10th World Congress on Structural and Multidisciplinary Optimization
May 19 -24, 2013, Orlando, Florida, USA

A Design Method for Optimal Truss Structures with Certain Redundancy Based on Combinatorial Rigidity Theory

Rie Kohta[1], Makoto Yamakawa[2], Naoki Katoh[3], Yoshikazu Araki[4] and Makoto Ohsaki[5]

[1] Kyoto University, Kyoto, Japan, se.kohta@archi.kyoto-u.ac.jp
[2]Tokyo Denki University, Tokyo, Japan, myamakawa@mail.dendai.ac.jp
[3] Kyoto University, Kyoto, Japan, naoki@archi.kyoto-u.ac.jp
[4] Kyoto University, Kyoto, Japan, araki@archi.kyoto-u.ac.jp
[2] Hiroshima University, Hiroshima, Japan, ohsaki@hiroshima-u.ac.jp

Abstract
The Truss Topology Design (TTD) problem deals with the selection of optimal configuration for pin-jointed trusses, in particular, the optimization of the connectivity of the nodes by the members, in which volume and/or compliance are minimized. In general, it is known that such truss structures are statically determinate and not redundantly rigid, that is, if just one member is damaged or lost, the entire structure cannot support loads. Therefore, it is important to take redundancy of structures into consideration in the TTD. We present a new practical design method for finding a redundant TTD based on combinatorial rigidity theory. In this paper, we define, as redundancy, the margin of the number of members until the collapse of the entire structure when some components are damaged or lost. A truss structure is said to be a “2-edge-rigid truss” if we need to remove at least two members from the truss structure so that the structure becomes non-rigid. We can find a 2-edge-rigid TTD by using a method based on combinatorial rigidity theory. The present method enables to find an approximately optimal TTD with low computational cost. In numerical examples, we compute redundantly rigid truss structures, in which objective value of the solution is about one percent greater than that of the lower bound. Therefore, we can conclude that the method is effective to design an optimal redundant truss structure.
Keywords: Truss Topology Optimization, Redundancy, Combinatorial Rigidity Theory

1. Introduction

It is very important to take redundancy into consideration in structural design, because it helps us to design against the force beyond our estimation. Generally it is difficult to ensure the safety against such unknown future forces. In particular, there is a concern over “progressive collapse”, which is the collapse of the whole structure triggered by the partial damage. In order to overcome these dangers, we must consider redundancy of structural design. This paper deals with the Truss Topology Design (TTD) problem with redundancy. There is not many researches on redundancy in the TTD although truss structures designed by the TTD are generally statically determinate and do not have redundancy. In the following subsections, we provide some backgrounds for this study.

1.1.Truss Topology Design (TTD) problem

The TTD problem deals with the selection of optimal configuration for pin-jointed trusses, in particular, the optimization of the connectivity of the nodes by the members, in which volume and/or compliance are minimized. Ground structure method is known as the most popular method of TTD. In this method, a ground structure with a sufficiently large number of members is given as the initial configuration (as shown in Figure 1). Then, the optimal topology is obtained by solving the optimization problem in which design variables are the cross section areas of the members, and the members with no cross section area are removed. It is known that the optimal truss structure by the method is statically determinate [3]. Statically determinate truss structures are not redundantly rigid, that is, if just one member is damaged or lost, the entire structure cannot support loads.

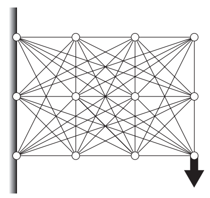


Figure 1: Ground structure

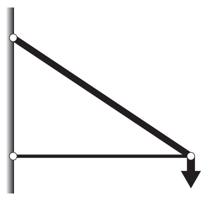


Figure 2: Optimal truss structure

1.2. Previous studies on redundancy of bar-and-joint framework

A truss structure is a kind of bar-and-joint framework, which is a structure composed of rigid rods (bars) connected at their ends (joints). These frameworks are denoted by "graphs" such as Figure 3. A graph is said to be "rigid" if it has no nontrivial deformations preserving the edge lengths. The rigidity of a graph has been studied in combinatorial rigidity theory, a field of discrete mathematics.

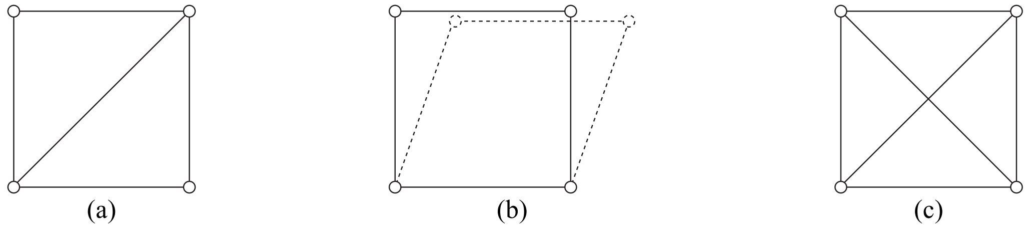


Figure 3: (a) A rigid graph. (b) A non-rigid graph. (c) A redundantly rigid graph

A. Garcia et al.(2011) provide an $O(n^2)$ algorithm to augment a non-redundant rigid graph to a "redundantly rigid" graph by adding a minimum number of edges [4], where a graph is redundantly rigid when the graph remains rigid even if any arbitrary edge is removed. C. Yu et al. (2008) propose the concept of "k-edge-rigid" graphs[7]. The *k*-edge-rigid graph is still rigid after deletion of any (k - 1) edge(s), while redundantly rigid graph is still rigid after deletion of any one of edges. The redundantly rigid truss corresponds to a 2-edge-rigid graph by this definition. The method to generate a *k*-edge-rigid graph is proposed by N. Katoh et al. (2012)[5].

1.3. Purpose of this paper

We propose a new practical design method for finding a redundant TTD based on combinatorial rigidity theory in this paper. Since a truss structure can be abstracted to a graph, the rigidity of truss structures can be studied by combinatorial rigidity theory. Some important concepts about combinatorial rigidity theory and the algorithm "find a minimal covering" proposed by Garcia [4] are presented in section 2. The formulation of our problem for finding a redundant TTD and its approximate method are given in section 3. Finally, some numerical examples are shown in section 4.

2. Combinatorial rigidity theory

In this section, we explain the outline of this theory and the key concept of this paper, redundant range $L(i,j)$.

2.1. Outline of combinatorial rigidity theory

Combinatorial rigidity theory studies the rigidity of bar-joint frameworks from the viewpoint of graph theory. In fact, a bar-joint framework can be viewed as a graph G=(V,E) embedded on the plane where V and E denote the set of vertices which correspond to joints and that of edges which correspond to members, respectively. Besides, $V(G)$ and $E(G)$ denote vertices and edges included in graph G, respectively.

The graph is called rigid in R^2, if the corresponding framework allows only trivial deformation, i.e., translation and rotation. In addition, a rigid graph with minimal number of edges is called a "Laman graph", and the necessary and sufficient condition for a graph G=(V,E) to be a Laman graph is given as follows:

$$|E| = 2|V| - 3, \quad (1.a)$$

$$|E(X)| \leq 2|X| - 3 \quad \text{for every } X \subseteq V \text{ with } 2 \leq |X| \leq |V|, \quad (1.b)$$

where $E(X) \subseteq E$ denotes the set of edges in the subgraph induced by vertices $X \subseteq V$, and $|E|$, $|V|$ are the number of edges and vertices, respectively. In addition, the notation $E(X)$ is often used to stand for a graph G'=(X,$E(X)$). For example, the graph illustrated in Figure 4(a) is a Laman graph: the equation (1.a) is satisfied with $|E|$=5 and $|V|$=4, and the subgraph $E(X)$ in Figure 4(b) satisfies the equation (1.b) with $|E(X)|$=3 and $|X|$=3. In the following, $\mathcal{L}$ denotes the set of Laman graphs.

(a) (b)

Figure 4: (a) A Laman graph. (b) A subgraph $E(X)$

Based on this theory, Garcia et al. provide an $O(n^2)$ algorithm to augment a Laman graph G to a 2-edge-rigid graph (a redundantly rigid graph), by adding a minimum number of edges [4].

(a) (b)

Figure 5: (a) A 2-edge-rigid graph. (b) A rigid graph

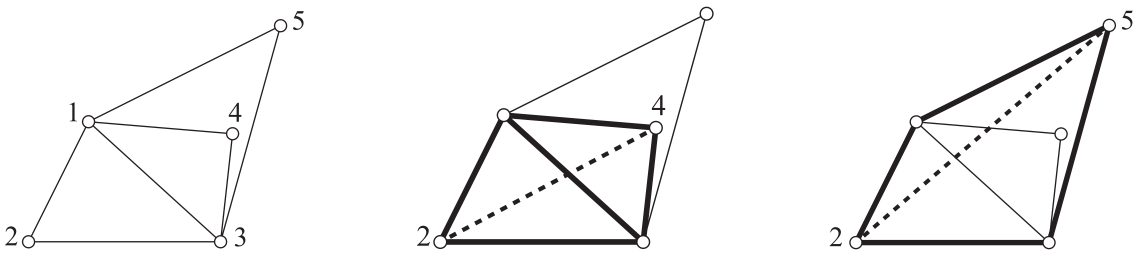
An edge e of a rigid graph G is “redundant” if $G - e$ is still rigid. A graph G is a “2-edge-rigid” if $G - e$ is rigid for any edge $e \in E(G)$ (see Figure 5(b)). In particular, a “generic circuit” is a graph G such that $G - e$ is a Laman graph for any edge $e \in E(G)$. Namely, G is a generic circuit if and only if

$$G - e \in \mathcal{L}, \ \forall \ e \in G. \tag{2}$$

Generic circuits can be said to be minimal 2-edge-rigid graphs. Since a truss structure can be abstracted to a graph, we can study its rigidity by combinatorial rigidity theory. In our study, we require that a truss is 2-edge-rigid. Then, we name this constraint as “2-edge-rigid constraint” in this study.

2.2. Redundant range $L(i,j)$

In the following, an edge connecting vertices i and j is denoted by $e = (i,j)$. When a new edge $e = (i,j)$ is added to a Laman graph L, the graph $L \cup (i,j)$ contains a unique generic circuit, which is denoted by $C(i,j)$ or $C(e)$. Since this generic circuit contains the edge $e = (i,j)$, the subgraph $C(i,j) - (i,j)$ is a Laman graph, which is called a “Laman subgraph”. Garcia et al. denote this Laman subgraph by $L(i,j)$ or $L(e)$ [4] . Figures 6(a), 6(b) and 6(c) illustrate the concept of $L(i,j)$. Each number in these figures represents the number of each vertex. In the Figure 6(b), the edges of the Laman subgraph $L(2,4)$ are illustrated in bold straight lines. If we add to this Laman graph an edge connecting the vertex "2" and the vertex "4", then the edges of $L(2,4)$ become redundant. In the same way, the edges of $L(2,5)$ illustrated in Figure 6(c) become redundant if we add the edge (2,5).



(a) (b) (c)

Figure 6: (a) A Laman graph L. (b) $L(2,4)$. (c) $L(2,5)$

Garcia et al. prove in their paper [4] that, when the edges $e_1, e_2, \ldots, e_k$ are added to a Laman graph L, a subset of edges of L will become redundant, which is denoted by $L(e_1, e_2, \ldots, e_k)$. Then this set coincides with $L(e_1) \cup L(e_2) \cup \ldots \cup L(e_k)$:

Lemma [4] *If L is a Laman graph, then* $L(e_1, e_2, \ldots, e_k) = L(e_1) \cup L(e_2) \cup \ldots \cup L(e_k)$.

As illustrated in Figures 6(b) and 6(c), if we add both edges (2,4) and (2,5), all edges of the Laman graph L become redundant. In other words, we can say that $L(2,4) \cup L(2,5) = L((2,4), (2,5))$ is covering the whole of L. In this way, every Laman graph can become a 2-edge-rigid graph by adding appropriate edges to the graph.

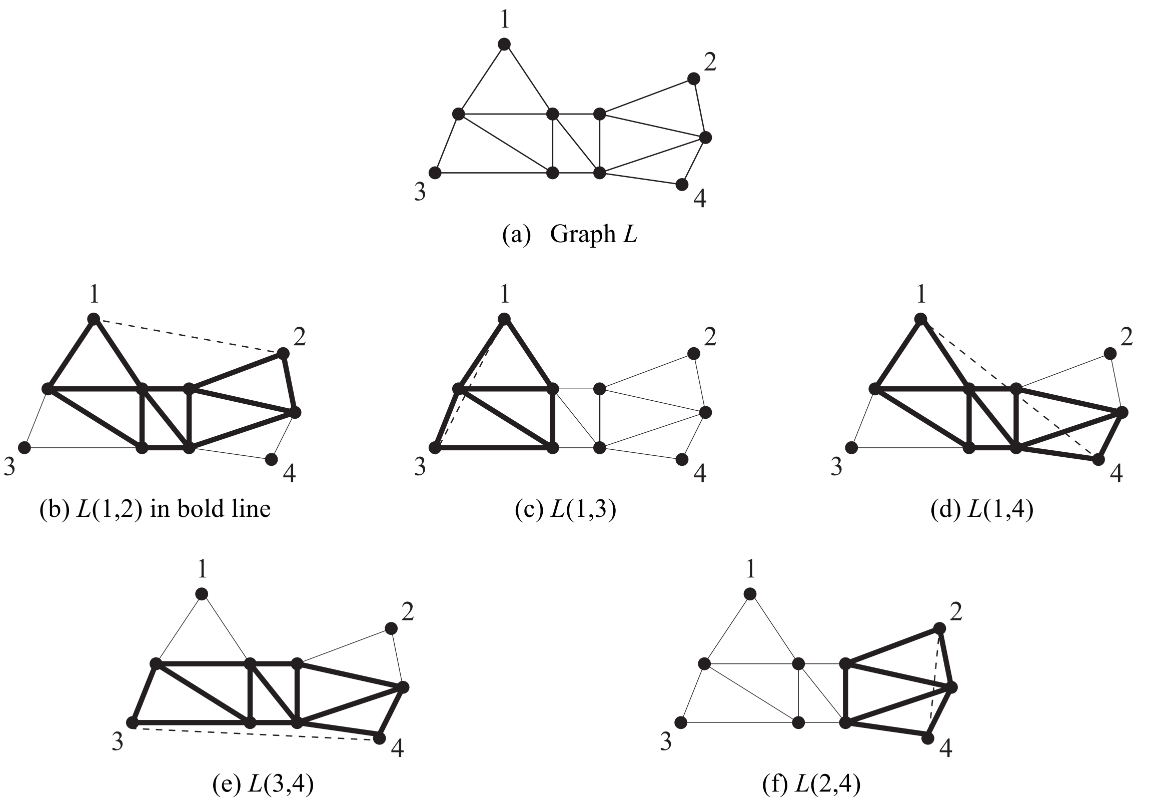
2.3. Algorithm for finding a minimal covering

Garcia proposed the algorithm called "Find a minimal covering" to calculate a set of edges, $E' = \{e_1, e_2,...,e_k\}$, with minimum cardinality, such that $L(e_1, e_2,...,e_k) = L(e_1) \cup L(e_2) \cup \ldots \cup L(e_k)$ covers all the edges of a given Laman graph L .

By using this algorithm, we can find a minimum number of edges we need to add. Degree of a vertex denotes the numbers of edges incident to the vertex. The algorithm "Find a minimal covering" first chooses a vertex i_1 of degree two or three which always exists, and activates the algorithm "Find a covering rooted in i_1", which finds vertices $j_2, j_3, \ldots, j_h$ such that $\bigcup_{k=2}^{h} L(i_1, j_k)$ covers all the edges of L, where a set of these vertices is denoted as $Extr(L)$. In the algorithm, if L has no vertices with degree two, we take a vertex k of degree three as i_1, find $Extr(L)$ using the algorithm "Find a covering rooted in i_1". After this, we rerun the algorithm by taking $j_2 \in Extr(L)$ as new i_1.

At this point, the number of edges added is $h-1$. The algorithm then further reduces $h-1$ edges to $[h/2]$ edges by repeatedly replacing consecutive three edges (i_1, j_l), (i_1, j_{l+1}), (i_1, j_{l+2}) with two edges based on the combinatorial observation in the procedure "Reduce a covering". In this procedure, we calculate $L(i_1, j_l)$ and $L(j_{l+1}, j_{l+2})$. Garcia showed that if $L(i_1, j_l) \cap L(j_{l+1}, j_{l+2}) \neq \emptyset$, then $L(i_1, j_l) \cup L(j_{l+1}, j_{l+2}) = L(i_1, j_l) \cup L(j_1, j_{l+1}) \cup L(j_1, j_{l+2})$, and otherwise $L(i_1, j_{l+1}) \cup L(j_l, j_{l+2}) = L(i_1, j_l) \cup L(j_1, j_{l+1}) \cup L(j_1, j_{l+2})$. For example, let us consider a Laman graph L illustrated in Figure 7(a) and the edges (1,2), (1,3), (1,4). As illustrated in Figure 7(b), (c) and (d), $L(1,2) \cup L(1,3) \cup L(1,4)$ equals L. The procedure "Reduce a covering" first calculates $L(1,2)$ and $L(3,4)$ as illustrated in Figure 7(b) and (e). In this case, as $L(1,2) \cap L(3,4)$ is not an empty set, $L(1,2) \cup L(3,4)$ equals $L(1,2) \cup L(1,3) \cup L(1,4)$, i.e. the edges (1,2), (1,3), (1,4) can be replaced by (1,2) and (3,4).

The flowchart of the algorithm "Find a minimal covering" is illustrated in Figure 8.



(a) Graph L

(b) $L(1,2)$ in bold line

(c) $L(1,3)$

(d) $L(1,4)$

(e) $L(3,4)$

(f) $L(2,4)$

Figure 7: Graph L and some $L(i,j)$

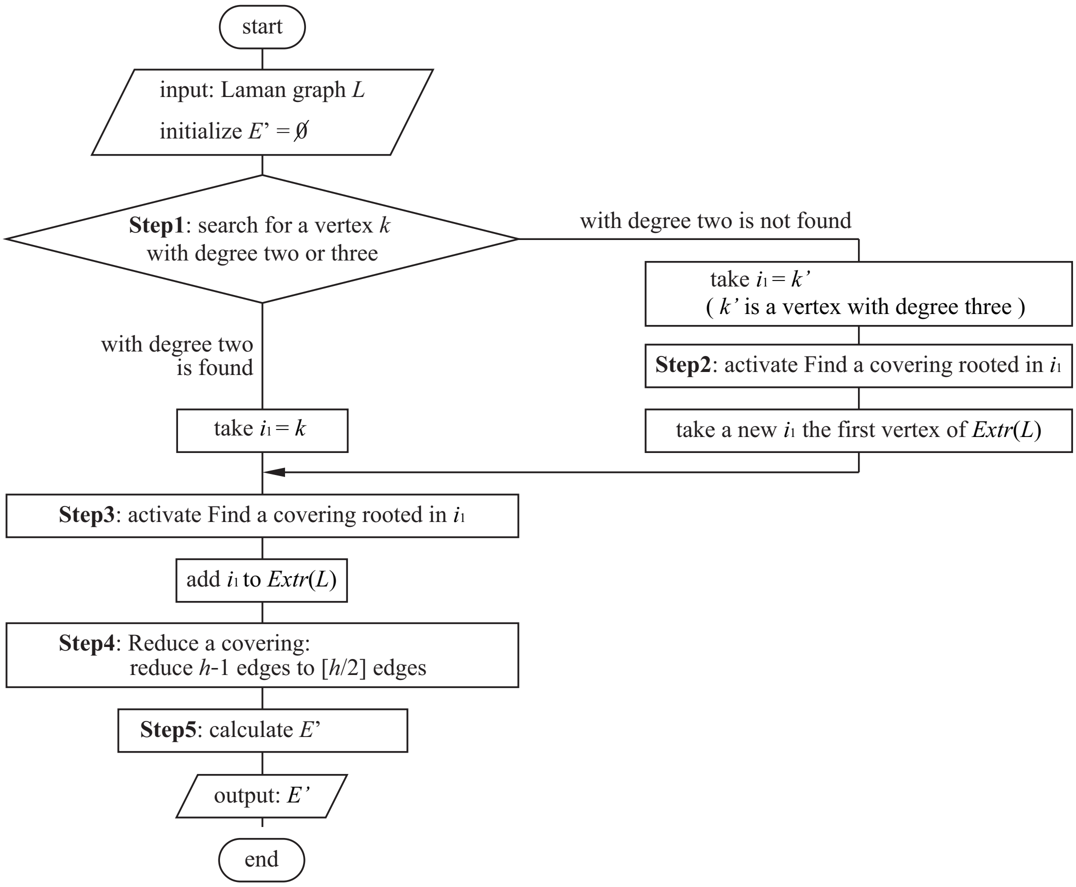


Figure 8: Algorithm Find a minimal covering

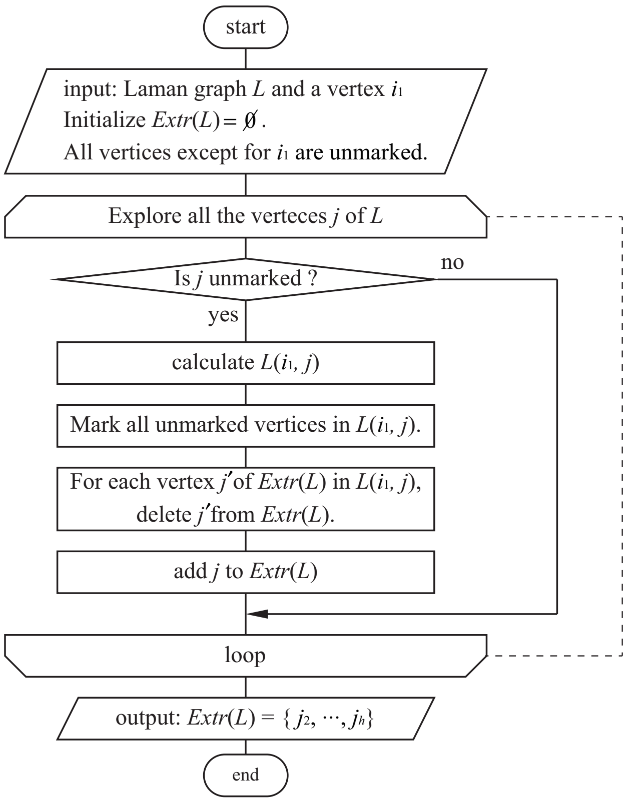


Figure 9: Algorithm Find a covering rooted in i_1

3. Formulation of the problem

3.1. Formulation of the original problem

In this subsection, the original problem and its formulation will be shown. First, we consider the ground structure like Figure 1: the total number of nodes is n and that of members is $m={}_nC_2$. The objective function to minimize is compliance, i.e., the work of the external force. We consider four constraints: the equilibrium of force, upper bound of the total volume, lower bound of the cross section area of each member and 2-edge-rigid constraint. This problem can be formulated as follows:

$$\text{Minimize} \quad \boldsymbol{P}^T\boldsymbol{U}, \tag{3.a}$$

$$\text{subject to} \quad \sum_{i\in I} A_i K_i \boldsymbol{U} = \boldsymbol{P}, \tag{3.b}$$

$$\sum_{i\in I} A_i L_i \le V^U, \tag{3.c}$$

$$A_i \ge A_i^L \ (i \in I), \tag{3.d}$$

$$I - e \in \mathcal{L}, \ \forall\, e \in I, \tag{3.e}$$

where $\boldsymbol{P}$ and $\boldsymbol{U}$ denote the vectors of external force and displacement of nodes, A_i, K_i, L_i and $A_i{}^L$ are respectively the cross section area, the stiffness matrix, the length and the lower bound of the cross section area of the member i, V^U denotes the upper bound of the total volume and I is a set of members in a topology. I can be referred to as the topology. Design variables are both the topology I and the set of cross section areas in the topology $\{ A_i \ge A_i^L \ (i \in I) \}$.

In general, it is difficult to find a 2-edge-rigid optimal truss structure for the problem (3). The 2-edge-rigid constraint is a combinatorial constraint. For finding the exact optimal solution, we may have to enumerate all possible topologies based on exact method. Because of its large calculation cost, it is difficult to solve large-scale problems by the method.

3.2. Approximation method for 2-edge-rigid TTD problem

As mentioned above, the TTD problem with 2-edge-rigid constraint is difficult to solve. From a practical point of view, however, we may not need to find an exact optimal solution of this problem. Thus, we find a 2-edge-rigid approximately optimal truss structure by using the method based on combinatorial rigidity theory.

The proposed method consists of three steps: (1) as a relaxation problem, we solve the TTD problem without 2-edge-rigid constraint (step1), after that, (2) we then find a 2-edge-rigid truss topology by adding a minimum number of members to the solution of step1 (step2), and finally, (3) under the truss topology of step2, we determine the optimal cross section areas of the members of the truss structure (step3). The solution of step1 corresponds to the lower bound of the objective value of the global optimal solution, and that of step3 corresponds to the upper bound of the objective value. Hence, we can also estimate the accuracy of the solution from the difference of objective values between the two solutions. In the following, each step is described in detail.

3.2.1. Step1 (lower bound of the original problem)

In this step, we solve problem (3) as a relaxation problem: we will not consider 2-edge-rigid constraint and replace lower bound constraints of the cross section areas of the members by non-negative constraints. Therefore the solution of this problem (4) is the lower bound of the original problem (3).

$$\text{Minimize} \quad \boldsymbol{P}^T\boldsymbol{U}, \tag{4.a}$$

$$\text{subject to} \quad \sum_{i\in I_g} A_i K_i \boldsymbol{U} = \boldsymbol{P}, \tag{4.b}$$

$$\sum_{i\in I_g} A_i L_i \le V^U, \tag{4.c}$$

$$A_i \ge 0 \ (i \in I_g), \tag{4.d}$$

where I_g denotes a set of members in a ground structure. Note that the number of elements of I_g may be extremely large depending on the size of the problem. Problem (4) is a convex programming problem and the global optimal solution can be found by the appropriate way (see Appendix). In general, it is known that the optimal solution of (4) is a statically determinate truss structure under single loading cases [3].

3.2.2. Step2 (augmenting the rigidity)

The optimal truss structure in step1 is a statically determinate truss and not redundantly rigid. Then in this step2, we make a 2-edge-rigid truss topology by adding minimum number of edges to the topology of the optimal truss of step1. We use the algorithm **Find a minimal covering** illustrated in Subsection 3.3 with a slight modification. Since the original algorithm does not take the lengths of edges into consideration, there is a possibility that long edges are added in the original algorithm. This is undesirable in terms of the increase of the total volume by adding new edges. In the algorithm **Find a covering rooted in** i_1, the order of exploring all the vertices j of L is not be specified. Therefore, if there are two edges (i_1, j_s) and (i_1, j_t) such that $L(i_1, j_s)=L(i_1, j_t)$ as illustrated in Figure 10(b) and 10(c), depending on the order of exploration, either j_s or j_t will be selected as $Extr(L)$. When the vertex j_s is explored before j_t, j_s will be included in $Extr(L)$, and vice versa. Then, we modify this order of exploration: the shorter the length of an edge (i_1, j) is, the higher priority is given to the vertex to be explored than other vertices.

By this modification, the shorter edge is selected as E', especially in the case of $|E'|=1$, the shortest edge will be picked. In the case $|E'|\geq 2$, however, it is not sure that the sum of the lengths of edges is minimum. Here we just indicate that getting a set of shortest edges E' is possible by applying the algorithm with "tight vertex set" presented by N. Katoh [5]. For more details, readers can refer to [5].

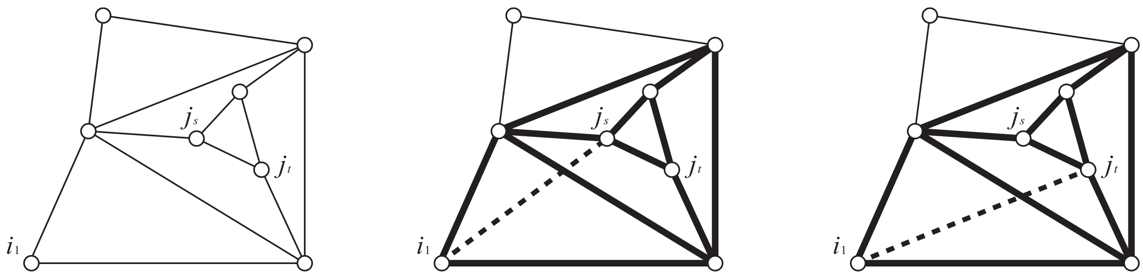


Figure 10: (a) A Laman graph L. (b) $L(i_1, j_s)$. (c) $L(i_1, j_t)$

3.2.3. Step3 (upper bound of the original problem)

In this step, we determine the optimal cross section areas of the members of the truss structure under the truss topology of step2. This problem is formulated as follows:

$$\text{Minimize} \quad \boldsymbol{P}^T\boldsymbol{U}, \tag{5.a}$$

$$\text{subject to} \quad \sum_{i\in I_2} A_i K_i \boldsymbol{U} = \boldsymbol{P}, \tag{5.b}$$

$$\sum_{i\in I_2} A_i L_i \leq V^U, \tag{5.c}$$

$$A_i \geq A_i^L = A^L \ (i \in I_2), \tag{5.d}$$

where I_2 denotes the set of members in the truss topology of step2 and all lower bounds of the cross section areas of the members equal A^L for brevity. The optimal value of Problem (5) is the upper bound of the one of Problem (3), because the optimal solution of (5) satisfies all of the four constraints: the equilibrium of force, upper bound of the total volume, lower bound of the cross section area of each member and 2-edge-rigid constraint.

4. Numerical Example

The algorithm described in the previous section has been thoroughly examined in some numerical tests. We will show the results bellow.

4.1. case 1

In this example, we consider a cantilever beam illustrated in Figure 11. The left side is pin-jointed to the wall and the vertical force of 10 [kN] is loaded at the lower right point. We give 10×20 vertices and 19,900 members that connect all pairs of vertices as the ground structure shown in Figure 12. The upper bound of the total volume V^U is $1.0\,\text{m}^3$, and in Step3, the lower bounds of the cross section areas of all members, A_i^L $(i \in I_2)$ are 25 cm^2.

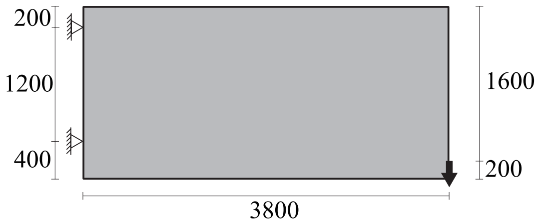


Figure 11: Design area (case 1)

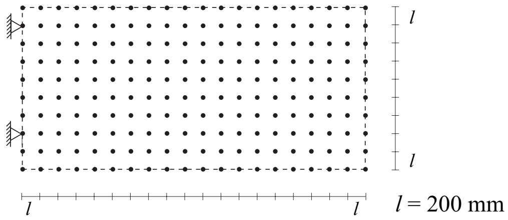


Figure 12: Ground structure (case 1)

The followings are the solutions of each step.

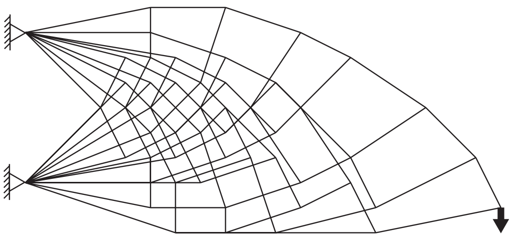


Figure 13: Solution of Step1 (case 1)

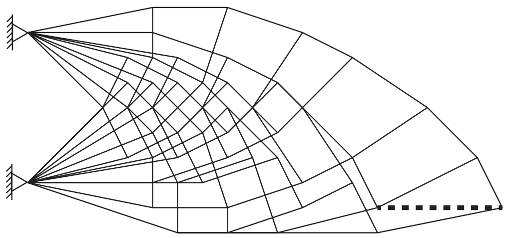


Figure 14: Solution of Step2 (case 1)

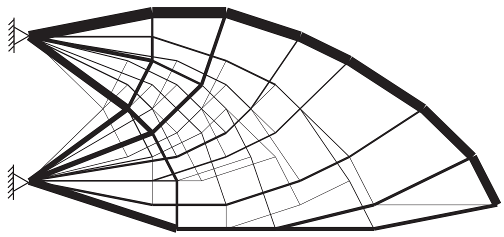


Figure 15: Solution of Step3 (case 1)

In Figure 14, the dotted line is the member added in Step2. By adding this new member, the truss structure becomes 2-edge-rigid truss, in other words, it is still rigid even if any one of the members is damaged or lost.

The width of lines shown in Figure 15 represents the cross section area of each member. The maximum cross section area is about 28 times as large as the minimum.

The objective values of Step1 and Step3, denoted by f_1 and f_3, are the lower and upper bounds of the objective value of the original problem (3), respectively. We achieve the result that f_3/f_1 equals 1.0108, that is, the upper bound is about just one percent greater than the lower bound. As a result, it can be said that the optimal truss structure of Step3 is an approximate solution close to the optimum.

In the above example, the difference between the lower and upper bounds is about one percent. However, the level of approximation depends on the lower bound value of cross section areas, A_L in (5.d). Figure 16 shows the relation between f_3/f_1 and A_L. It is obvious that f_3/f_1 is a monotonically increasing function of A_L.

Although it is natural to determine the value of A_L by the level of approximation so that the stress of all members does not exceed an appropriate value, it is our future issue to determine the value of A_L by considering the stress constraints.

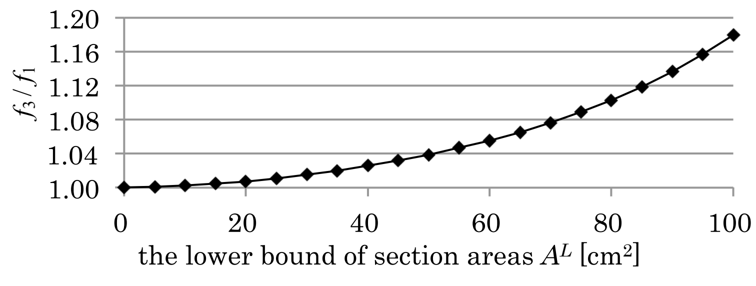


Figure 16: Relation between and A_L

4.2. cases 2 and 3 (multiple loads)

We also consider the TTD problems under multiple loads, such as the figures illustrated in Figures 17 and 18. The left side is pin-jointed to the wall and the vertical forces of 10 [kN] are loaded at the points indicated as downward arrows. The lower bounds of the cross section areas of all members, A_i^L ($i \in I_2$) are 25 cm^2, and the upper bound of the total volume V^U is 6.7×10^{-1} m^3.

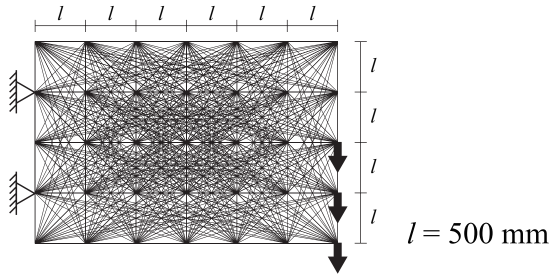


Figure 17: Ground structure (case 2)

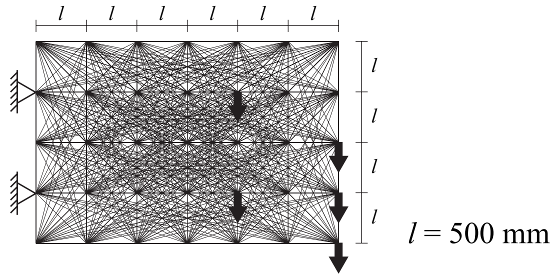


Figure 18: Ground structure (case 3)

The followings are the solutions of each case.

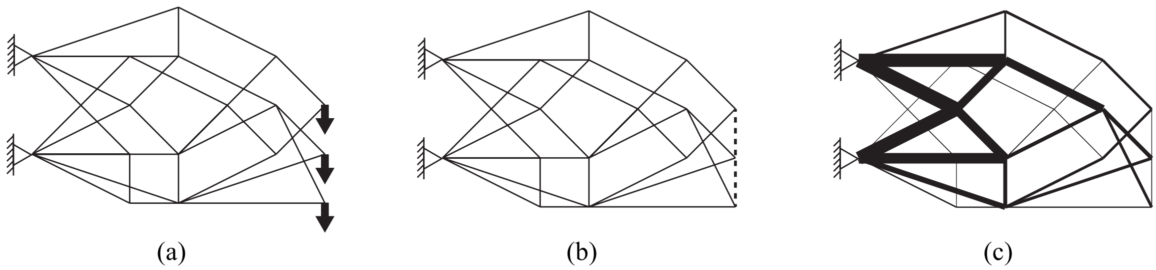


Figure 19: Solutions in case 2 of (a) Step1, (b) Step2, (c) Step3

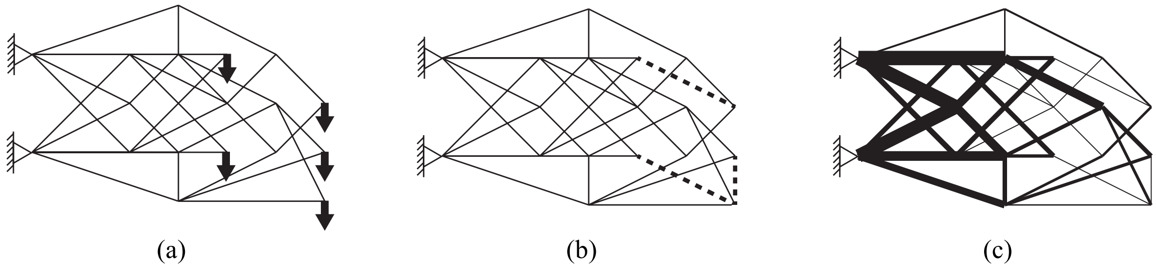


Figure 20: Solutions in case 3 of (a) Step1, (b) Step2, (c) Step3

As illustrated in Figures 19(b) and 20(b), we can achieve the 2-edge-rigid topologies in Step2 by adding two members in case 2 and three members in case 3, respectively. In each case, the maximum cross section areas of the Step3 are 6.61×10^{2} cm^2 and 4.94×10^{2} cm^2, and they are about 26 times and 20 times as large as the minimum cross section areas, respectively. Regarding the objective values, we achieve the result that f_3/f_1 equals 1.0045 in case 2 and 1.0088 in case 3, respectively.

5. Conclusion

In this study, we achieve the following conclusions.

1. We present a new practical design method based on combinatorial rigidity theory for finding an optimal truss structure with redundancy, in which the truss structure is still rigid even if any one of members is damaged or lost.
2. By this method, we can achieve upper and lower bound solutions for global optimal solution. In addition, its calculation cost is $O(n^2)$, while that of the method based on enumeration is $O(2^n)$.
3. In the numerical example of the ground structure with 200 vertices and 19,900 members, it is shown that the upper bound is about one percent greater than the lower bound, though the level of relaxation depends on the value of the lower bound, A_L.

Appendix

Problem (3) is a convex programming problem and its decision variables are cross section areas $A_i\,(i \in I_g)$ and displacements U. This problem can be reformulated to a linear programming problem as follows [3,6]:

$$\text{Minimize} \qquad -\boldsymbol{P}^T \tilde{\boldsymbol{U}}, \qquad (6.a)$$

$$\text{subject to} \qquad -1 \leq \sqrt{\frac{V^U E}{2}}\frac{\boldsymbol{b}_i^T \tilde{\boldsymbol{U}}}{L_i} \leq 1 \qquad (i \in I_g), \qquad (6.b)$$

where $\boldsymbol{b}_i$ denotes the unit orientation vector of the member i and $\tilde{\boldsymbol{U}}$ is the scaled value of $\boldsymbol{U}$ as $\tilde{\boldsymbol{U}} = (-f^* / 2)\,\boldsymbol{U}$; ($f^*$ is the objective value at the optimal solution of (6)).

Acknowledgment

We would like to acknowledge the support of The Kyoto University Foundation.

References

[1] Y.Kanno and Y.Ben-Haim, Redundancy and Robustness, or When Is Redundancy Redundant?, *Journal of Structural Engineering*, 137, 935-945, 2011.

[2] M.Ohsaki, N.Katoh, T.Kinoshita, S.Tanigawa, D.Avis and I.Streinu, Enumeration of optimal pin-jointed bistable compliant mechanisms with non-crossing members, *Structural and Multidisciplinary Optimization*, 37, 645-651, 2009

[3] M.Ohsaki, Optimization of Finite Dimensional Structures, CRC Press, 2010.

[4] A.Garcia and J.Tejel, Augmenting the Rigidity of a Graph in R^2, *Algorithmica*, 59, 145-168, 2011.

[5] Y.Yoshinaka, N.Katoh and N.Kamiyama, Generating Redundantly Rigid Frameworks in 2-dimension, Asian Association for Algorithms and Computation, 2012

[6] Aharon Ben-Tal, Arkadi Nemirovski, Lectures on Modern Convex Optimization, Society for Industrial and Applied Mathematics, 1987

[7] C.Yu and B.D.O.Anderson, Agent and Link Redundancy for Autonomous Formations, 17th IFAC World Congress, 6584-6589, 2008