k, $\mathbf{s}_k$, and γ, the algorithm of the continuation method is summarized as

Initialization ($k = 1$, $\varepsilon_k \leftarrow \varepsilon_0$, and $\mathbf{s}_k \leftarrow \mathbf{s}_0$)
While ($\varepsilon_k > \varepsilon_{min}$)
  Solve (P3$_{\varepsilon_k}$) with $\mathbf{s}_k$ as the initial solution.
  Obtain solution $\mathbf{s}$, and set $k = k + 1$.
  Update $\mathbf{s}_k \leftarrow \mathbf{s}$, $\varepsilon_k \leftarrow \gamma \times \varepsilon_{k-1}$ $(0 < \gamma < 1)$.
End

The continuation method would reduce the cost for adjusting ε; however, (P3$_{\varepsilon_k}$), namely the relaxed NLP problem, needs to be solved at each iteration. In Section 6, we apply the method to evaluate the efficiency of the *GGSM* by simply comparing the number of times the NLP problems are solved to obtain optimal, or nearly optimal, solution.

It should be noted that, in the continuation method, the values of ε_0, γ, $\mathbf{s}_0$ and ε_{min} should be determined through trial-and-error. However, the values of $\mathbf{s}_0$, namely the initial cross-sectional areas, are not influential for obtaining better solution if ε_0 is large because, in that case, the constraints (14) at first iteration become too severe, accordingly the cross-sectional areas would reach their lower bounds.

ε_{min} can be set very small because we can observe the performance of the method with a larger value of ε_{min} by tracing the history. Hence, we would change the values of ε_0 and γ several times and compare the efficiency with *GGSM* in Section 6.

3 Growth strategies for addition and removal of bars and nodes

In this section, five growth strategies for addition and removal of bars and nodes are presented based on mechanical properties of the truss. These strategies are vital components to expand and/or reduce, or grow, the ground structures in the algorithms of *GGSM* proposed in Section 4.

3.1 Growth strategy 1: addition of a bar

The growth strategy 1, hereafter abbreviated as *GSTRG1*, for addition of a single bar is introduced. *GSTRG1* is used in Algorithms 1 and 2 in Section 4. First, the strategy chooses *candidate bars*, which do not exist for a given set of nodes (see Fig. 1). Then, *potential strain* is calculated for each candidate bar from the nodal displacements of the truss. The procedure for calculating the potential strain is summarized as follows (see Fig. 2):

(a) Let $\mathbf{w}_i = (u_{i,x}, u_{i,y})^T$ and $\mathbf{w}_j = (u_{j,x}, u_{j,y})^T$ denote the displacement vectors of end nodes i and j of bar k, where $u_{i,x}$ denotes the displacement of node i in x-direction, and $u_{i,y}$, $u_{j,x}$ and $u_{j,y}$ are used similarly. The coordinates of nodes i and j are denoted by $\mathbf{x}_i = (x_i, y_i)^T$ and $\mathbf{x}_j = (x_j, y_j)^T$, respectively.

(b) The potential strain $\bar{\eta}_k$ of bar k is calculated following the definition of the engineering strain as

$$\mathbf{w}_{ij} = \mathbf{w}_j - \mathbf{w}_i, \quad \mathbf{d}_{ij} = \frac{\mathbf{x}_j - \mathbf{x}_i}{||\mathbf{x}_i - \mathbf{x}_j||}$$

$$\bar{\eta}_k = \frac{\mathbf{w}_{ij}^T \mathbf{d}_{ij}}{||\mathbf{x}_i - \mathbf{x}_j||} \tag{15}$$

where $||\bullet||$ denotes Euclidean vector norm.

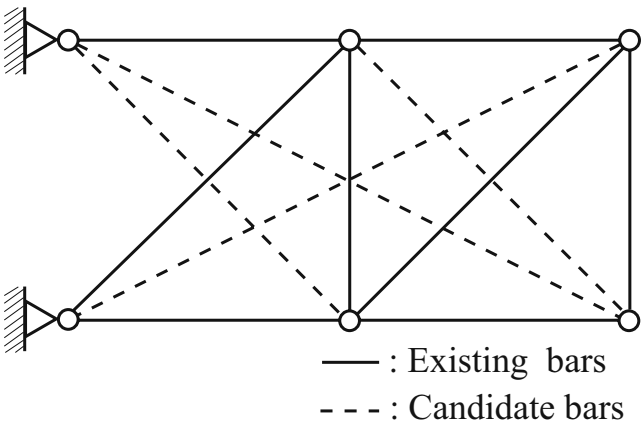


Fig. 1 Candidate bars

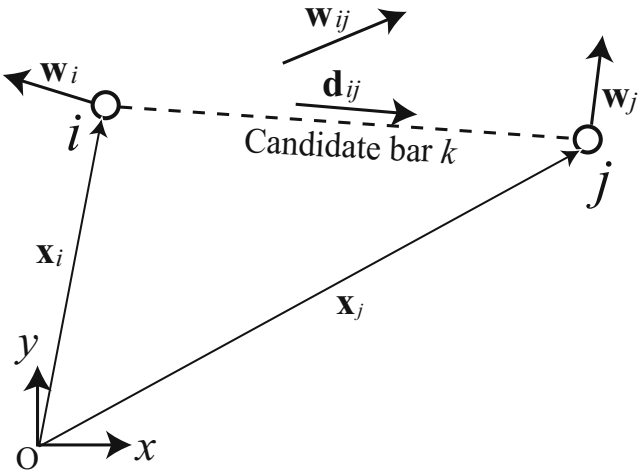


Fig. 2 Potential strain

GSTRG1 decides the bar to be added by comparing the absolute values of *potential strains* for candidate bars, namely the most strained bar should be added. However, the effect of the addition of a bar should be compared by assigning the same volume to each candidate bar. Hence, we multiply the potential strain by the cross-sectional area $1/l_k$ corresponding to the unit volume to the potential bar. Therefore, *GSTRG1* can be written as

$$\text{(GSTRG1)} \qquad \text{Find} \quad k \in \{candidate\ \ bars\}$$
$$\text{which} \quad \text{maximize} \quad \frac{1}{l_k}|\bar{\eta}_k| \tag{16}$$

It should be noted that *GSTRG1* does not directly evaluate the nodal displacements of the modified truss with a candidate bar added. Therefore, it does not require additional structural analysis; i.e., the computational cost is very small.

To clarify this point, we consider the effect of adding bar k with unit volume to a given truss. The stiffness equation for finding the additional nodal displacements $\delta\mathbf{u}_k$ is given as

$$(\mathbf{K} + \mathbf{K}_k)\delta\mathbf{u}_k = -\mathbf{K}_k\mathbf{u} \tag{17}$$

$$\mathbf{K}_k = \frac{E_k}{l_k^2}\mathbf{b}_k\mathbf{b}_k^T \tag{18}$$

where $\mathbf{u}$ is the nodal displacement vector before adding bar k. It is seen from (17) that the internal force $-\mathbf{K}_k\mathbf{u}$ emerges by adding bar k, and it generates $\delta\mathbf{u}_k$. If we specify $E_k = 1$, the right term in (17) is equivalent to the potential strain multiplied by cross-sectional area $a_k = 1/l_k$. Hence, *GSTRG1* only evaluates the internal force from $\mathbf{u}$ without calculating $\delta\mathbf{u}_k$.

Growth strategies 2, 3, and 4, hereafter abbreviated as *GSTRG2*, *GSTRG3*, and *GSTRG4*, respectively, are introduced in the following. These strategies directly evaluate the effect of adding bar k by calculating $\mathbf{u} + \delta\mathbf{u}_k$.

3.2 Growth strategies 2 and 3: addition of a bar

GSTRG2 and *GSTRG3* for addition of a single bar are introduced. *GSTRG2* is used in Algorithms 1 and 2 and *GSTRG3* is used in Algorithm 3 in Section 4. Both *GSTRG2* and *GSTRG3* choose *candidate bars* in the same manner as in *GSTRG1*. These strategies use criteria based on the displacement after the addition of a candidate bar to a given truss to compare their effectiveness. Hence, the structural analysis should be performed many times to calculate the displacements for the candidate trusses corresponding to candidate bars. However, we can apply the so-called *exact reanalysis* well-developed in, e.g., Kirsch (2002), Ohsaki (2001) because we assume addition of a single candidate bar. Hence, the computational effort is reduced if we use the methods. The criteria used to compare the effectiveness of adding a bar for two strategies are introduced in the following.

For *GSTRG2* and *GSTRG3*, $\mathbf{u} + \delta\mathbf{u}_j$ is calculated after adding a candidate bar j. Then, the elongation $d_{k,j}$, the strain $\eta_{k,j}$, and the stress $\sigma_{k,j}$ are calculated for each bar k, $(k = 1, .., m+1)$ of the modified truss as

$$d_{k,j} = \mathbf{b}_k^T(\mathbf{u} + \delta\mathbf{u}_j), \quad \eta_{k,j} = \frac{d_{k,j}}{l_k}, \quad \sigma_{k,j} = E_k\eta_{k,j} \tag{19}$$

GSTRG2 is used in Algorithms 1 and 2 in Section 4 for solving P1 and P2 formulated in Section 2, namely, the standard compliance minimization problem, for which the optimal solution is fully strained. Hence, it is appropriate to evaluate the effect of adding candidate bar j with the specified volume based on the deviation τ_j of the strains for all the bars of the truss as:

$$\tau_j = \frac{1}{m+1}\sum_{k=1}^{m+1}(|\eta_{k,j}| - \nu_j)^2, \quad \nu_j = \frac{1}{m+1}\sum_{k=1}^{m+1}|\eta_{k,j}| \tag{20}$$

Because a larger value of τ_j gives more disturbance to the truss, it increases the possibility of obtaining a better optimal solution as a result of searching different design space. Hence, *GSTRG2* can be written as:

$$\begin{aligned} &\text{(GSTRG2)} \qquad \text{Find} \quad j \in \{\textit{candidate bars}\} \\ &\qquad\qquad\quad \text{which} \quad \text{maximize} \quad \tau_j \\ &\qquad\qquad\quad \text{s.t.} \quad a_j l_j = \textit{const}. \end{aligned} \tag{21}$$

GSTRG3 is used in Algorithm 3 in Section 4 for solving P3 formulated in Section 2, that is, the volume minimization problem under stress constraints, for which the optimal solution may be fully stressed in almost all bars. Hence, we can directly evaluate the effectiveness of adding bar j with the specified volume based on the deviation H_j of the stress constraints defined as:

$$H_j = \frac{1}{m+1}\sum_{k=1}^{m+1}\left(\frac{|\sigma_{k,j}|}{\sigma_k^a} - 1\right)^2 \tag{22}$$

Based on the similar observation as *GSTRG2*, the candidate bar with the maximum H_j is to be added. Therefore, *GSTRG3* is summarized as:

$$\begin{aligned} &\text{(GSTRG3)} \qquad \text{Find} \quad j \in \{\textit{candidate bars}\} \\ &\qquad\qquad\quad \text{which} \quad \text{maximize} \quad H_j \\ &\qquad\qquad\quad \text{s.t.} \quad a_j l_j = \textit{const}. \end{aligned} \tag{23}$$

3.3 Growth strategy 4: removal of a bar

GSTRG4, for removing a single bar from a given truss, is introduced. *GSTRG4* is used in Algorithm 3 in Section 4, which is used to solve P3 formulated in Section 2. It is known that there sometimes exists a singular optimum solution for P3. Hence, the addition of bars is not enough to obtain optimal ground structure that can generate the singular optimum solution. Therefore, we extend the concept of *GSTRG3* to formulate *GSTRG4* to remove ineffective bars. To evaluate the effectiveness of removing bar j, M_j is introduced as

$$M_j = \frac{1}{m}\sum_{k=1}^{m}\left(\frac{|\sigma_{k,j}|}{\sigma_k^a} - 1\right)^2 \tag{24}$$

where $\sigma_{k,l}$ is calculated by removing existing bar j from a given truss in the similar manner as (19). However, it should be noted that bar j is not removed completely. Instead, the cross-sectional area of bar j is reduced to a very small value ζ_a to prevent the truss to be unstable. The bar that minimizes M_j is to be removed. Hence, *GSTRG4* is summarized as:

$$\begin{aligned} &\text{(GSTRG4)} \qquad \text{Find} \quad j \in \{\textit{existing bars}\} \\ &\qquad\qquad\quad \text{which} \quad \text{minimize} \quad M_j \\ &\qquad\qquad\quad \text{s.t.} \quad a_j = \zeta_a. \end{aligned} \tag{25}$$

It is well-known that P2 in Section 2, which simultaneously optimizes the cross-sectional areas and nodal coordinates, sometimes has *melting nodes*, where a few nodes connected with each other share the same coordinate, and/or *collinear bars*, where two or more bars are aligned in one line and can be substituted by a single bar.

When the melting nodes exist, these nodes cannot move separately because these nodes are not distinguishable if the nodes connected to each of them by bars are the same during the optimization. Accordingly, the nominal degrees of freedom decrease. Similarly, if the collinear bars emerge, a few bars have the same cross-sectional area and the nodes connecting these bars are functionless, namely, the nominal degrees of freedom are reduced. Hence, it is highly likely that the solution with melting nodes and/or collinear bars is a local optimal solution with respect to the locations of the nodes of truss because the solution can be replaced by a truss with fewer degrees of freedom generated by replacing melting nodes with a node and collinear bars with a bar.

The fundamental difficulty in topology optimization with melting nodes and/or collinear bars is the reduction of predefined degrees of freedom. The possibility of emergence of melting nodes and/or collinear bars depends on the connectivity of the initial ground structure, and it is very difficult to prevent these phenomena (Ohsaki 1998). Therefore, we introduce *GGSM*, which can iteratively change the connectivity of a truss to alleviate these difficulties. Moreover, the strategy to add a node is introduced to complement the reduction of degrees of freedom.

3.4 Growth strategy 5: addition of a node and bars

Growth strategy 5, *GSTRG5* in short, is presented for adding a node and n_f bars connected to the node. *GSTRG5* is used only in Algorithm 2 in Section 4. Let $\mathbf{x}_k = (x_k, y_k)^T$ denote the coordinate vector of the new node k. The displacement vectors of nodes of a truss are denoted as $\mathbf{w}'_i = (u_{i,x}, u_{i,y})^T$, $(i \in \{existing\ nodes\})$. Let $\mathcal{F}$ denote a subset of the nodes with the size n_f, which are candidates to be connected to the node k by bars. The relative displacement vectors $\mathbf{w}_j$ $(j \in \mathcal{F})$ are calculated by subtracting the average nodal displacement vector $\mathbf{w}_c$ over $\mathcal{F}$ as

$$\mathbf{w}_j = \mathbf{w}'_j - \mathbf{w}_c, \quad \mathbf{w}_c = \frac{1}{n_f}\sum_{j\in\mathcal{F}} \mathbf{w}'_j \tag{26}$$

Let the coordinate vectors of node j in $\mathcal{F}$ be denoted by $\mathbf{x}_j = (x_j, y_j)^T$. The length $l_{k,j}$ and the unit directional vectors $\mathbf{e}_{k,j}$ of the candidate bars, where the subscript indicates the candidate bar that connects the node j to node k, are calculated as

$$l_{k,j} = \sqrt{(x_k - x_j)^2 + (y_k - y_j)^2} \quad \text{for} \quad j \in \mathcal{F} \tag{27}$$

$$\mathbf{e}_{k,j} = \frac{1}{l_{k,j}}(x_k - x_j, \ \ y_k - y_j)^T \quad \text{for} \quad j \in \mathcal{F} \tag{28}$$

If we specify unit volume for each candidate bar as $a_{k,j} = 1/l_{k,j}$ $(j \in \mathcal{F})$, and assume constant Young's modulus as $E_{k,j} = E$ $(j \in \mathcal{F})$, where $a_{k,j}$ and $E_{k,j}$ are the cross-sectional area and Young's modulus for candidate bar connecting nodes k and j, the equilibrium equation at node k is written as

$$\sum_{j\in\mathcal{F}} \frac{E(\mathbf{e}_{k,j}^T \mathbf{w}_j)\mathbf{e}_{k,j}}{l_{k,j}^2} = \mathbf{0} \tag{29}$$

It should be noted that the displacement of the candidate node k is assumed to be $\mathbf{w}_c$ to formulate (29).

In a similar manner as Strategy 1, the sum of the absolute values of the potential strains multiplied by the cross-sectional area $a_{k,j} = 1/l_{k,j}$ $(j \in \mathcal{F})$, and divided by the number of candidate bars, to evaluate the efficiency for adding a node and n_f bars is calculated as

$$D_{\mathcal{F}} = \sum_{j\in\mathcal{F}} \frac{|\mathbf{e}_{k,j}^T \mathbf{w}_{k,j}|}{l_{k,j}^2 n_f} \tag{30}$$

GSTRG5 is summarized as

(GSTRG5) Find n_f, $\mathbf{x}_k$, $\mathcal{F} \subset \{\text{existing nodes}\}$ which maximize $D_{\mathcal{F}}$

$$\text{s.t.} \quad \sum_{j\in\mathcal{F}} \frac{E(\mathbf{e}_{k,j}^T \mathbf{u}_{k,j})\mathbf{e}_{k,j}}{l_{k,j}^2} = \mathbf{0} \tag{31}$$

4 Growing ground structure method (*GGSM*)

In this section, we introduce three algorithms of *GGSM* that extend the *GSM* to include addition of bars and nodes based on the five strategies introduced in Section 3. Algorithms 1, 2, and 3 are used to solve P1, P2, and P3 formulated in Section 2, respectively. PropV and PropE used in the statement of the algorithms represent the properties of the nodes and bars of the truss, whose details are presented in Section 5. We use the terms *vertex* and *edge*, instead of node and bar, and *degree* in the algorithms, which are the terminologies in the graph theory (West 2001). Algorithms 1 and 2 are simultaneously summarized as follows:

4.1 Algorithms 1 and 2

I. Construct property graph $G = (V \cup \text{PropV}, E \cup \text{PropE})$ of the initial sparse ground structure.
II. Prepare data for optimization from G and perform optimization by *GSM*. The optimal cross-sectional areas, nodal displacements, and nodal coordinates are stored in PropV and PropE of G.

III. Make a copy RG of G. Remove the edges with zero cross-sectional area from RG by examining PropE. Then, remove vertices with zero degree.
IV. Extract the candidate bars from RG. Then, find the bar e to be added to G by applying *GSTRG1* or *GSTRG2* for RG. $G \leftarrow G(V, E \cup e)$, and go to II. If there is no candidate bar, go to V.
V. Carry out structural analysis for G, assuming that each bar has the unit volume, and store the nodal displacements in PropV. Extract the candidate bars from G and find the bar e to be added to G by applying *GSTRG1* or *GSTRG2* for G. $G \leftarrow G(V, E \cup e)$, and go to II. Terminate if there is no candidate bar for Algorithm 1. Go to II for Algorithm 2 if the number of nodes does not reach the predefined maximum value. Otherwise, terminate.
VI. Carry out structural analysis for G, assuming that each bar has the unit volume, and store the nodal displacement in PropV. Apply *GSTRG5* for G to find the location of the new node v and the n_f bars $e_1, .., e_{n_f}$ to be added to G. $G \leftarrow G(V \cup v, E \cup (e_1, .., e_{n_f}))$, and go to II.

Algorithm 3 is summarized below.

4.2 Algorithm 3

I. Construct property graph $G = (V \cup \text{PropV}, E \cup \text{PropE})$ of the initial sparse ground structure. The connectivity of G is stored in the tabu list as $\mathcal{T} \leftarrow E(G)$.
II. Carry out optimization by *GSM* for G and find the optimal objective value Obj1. The results of optimization are stored in PropV and PropE. Make a copy RG of G and remove the edges and vertices from RG in a manner similar to Algorithms 1 and 2.
III. Extract the candidate bars from RG and find the bar e to be added to G with $E(RG) \cup e \notin \mathcal{T}$ by applying *GSTRG3* for RG. Terminate if all the connectivities are in $\mathcal{T}$. Construct $G2 \leftarrow RG(V, E \cup e)$.
IV. Carry out optimization by *GSM* for $G2$ and find the optimal objective value Obj2. The results of optimization are stored in PropV and PropE of $G2$. The connectivities of RG and $G2$ are stored in the tabu list as $\mathcal{T} \leftarrow E(RG), E(G2)$
V. If Obj1 > Obj2, $G \leftarrow G2$, and go to II. If Obj1 = Obj2, $G2 \leftarrow G$, and go to VI. If Obj1 < Obj2, go to VI.
VI. Find the bar e with $E(G2)\backslash e \notin \mathcal{T}$ to be removed from the existing edges of $G2$ applying *GSTRG4* for $G2$. Store the connectivity of $G2(V, E\backslash e)$ in the tabu list as $\mathcal{T} \leftarrow E(G2)\backslash e$. Terminate if all the connectivities defined by $E(G2)\backslash e$ are in $\mathcal{T}$; otherwise, $G \leftarrow G2(V, E\backslash e)$, and go to II.

5 Implementation using C++ template libraries

C++ Standard Template Library and Boost Graph Library (BGL) are used to implement the proposed algorithms (Boost 2007; Meyers 2002; Siek et al. 2002). They provide a wide variety of data structures. Hence, the reformulation used intensively in the algorithms can easily be implemented. In this section, the major parts of program using BGL are introduced.

5.1 Construction of property graph

BGL realizes the *property graph*, in which the vertices and edges of a directed graph can have multiple properties.

We set up six properties for vertex and three properties for edges. The tags of properties and the type of data are declared in the definition of the graph type *graph-t* as below. The meaning of each property can be understood by its name. The sixth property for vertex is the identifier, which distinguishes free nodes, fixed nodes, and loaded nodes. The graph G is defined as the object of *graph-t*.

```
typedef adjacency_list<listS,listS,
 directedS,
property<vertex_node_no_t,int,
 property<vertex_xcoord_t,double,
  property<vertex_ycoord_t,double,
   property<vertex_xdisp_t,double,
    property<vertex_ydisp_t,double,
     property<vertex_load_t ,int >>>>>>,
property<edge_ith_t,int,
 property<edge_jth_t,int,
  property<edge_area_t,double> >>>
   graph-t;
graph-t G;
```

5.2 Definition of property maps and iterators

To associate the properties to graph G, *property map* should be defined. PropV and PropE used in Algorithms 1–3 correspond to the property maps. Only two property maps are defined as examples below. *Vertex iterators* and *edge iterators* are used to traverse all the vertices and edges of a graph. With these iterators, we can manage, read, and write the properties. The iterators are defined and the properties are written

to file indicated by *file-out*, the object of the *ofstream class*, in the following.

```
property_map
 <graph-t,vertex_node_no_t>::type
    G_v1_map=get(vertex_nodeid,G);
property_map <graph-t,edge_ith_t>::type
    G_e1_map=get(edge_ith,G);
graph_traits<graph-t>::
 vertex_iterator vi,vd;
graph_traits<graph-t>::
 edge_iterator ei,ed;
for(tie(vi,vd)=vertices(G);vi!=vd;++vi){
    file-out>> G_v1_map[*vi]
          >> G_v2_map[*vi]
          >> G_v3_map[*vi]
          >> G_v4_map[*vi]
          >> G_v5_map[*vi]
          >> G_v6_map[*vi];}
for(tie(ei,ed)=edges(G);ei!=ed;++ei){
    file-out>> G_e1_map[*ei]
            >> G_e2_map[*ei]
            >> G_e3_map[*ei] ;}
```

5.3 Removal of edges with zero cross-sectional area

In algorithms in Section 4, the bars with zero cross-sectional area are removed. We can implement this process using the edge iterators and *remove edge* function defined in BGL as follows:

```
graph_t RG(G);
EDGE: tie(ei,ed)=edges(RG);
 while(ei != ed)
 {if(G_e3_map[*ei]<1.0E-10)
 {remove_edge(*ei,RG);goto EDGE;} ei++;}
```

6 Numerical examples

Six numerical examples, Examples 1–6, are solved to demonstrate the performance of the *GGSM*. Examples 1 and 2 correspond to P1. Examples 3 and 4 are classified to P2, and Examples 5 and 6 to P3. We use SNOPT Ver. 7 (Gill et al. 1997), based on sequential quadratic programming, for solving LP and NLP problems. For Examples 5 and 6, design sensitivity coefficients are calculated analytically (Pedersen 1972).

In the following six examples, the structural parameters are given without units for simple presentation of the results. For all the examples, the structural dimensions are based on 1.0×1.0 unit. For Examples 1–4, we use Young's modulus $E_i = 1.0 \times 10^4$ for all bars, and the upper bound V on total structural volume is 10.0. The nodal load is $\mathbf{P}_1 = (0.0, -100.0)^T$. The number and locations of nodal loads are different for each example. For Algorithm 2 applied to Examples 3 and 4, n_f in *GSTRG5* are limited to be 3 through 6 because it is not probable that n_f is optimized to be larger numbers. For Examples 5 and 6, Young's modulus is $E_i = 1.0 \times 10^2$ for all bars, and the two allowable stresses $\sigma_I = \pm 5.0$ and $\sigma_{II} = \pm 20.0$ are used. The specific external nodal load is $\mathbf{P}_2 = (1.0, -5.0)^T$. The number and locations of $\mathbf{P}_2$ are specified in each problem setting. For Examples 5 and 6, the continuation method summarized in Section 3 is also applied. We specified the values for parameters in the method as: $\varepsilon_{min} = 1.0^{-4}$, $s_{0,i} = 2.0$, $(i = 1, ...m)$, where $s_{0,i}$ is ith component of $\mathbf{s}_0$. The objective function values for the six examples are indicated using the same notation Φ, and the bar connecting nodes i and j is indicated by i–j for simplicity.

6.1 Example 1: 12-bar truss P1

Figure 3 shows six-node truss for which we apply Algorithm 1, where the densest truss has 12 bars. We set the locations of nodes at all of the vertices of 2×1 grids. The single-load case is assumed with $\mathbf{P}_1$ values applied

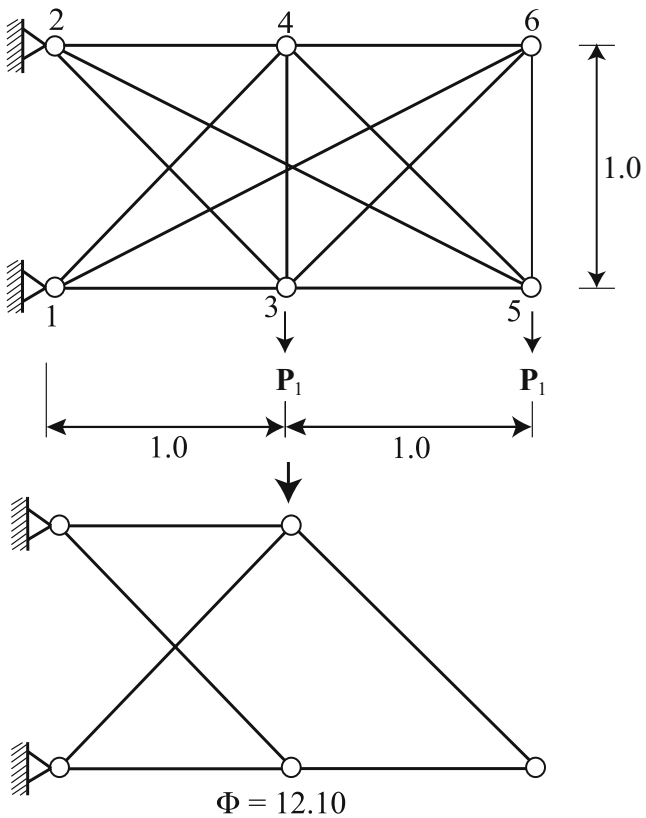


Fig. 3 Fully connected ground structure of six-node truss and its optimal solution for Example 1

at nodes 3 and 5 simultaneously, as shown in Fig. 3. Application of *GSM* for the 12-bar truss generates statically determinate six-bar truss with the objective function value $\Phi = 12.10$ as illustrated in Fig. 3, which is the global optimal truss. Note that the term *global optimal truss* is used for P1 to indicate the optimality of connectivity of bars for a given number of nodes located at fixed positions. In other words, the term is used to indicate the single or multiple trusses with the minimum objective value among the trusses obtained by applying *GSM* for any ground structures that can be constructed from a set of fixed nodes. It should be noted that there may exist multiple optimal trusses for P1, while it is unlikely for P2 or P3.

The optimization results by Algorithm 1 with *GSTRG1* are illustrated in Fig. 4. We use eight-bar and nine-bar trusses as the initial ground structures. For eight-bar truss, ground structure grows to a nine-bar truss by adding single bar, which can generate global optimal truss as shown in Fig. 4. Therefore, the growing process finds the *optimal ground structure* by adding a bar, although there are three more candidate bars to be added. It should be noted that, for P1, the term *optimal ground structure* is used to indicate the ground structure modified, or grown, by Algorithm 1, which can generate the global optimal truss by applying *GSM*. If we start from a nine-bar truss, the ground structure grows to a 10-bar truss that can obtain the optimal ground structure as shown in Fig. 4, although there are two more candidate bars to be added. Hence, Algorithm 1 with *GSTRG1* has been demonstrated to be effective. Other initial ground structures have also been optimized by Algorithm 1 to examine its performance. Almost all the structures follow the shortest paths to be the *optimal ground structures*.

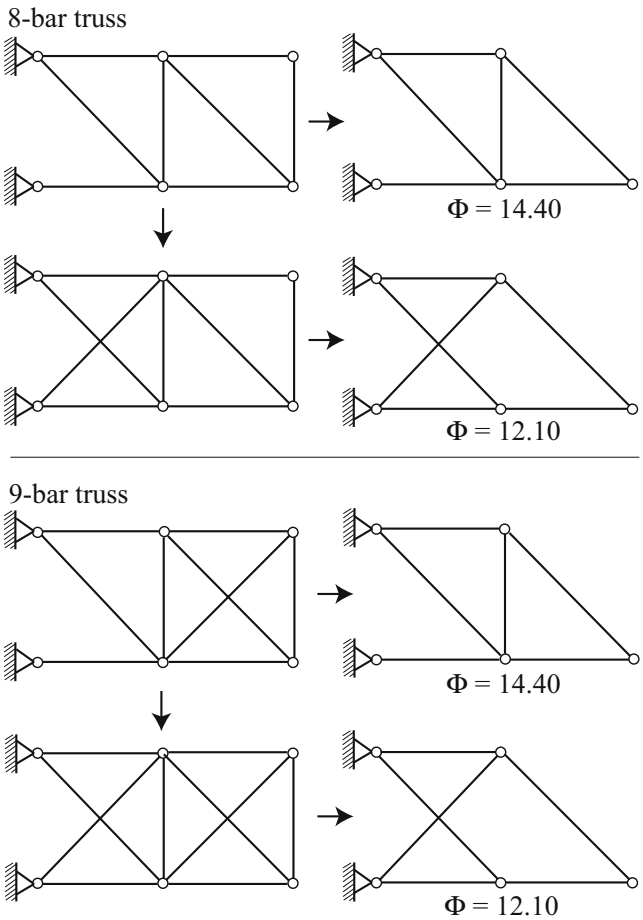


Fig. 4 Optimization results by Algorithm 1 (GSTRG1) for Example 1

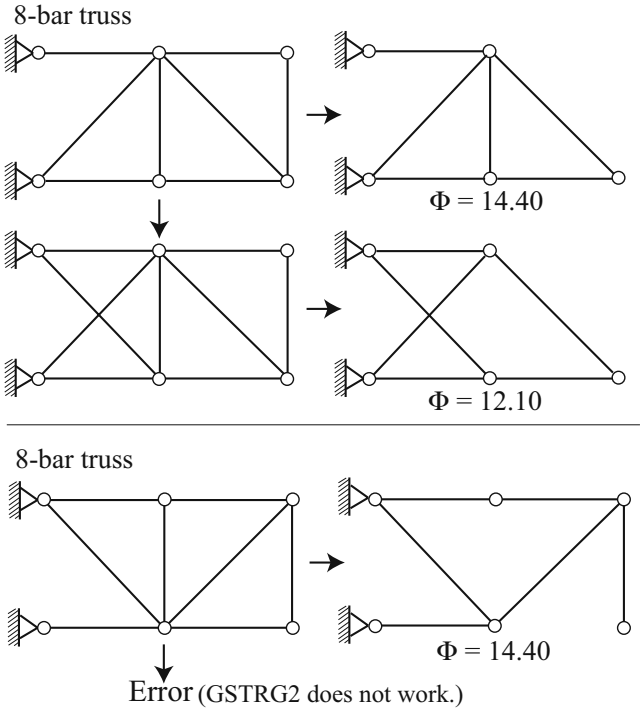


Fig. 5 Optimization results by Algorithm 1 (GSTRG2) for Example 1

We also used *GSTRG2* for Algorithm 1 for eight-bar truss, and *GSTRG2* has been proven to be effective, as shown in Fig. 5. However, *GSTRG2* sometimes fails if the truss optimized by *GSM* is judged as unstable, as shown in Fig. 5, because *GSTRG2* needs structural analysis. Hence, to ensure the stability of the optimization by *GGSM*, we use *GSTRG1* in Examples 2, 3, and 4 for the addition of a bar.

6.2 Example 2: 47-bar truss P1

Figure 6 shows 12-node truss for which we apply Algorithm 1, where the densest truss has 47 bars without the *overlapping bars*. The term *overlapping bars*

is used to indicate the bars that can be substituted by combining other bars. We excluded overlapping bars from the ground structure because they are not preferable from the construction point of view. We set the locations of nodes at all vertices of a 3×2 grid. The single-load case is assumed with $\mathbf{P}_1$ values applied at nodes 4, 7, and 10 simultaneously, as shown in Fig. 6. Application of *GSM* for the 47-bar truss generates a statically determinate 10-bar truss, which is a global optimal truss with the objective function value $\Phi = 36.10$, as illustrated in Fig. 6.

The optimization results by Algorithm 1 with *GSTRG1* are illustrated in Fig. 7 starting from the 21-bar truss as the initial ground structure. The ground structure grows to a 24-bar truss, or optimal ground structure, after sequentially adding three bars, 8–10, 1–8, and 3–8, which can generate an eight-bar truss with the objective function value $\Phi = 36.10$, that is also the global optimal truss. Then, after addition of bars 3–4, 3–10, 1–5, and 3–5 one by one, the *GGSM* can find a different optimal ground structure that can generate global optimal trusses with the objective function value $\Phi = 36.10$. After adding seven bars in total, the ground structure grows to a denser optimal truss that can generate the 10-bar truss that is exactly the same for the result obtained from the 47-bar ground structure. It should be noted that the Algorithm 1 has found six different optimal ground structures.

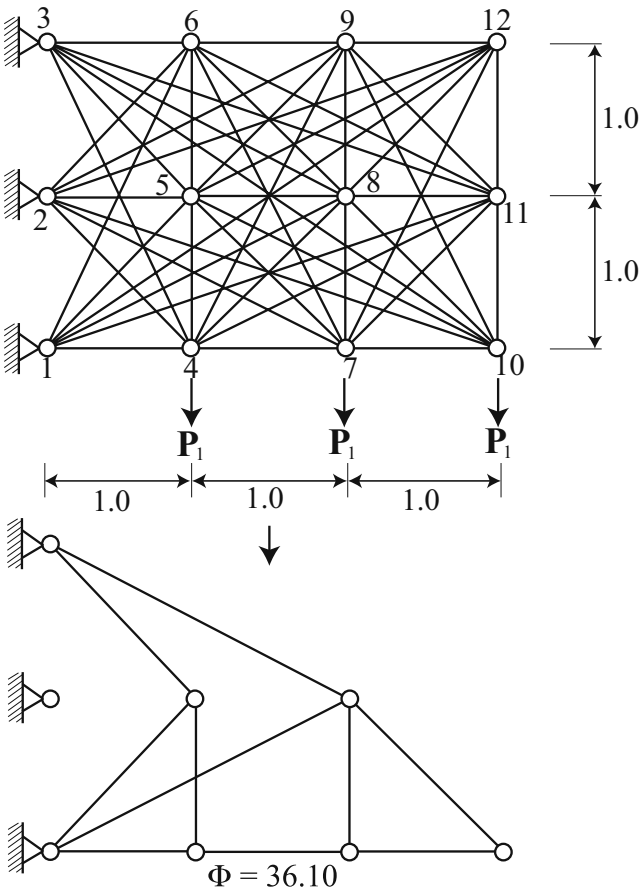


Fig. 6 Fully connected ground structure of 12-node truss and its optimal solution for Example 2

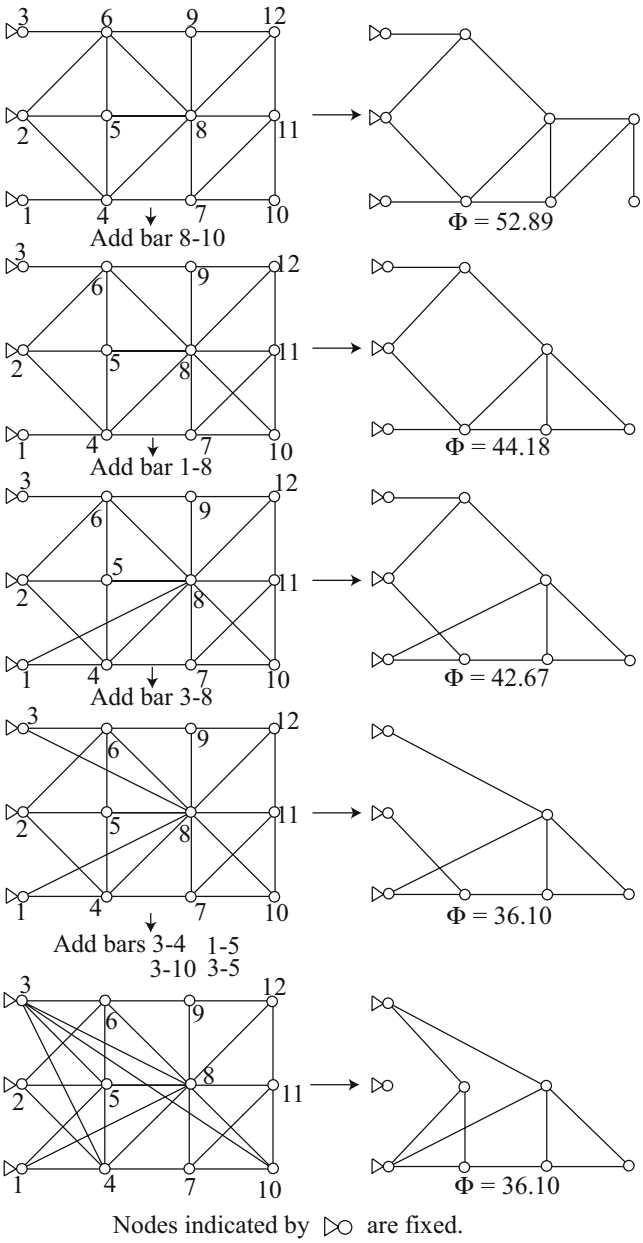
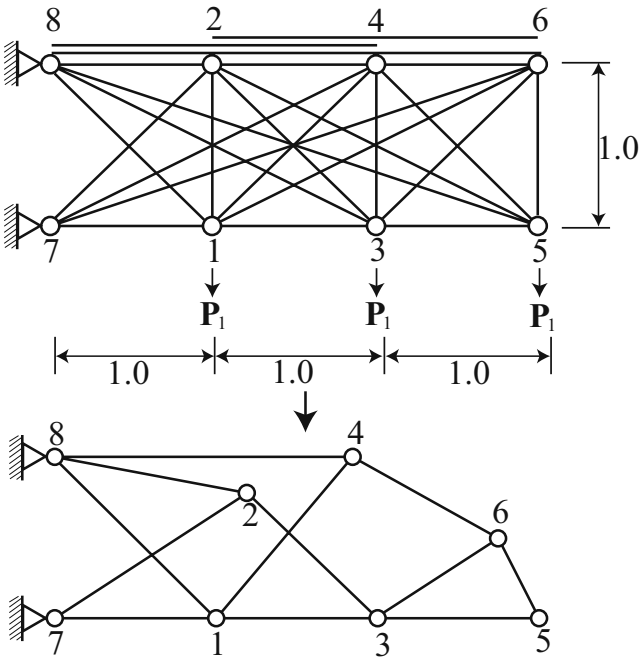


Fig. 7 Optimization results by Algorithm 1 (GSTRG1) for Example 2

6.3 Example 3: 24-bar truss P2

Figure 8 shows an eight-node truss for which we apply Algorithm 2, where the densest truss has 24 bars with several overlapping bars. We include the overlapping bars, which can change their location due to change

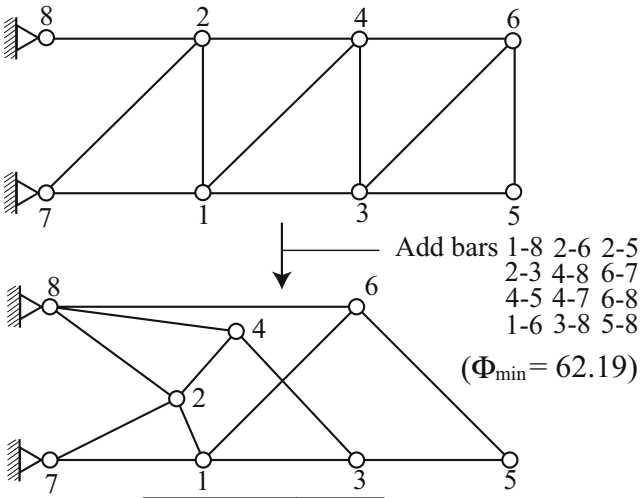


Node	x	y
2	1.1895	7.7654
4	1.8453	1.0000
6	2.7461	0.4949

$\Phi = 64.75$

Fig. 8 Fully connected ground structure of eight-node truss and its optimal solution for Example 3

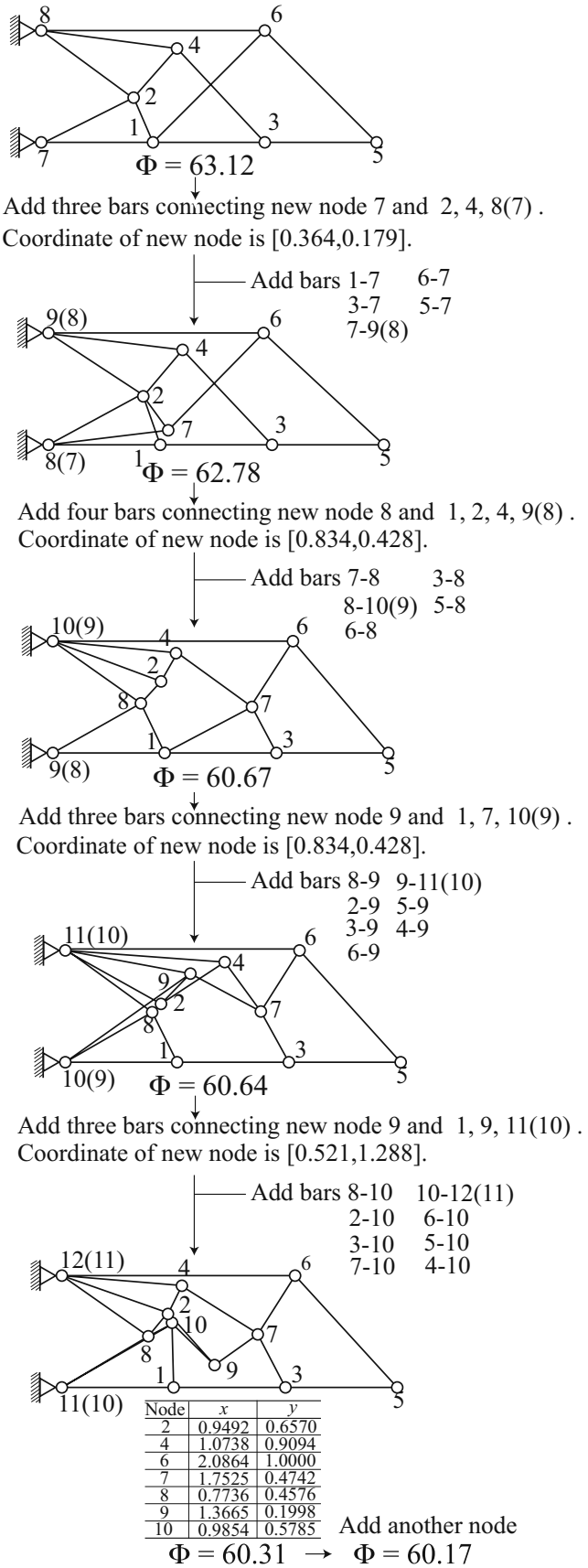
of the position of the nodes with variable coordinates in optimization, e.g., bars 2–6, 4–8, and 6–8 shown in Fig. 8. However, the overlapping bars whose end nodes



Node	x	y
2	0.8269	0.4000
4	1.2149	0.8395
6	2.0000	1.0000

$\Phi = 63.12$

Fig. 9 Optimization result by Algorithm 1 (GSTRG1) for Example 3



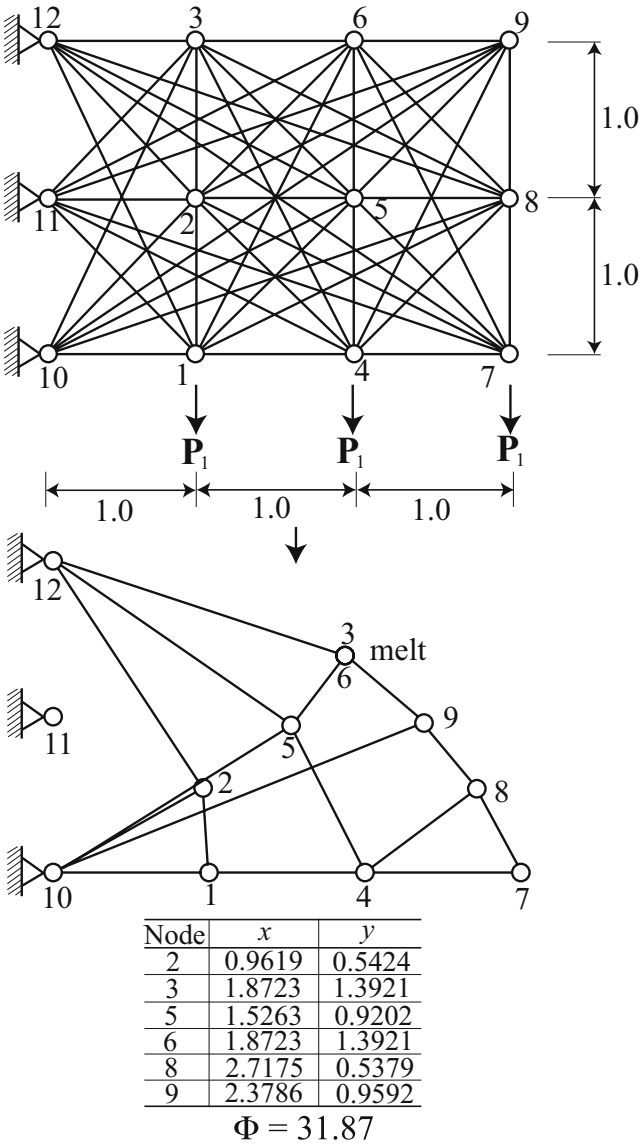
Node	x	y
2	0.9492	0.6570
4	1.0738	0.9094
6	2.0864	1.0000
7	1.7525	0.4742
8	0.7736	0.4576
9	1.3665	0.1998
10	0.9854	0.5785

Add another node

$\Phi = 60.31 \rightarrow \Phi = 60.17$

Fig. 10 Optimization result by Algorithm 2 (GSTRG1) for Example 3

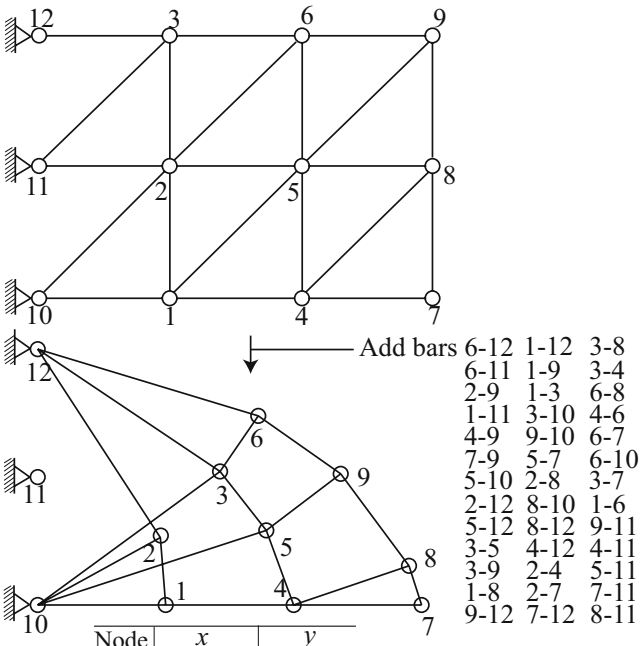
cannot move are excluded. We set the initial locations of nodes at all of the vertices of 3 × 1 grid. The single-load case is assumed with $\mathbf{P}_1$ values applied at nodes 1, 3 and 5 simultaneously, as indicated in Fig. 8. Application of the simultaneous topology and geometry optimization for the 24-bar truss generates the statically determinate 12-bar truss, which may not be global optimal truss with the objective function value $\Phi = 64.75$ as illustrated in Fig. 8, where the nodal coordinates of free nodes are also shown. For P2, the optimality is discussed with respect to the locations of nodes and the connectivity of the bar of a truss with specific numbers of free and fixed nodes. It is well-known that the global optimal truss with respect to the nodal coordinates is difficult to be found for P2. Hence, we do not use the term *global optimal* for Examples 3 and 4. Application of Algorithm 1 for this eight-node truss with the initial 12-bar sparse ground structure generates the statically determinate 12-bar truss with $\Phi = 63.12$ after adding 12 bars, which is illustrated in Fig. 9, where truss with $\Phi = 62.19$ is found after adding the 10th bar. Hence, the solution obtained from the densest ground structure is a local optimal solution.



Node	x	y
2	0.9619	0.5424
3	1.8723	1.3921
5	1.5263	0.9202
6	1.8723	1.3921
8	2.7175	0.5379
9	2.3786	0.9592

$\Phi = 31.87$

Fig. 11 Fully connected ground structure of 12-node truss and its optimal solution for Example 4

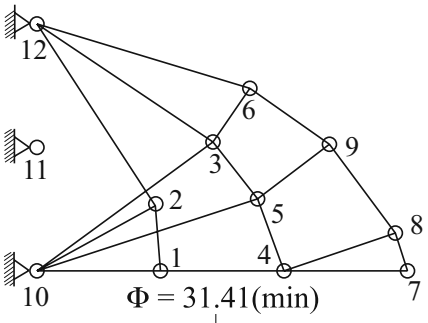


Node	x	y
2	0.9620	0.5427
3	1.4250	1.0439
5	1.7863	0.5832
6	1.7236	1.4785
8	2.9016	0.3063
9	2.3679	1.0254

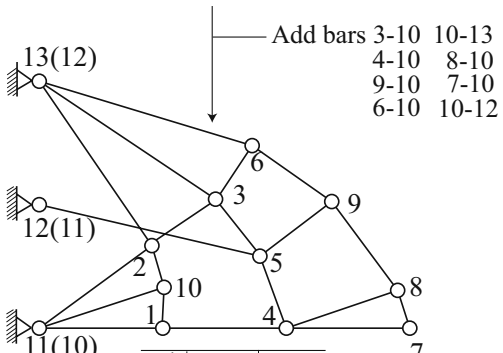
$\Phi = 31.41$(min)

Fig. 12 Optimization results by Algorithm 1 (GSTRG1) for Example 4

The results of simultaneous topology and geometry optimization by Algorithm 2 are summarized in Fig. 10 with the 12-bar sparse truss as the initial ground structure. The total number of nodes is limited to 13, namely five nodes are added after the number of bars has reached 24. The objective function value reduces to $\Phi = 60.67$ after the addition of two nodes. However, after that, the Φ values have not been reduced so greatly. Finally, the objective function value decreases to $\Phi = 60.17$, as shown in Fig. 10, after the addition of five nodes in all. The effect of *GSTRG5* is not so great. Moreover, it is seen from Fig. 10 that the effectiveness of the addition of nodes decreases as the number of nodes increases. It should be noted that the melting nodes have not occurred at any stage in the optimization, although the number of nodes has been



Add four bars connecting new node 10 and 1, 2, 5, 11(10) .
Coordinate of new node is [0.995,0.293].



Node	x	y
2	0.9120	0.6686
3	1.4250	1.0439
5	1.7863	0.5832
6	1.7236	1.4785
8	2.9016	0.3063
9	2.3679	1.0255
10	1.0106	0.3307

$\Phi = 31.33$ (No further improvement)

Fig. 13 Optimization results by Algorithm 2 (GSTRG1) for Example 4

increased and the node indicated by 6 in Fig. 10 has been located at the upper limit of the y-coordinate of the design domain.

6.4 Example 4: 60-bar truss P2

Figure 11 shows a 12-node truss for which we apply Algorithms 1 and 2, where the densest truss has 60 bars with several overlapping bars. The overlapping bars that cannot change the locations in the optimization are not included. We set the initial locations of nodes at all vertices of a 3×2 grid. The single load case is assumed with $\mathbf{P}_1$ values applied at nodes 1, 4 and 7, simultaneously as indicated in Fig. 11. Application of the simultaneous topology and geometry optimization for the 60-bar truss generates a statically determinate 16-bar truss, which may be a local optimal truss with $\Phi = 31.87$, as illustrated in Fig. 11. The melting nodes occurred with coincidence of nodes 3 and 6. Application of Algorithm 1 for this 12-node truss with the initial 21-bar sparse ground structure generates a statically determinate 18-bar truss with $\Phi = 31.41$, as shown in Fig. 12. The melting nodes did not occur for this optimization. Hence, the solution obtained using *GSM* with the densest truss is not global optimal truss. The simultaneous topology and geometry optimization results by Algorithm 2 are summarized in Fig. 13 with the initial 21-bar sparse ground structure. The total number of nodes is limited to 15, namely, three nodes are added after the number of bars has reached 60. Then, the objective function value is reduced to $\Phi = 31.33$ after adding a node. After that, two nodes are added; however, the objective function value has not improved with the formation of the melting nodes one by one. The benefit of applying strategy 5 for this example is less significant than for Example 3. It should be noted that there has been no node that is located at the limits of the design domain. There may be the *saturation* for the number of nodes, or degrees of freedom, for this problem, as pointed out in Mckeown (1998), where the addition of nodes at a later stage does not improve the solution.

6.5 Example 5: eight-bar truss P3

Figure 14a shows a five-node truss for which we apply Algorithm 3, where the densest truss has eight bars without overlapping bars. The two load cases are assumed with $\mathbf{P}_2$ values applied at nodes 3 and 5 separately, as indicated in Fig. 14a.

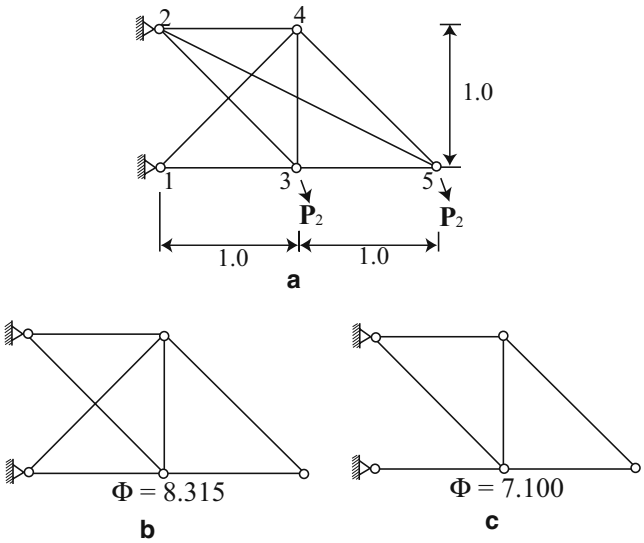


Fig. 14 Fully connected truss and other sparse trusses and their optimal solution for Example 5 (connectivities of ground structure and optimal truss are same for the sparse trusses). **a** Algorithm 3 applied for a five-node truss. **b** *GSM* applied for the densest eight-bar truss. **c** Six-bar truss

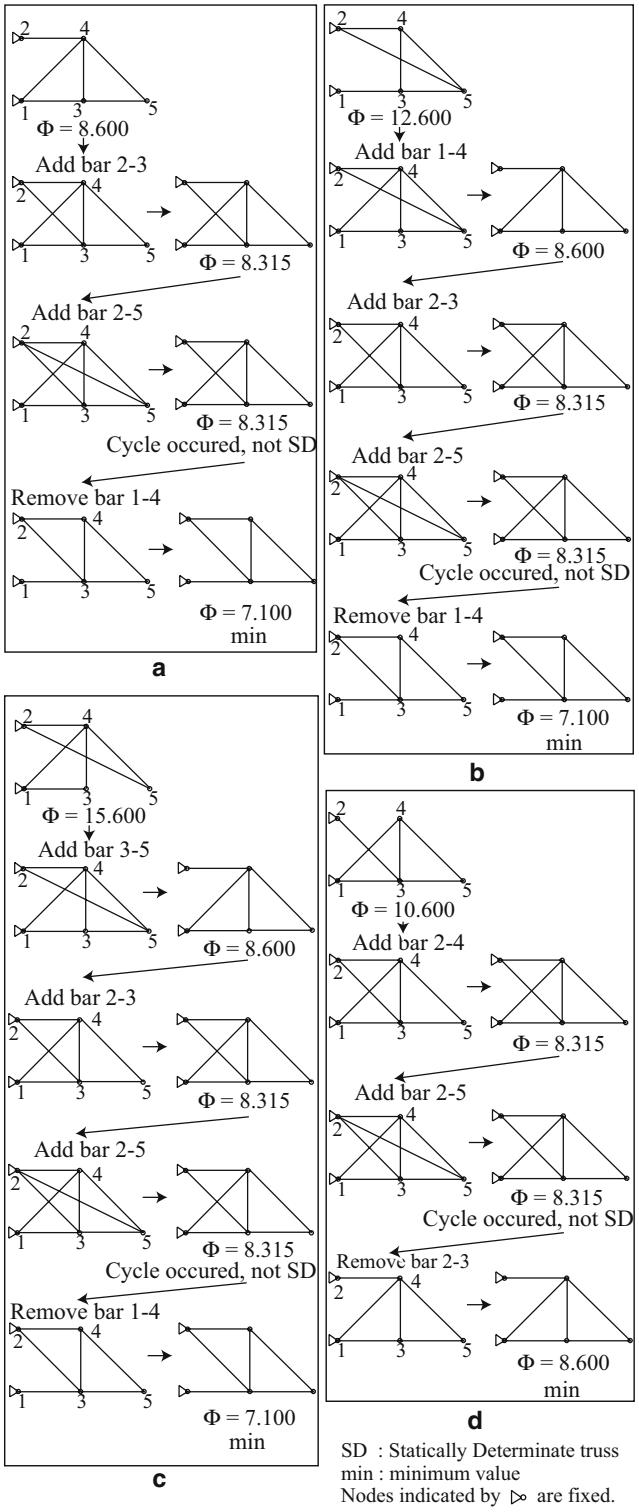


Fig. 15 Optimization results by Algorithm 3 for Example 5. **a–c** Statically determinate initial ground structure without bar 2–3. **d** Starting from the statically determinate truss with bar 2–3

The allowable stress for the bar 2–3 is σ_{II}, and that for other bars is σ_I. Application of *GSM* for the densest eight-bar truss generates a statically indeterminate

Table 1 The optimization results by the continuation method for Example 5

	γ	ε_0	Φ	Iterations
Case 1	0.7	20.0	8.315	35
Case 2	0.5	20.0	8.315	18
Case 3	0.3	20.0	7.100	11
Case 4	0.3	10.0	8.315	10
Case 5	0.3	5.0	8.315	9

seven-bar truss with $\Phi = 8.315$, as indicated in Fig. 14b. The same result is obtained by the ε-relaxation approach with $\varepsilon = 1.0^{-2}$. However, the *singular optimum solution* for this example is a six-bar truss with $\Phi = 7.100$, as shown in Fig. 14c, which was obtained by enumerating all the possible topologies. For P3, the term *optimal ground structure* is used to indicate the structure that can generate a singular optimum solution for Examples 5 and 6. The topology optimization results by Algorithm 3 are summarized in Fig. 15. *GSTRG4* with $\zeta_a = 0.1$ are used in Algorithm 3. We obtain the singular optimum solution and the optimal ground structure when the initial ground structure is statically determinate without bar 2–3, as shown in Fig. 15a–c. It should be noted that the topology of optimal ground structure and that of the optimized truss, or the singular optimum solution, are the same at the final stage. Hence, it can be said that the ground structures are successfully optimized without functionless bars. However, the singular optimum solution cannot be obtained if we start from the statically determinate truss with the bar 2–3 as shown in Fig. 15d.

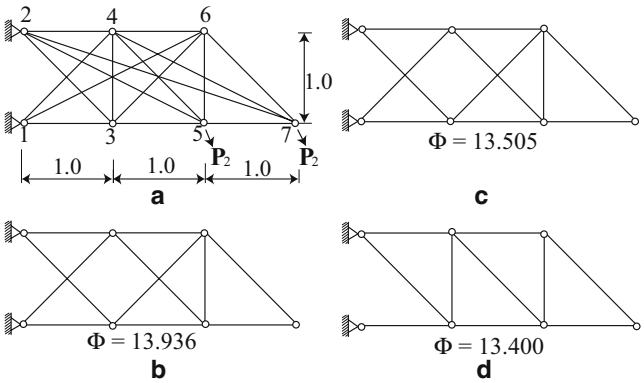


Fig. 16 Fully connected truss and other sparse trusses and their optimal solution for Example 5 (connectivities of ground structure and optimal truss are same for the sparse trusses). **a** Algorithm 3 applied for a seven-node truss. **b** Application of *GSM* for the densest 16-bar truss. **c** Application of *GSM* for the 11-bar truss. **d** 10-bar truss

For this example, P3 formulated in Section 3 needs to be solved four to five times in Algorithm 3 to obtain the singular optimum solution as shown in Fig. 15a–c. We applied the continuation method presented in Section 3 five times with the different values of ε_0 and γ to evaluate the efficiency of Algorithm 3. The optimization results for the five cases are summarized in Table 1.

It is seen from the table that case 3 gives the singular optimum solution; however, the relaxed NLP, or (P3$_\varepsilon$), needs to be solved 11 times. In addition, in other cases, they need many iterations, 9 to 35 iterations, to find a nearly optimal solution with $\Phi = 8.315$. Hence, Algorithm 3 is more effective than the continuation method for this example.

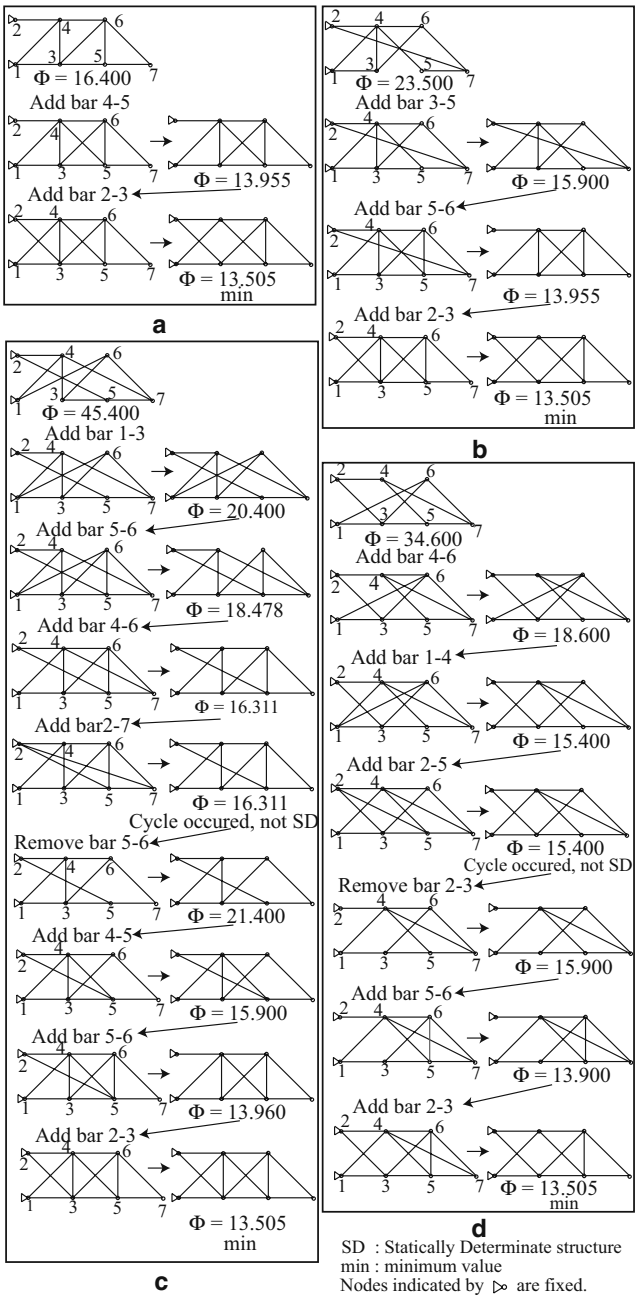


Fig. 17 Optimization results by Algorithm 3 for Example 6 (**a**–**d**)

Table 2 The optimization results by the continuation method for Example 6

	γ	ε_0	Φ	Iterations
Case 1	0.7	20.0	13.505	35
Case 2	0.5	20.0	13.505	18
Case 3	0.3	20.0	13.505	11
Case 4	0.3	10.0	13.505	10
Case 5	0.3	5.0	13.505	9

6.6 Example 6: 16-bar truss P3

Figure 16a shows a seven-node truss for which we apply Algorithm 3, where the densest truss has 16 bars without overlapping bars. The two load cases are assumed with $\mathbf{P}_2$ values applied at nodes 5 and 7 separately, as shown in Fig. 16a. The allowable stress for bars 2–3 and 4–5 is σ_{II}, and that for other bars is σ_I. Application of *GSM* for the densest 16-bar truss generates a statically indeterminate 11-bar truss with $\Phi = 13.936$ as indicated in Fig. 16b. Further application of *GSM* for the 11-bar truss generates a 11-bar truss with $\Phi = 13.505$ as illustrated in Fig. 16c. The same result is obtained by the ε-relaxation approach with $\varepsilon = 1.0^{-2}$. However, the singular optimum solution for this example is a 10-bar truss with $\Phi = 13.400$, as shown in Fig. 16d, which was obtained by enumerating all the possible topologies. The topology optimization results by Algorithm 3 are summarized in Fig. 17a–d. *GSTRG4* with $\zeta_a = 0.1$ is used in Algorithm 3. As is seen, we cannot obtain the singular optimum solution starting from the statically determinate initial ground structure. However, the nearly optimal solution with $\Phi = 13.505$ can be obtained starting from four arbitrarily chosen statically determinate trusses. Generally, it takes more iterations to reach the nearly optimal solution if we start from an irregular truss.

For this example, P3 formulated in Section 3 needs to be solved three to nine times in Algorithm 3 to obtain the near optimum solution with $\Phi = 13.505$ as shown in Fig. 17a–d. We applied the continuation method presented in Section 3 five times with the different values for ε_0 and γ to evaluate the efficiency of Algorithm 3. The optimization results for the five cases are summarized in Table 2.

It is seen from the table that all the cases give the nearly optimal solution with $\Phi = 13.505$; however, the relaxed NLP, or (P3$_\varepsilon$), needs to be solved several times,

9 to 35 times, to find the nearly optimal solution. Hence, Algorithm 3 is more effective to find a nearly optimal solution than the continuation method for this example.

7 Conclusions

Five strategies have been presented for addition and/or removal of bars and nodes for truss topology optimization problems under static loads. Based on these strategies, we proposed three algorithms of *GGSM* to optimize the ground structure as an extension of the *GSM* that utilizes only removal of the bars. The expansion and reduction of the ground structure are realized by the reformulation effectively implemented by C++ standard template library and BGL.

Numerical examples have been solved by *GGSM*. The effectiveness of *GGSM* is demonstrated for the topology optimization problems without singular optimum solutions. Next, singular optimum solutions have been found for a small truss. However, for a larger problem, *GGSM* could not find the singular optimum solution. *GGSM* should be modified to include other effective strategies to tackle this problem.

The advantages of applying *GGSM* are summarized as follows:

1. The processes of optimizing the ground structure can easily be seen with successive addition or removal of bars and nodes.
2. Multiple optimal solutions can be found for compliance minimization problem.
3. Optimal topology can be found from a sparse initial ground structure. This way, we can reduce the cost for constructing fully connected ground structure to find the global optimal solution.
4. Better solution can be found by modification of the connectivity in the ground structure, and the objective value can further be reduced with the addition of nodes.
5. Singular optimum solution, or nearly optimal solution, can be found for problems with stress constraints under multiple loading conditions.
6. Topology of optimized ground structure is identical to that of optimal truss at the final stage for problems with stress constraints under multiple loading conditions.

References

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

Achtziger W, Bendosoe M, Ben-Tal A, Zowe J (1992) Equivalent displacement based foumulation for maximum strength truss topology design. Impact Comput Sci Eng 4: 314–345

Bendsøe MP, Sigmund O (2003) Topology optimization. Springer, Berlin Heiderberg New York

Boost (2007) Boost C++ library http://boost.org/

Cheng GD, Guo X (1997) ε-relaxed approach. Struct Multidisc Optim 29:190–197

Dorn W, Gomory R, Greenberg H (1964) Automatic design of optimal structures. J Mec 3:25–52

Gill PE, Murray W, Saunders M (1997) User's guide for SNOPT version 7 software for large-scale nonlinear programming. Stanford Business Software, Mountain View

Kirsch U (1993) Fundamental properties of optimal topologies. In: Bendsoe MP (ed) Topology optimization of structures. Kluwer, Norwell, pp 3–18

Kirsch U (1996) Integration of reduction and expansion processes in layout optimization. Struct Optim 11:13–18

Kirsch U (2002) Design-oriented analysis of structures. Kluwer, Norwell

Martinez P, Marti P, Querin OM (2007) Growth method for size, topology, and geometry optimization of truss structures. Struct Multidisc Optim 33:13–26

Mckeown JJ (1998) Growing optimal pin-jointed frames. Struct Optim 15:92–100

Meyers S (2002) Effective STL. Addison Wesley, Reading

Ohsaki M (1998) On simultaneous optimization of topology and geometry of a regular plane truss. Comput Struct 79(6): 673–679

Ohsaki M (1998) Random search method based on exact reanalysis for topology optimization of trusses with discrete cross-sectional areas. Comput Struct 66(1):69–77

Ohsaki M, Katoh N (2005) Topology optimization of trusses with stress and local constraints on nodal stability and member intersection. Struct Multidisc Optim 29:190–197

Pedersen P (1972) On the optimal layout of multi-purpose trusses. Comput Struct 2:695–712

Reddy G, Cagan J (1994) An improved shape annealing algorithm for truss topology generation. ASME J Mech Eng 117:315–321

Rule WK (1994) Automatic design by optimized growth. ASCE J Struct Eng 120:3063–3070

Siek JG, Lee LQ, Lumsdaine A (2002) The boost graph library. C++ in-depth series. Addison Wesley, Reading

Stolpe M, Svanberg K (2001) On the trajectories of the epsilon-relaxation approach for stress-constrained truss topology optimization. Struct Multidisc Optim 21:140–151

Stolpe M, Svanberg K (2004) A stress-constrained truss-topology and material-selection problem that can be solved by linear programming. Struct Multidisc Optim 27:126–129

West DB (2001) Introduction to graph theory, 2nd edn. Prentice Hall, Englewood Cliffs