TITLE:

FDMopt: Force density method for optimal geometry and topology of trusses

AUTHOR(S):

Hayashi, Kazuki; Ohsaki, Makoto

CITATION:

Hayashi, Kazuki ...[et al]. FDMopt: Force density method for optimal geometry and topology of trusses. Advances in Engineering Software 2019, 133: 12-19

ISSUE DATE:

2019-07

URL:

http://hdl.handle.net/2433/244332

RIGHT:

© 2019. This manuscript version is made available under the CC-BY-NC-ND 4.0 license http://creativecommons.org/licenses/by-nc-nd/4.0/.; The full-text file will be made open to the public on 1 July 2021 in accordance with publisher's 'Terms and Conditions for Self-Archiving'; This is not the published version. Please cite only the published version.; この論文は出版社版でありません。引用の際には出版社版をご確認ご利用ください。

京都大学学術情報リポジトリ
KURENAI 紅
Kyoto University Research Information Repository

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

Submitted to Advances in Engineering Software

FDMopt: force density method for optimal geometry and topology of trusses

Kazuki Hayashi[a], Makoto Ohsaki[a]

[a]*Department of Architecture and Architectural Engineering, Graduate School of Engineering, Kyoto University, Kyoto, Japan*

Abstract

This paper presents a new efficient tool for simultaneous optimization of topology and geometry of truss structures. Force density method is applied to formulate optimization problem to minimize compliance under constraint on total structural volume, and objective and constraint functions are expressed as explicit functions of force density only. This method does not need constraints on nodal locations to avoid coalescent nodes, and enables to generate optimal solutions with a variety in topology and geometry. Furthermore, for the purpose of controlling optimal shapes, tensor product Bézier surface is introduced as a design surface. The optimization problem is solved using sensitivity coefficients and the optimizer is compiled as a component compatible with Grasshopper, an algorithmic modeling plug-in for Rhinoceros, which is a popular 3D modeling software. Efficiency and accuracy of the proposed method are demonstrated through two numerical examples of semi-cylindrical and semi-spherical models.

Keywords: Truss, Latticed shell, Force density method, Simultaneous optimization of topology and geometry, Tensor product Bézier surface, Grasshopper, Interactive design

1. Introduction

There are three major categories in the field of truss optimization. First one is size optimization, where cross-sectional areas of members are considered to be design variables. From 1960s, size optimization is extensively studied for optimum design under stress constraints, which is called *fully stressed design* [1], where the stress of any member is equal to its upper or lower bound for at least one of the specified loading conditions.

Although size optimization has become popular in practical applications, it is difficult to find unexpected solutions, since the truss configuration is fixed and only cross-sectional areas are optimized.

Second category is topology optimization, which controls the connectivity of members. Topology optimization is also a well established field of research and theoretical works are summarized in, e.g., Refs. [2] and [3]. Above all, *ground structure method* (GSM) is widely used for topology optimization; it starts from a highly connected structure called *ground structure* and eliminate unnecessary members [4].

The last category is geometry optimization, which is also called shape optimization, and controls nodal locations to change overall truss geometry. Although numerous mathematical programming approaches have been utilized for optimizing geometry [5, 6], they inevitably need constraints on nodal locations to prevent numerical difficulty due to existence of extremely short members, called *melting nodes* [7] or *coalescent nodes* [8]. Therefore, there is little possibility to obtain a sparse optimal topology by simply setting nodal coordinates as design variables. We omit the description of size optimization, since both processes of topology and geometry optimization usually includes size optimization [9].

To obtain optimal topology and geometry of trusses simultaneously, *growth method* is one of the well studied approaches [10, 11]. It starts from a relatively sparse set of nodes and members, and add them by heuristics. Although growth method yields an optimal solution of sparse topology and geometry, the addition of nodes and members does not satisfy any theoretically defined optimality criterion. By contrast, it needs substantial computational cost if the GSM is applied to obtain an optimal geometry with acceptable precision, because nodal locations are fixed in this method; and accordingly, a large number of nodes and members are needed to optimize nodal locations.

Another approach is a hybrid method of the three

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

types of optimization methods above. The frequently applied method is an alternating optimization of topology and geometry; however such two-level algorithms need too much computational cost to optimize large scale trusses [9]. In addition, quality of the result is hard to evaluate, since alternating optimization easily converges to a non-optimal solution.

Topology optimization and geometry optimization can be simultaneously conducted by setting cross-sectional areas of members and nodal coordinates as design variables in a single optimization problem. This is called *simultaneous optimization* of topology and geometry, which is very difficult to solve, because it is necessary to modify the topology by removing coalescent nodes while varying the nodal locations. Achtziger [8] presented a simultaneous optimization method based on implicit programming. Although it always converges to an optimal solution because a mathematical programming approach is used, side constraints are needed to avoid melting nodes.

Latticed shells are one of the large scale structures composed of a number of truss members. In determining shapes and topologies of latticed shells, not only the designer's preference but also mechanical properties of the members play an important role. Thus, its design problem is formulated as a multiobjective optimization problem considering geometrical and mechanical properties, and parametric surfaces are often used to design latticed shells with non-standard shapes [9]. Ramm *et al.* [12] used Bézier surfaces for geometry optimization of shells. Ohsaki *et al.* [13] formulated an optimization problem for a double-layer space truss to minimize compliace under constraints on total structural volume, where triangular Bézier patch is used to define the geometry of the upper layer surface.

Owing to a large number of researches in structural optimization, there is an increasing number of practical optimization tools available to designers and engineers. Especially, structural optimization in Grasshopper [14] is widely used among architects and structural engineers. Grasshopper is an algorithmic modeling plug-in for Rhinoceros, which is a 3D modeling software [15]. As a general approach to optimizing structures within the framework of Grasshopper, the users usually combine components of structural analysis and optimization independently. Therefore, evolutionary optimization tools such as Galapagos [16] and Goat [17] are frequently used, because they are applicable to most of the optimization problems.

According to Bradner *et al.* [18], an optimal solution obtained by an optimization tool is often used as the starting point for design exploration, not the end product. Thus, it is important that the optimizer generates diverse optimal solutions, and simultaneous optimization of topology and geometry has potential to present diverse candidate designs.

However, because of the difficulties mentioned above, there is no practical tool to simultaneously optimize topology and geometry of trusses. Even the optimization process is somehow constructed within Grasshopper by combining a structural analyzer and an evolutionary optimizer, there is little possibility to obtain feasible solutions because the complexity of the problem is difficult to resolve in a simple manner, and the solutions do not always satisfy any optimality criterion. Therefore, it is necessary to develop a "package" of the framework to conduct the simultaneous optimization.

This paper aims to develop a Grasshopper component for simultaneous optimization of topology and geometry of trusses based on the force density method (FDM) proposed by Ohsaki and Hayashi [19]. The optimization problem is formulated as functions of force densities only; therefore, computational cost can be drastically reduced compared with previous methods where nodal coordinates and cross-sectional areas are assigned as design variables. Moreover, numerical difficulties due to melting nodes are successfully avoided by simply setting upper bounds for design variables, and thus various optimal solutions of topology and geometry are generated from a relatively sparse initial ground structure. We further introduce a free-form design surface to control nodal locations. Once a design surface is specified by an architectural designer, the optimizer moves nodes along and on the surface.

The paper is organized as follows. In section 2, we summarize formulation of the optimization problem. In section 3, compilation of the Grasshopper component which "packages" the optimization problem is outlined. The overall workflow to optimize the model is also described in this section. In section 4, two numerical examples are demonstrated to evaluate the optimizer's solutions in terms of their feasibility. In section 5, we close the paper with concluding remarks.

2. Optimization problem

We focus on a minimization problem of compliance under constraint on total structural volume and on nodal locations such that they are on a prescribed tensor product Bézier surface. Note that all the following variables are described as functions of force density only.

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

2.1. Force density method

Outline of the FDM for simultaneous optimization is presented here for completeness of the paper. See Ohsaki and Hayashi [19] for details. Free nodal coordinates are formulated as functions of force density. Force density of member i is defined with respect to the axial force N_i and the length L_i as

$$q_i = \frac{N_i}{L_i} \tag{1}$$

Consider a truss with m members and n nodes. If member i connects nodes j and k, then components of connectivity matrix $\mathbf{C} \in \mathbb{R}^{m\times n}$ are defined as

$$C_{ij} = -1,\ C_{ik} = 1\ (i = 1, ..., m) \tag{2}$$

and the remaining components are 0. Using $\mathbf{C}$ and the force density vector $\mathbf{q} \in \mathbb{R}^m$, the force density matrix $\mathbf{Q} \in \mathbb{R}^{n\times n}$ is defined as

$$\mathbf{Q} = \mathbf{C}^{\mathrm{T}}\,\mathrm{diag}(\mathbf{q})\mathbf{C} \tag{3}$$

The same matrix $\mathbf{Q}$ can be obtained for components of force densities in x-, y-, and z- directions, because the ratios of axial force to member length are all the same. $\mathbf{Q}$ is re-assembled into $\tilde{\mathbf{Q}} \in \mathbb{R}^{3n\times 3n}$ so that the components of free coordinates precede those of fixed coordinates as

$$\tilde{\mathbf{Q}} = \left(\begin{array}{ccc:ccc} \mathbf{Q}^x_{\mathrm{free}} & \mathbf{0} & \mathbf{0} & \mathbf{Q}^x_{\mathrm{link}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^y_{\mathrm{free}} & \mathbf{0} & \mathbf{0} & \mathbf{Q}^y_{\mathrm{link}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}^z_{\mathrm{free}} & \mathbf{0} & \mathbf{0} & \mathbf{Q}^z_{\mathrm{link}} \\ \hdashline \mathbf{Q}^{x\mathrm{T}}_{\mathrm{link}} & \mathbf{0} & \mathbf{0} & \mathbf{Q}^x_{\mathrm{fix}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^{y\mathrm{T}}_{\mathrm{link}} & \mathbf{0} & \mathbf{0} & \mathbf{Q}^y_{\mathrm{fix}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}^{z\mathrm{T}}_{\mathrm{link}} & \mathbf{0} & \mathbf{0} & \mathbf{Q}^z_{\mathrm{fix}} \end{array}\right) \tag{4}$$

Note that we use the term *fixed node* to express nodes that do not move in the optimization process. Therefore, fixed nodes include not only pin-supported but also loaded ones. Let n_{free} and n_{fix} denote the numbers of free and fixed DOFs satisfying

$$n_{\mathrm{free}} + n_{\mathrm{fix}} = 3n \tag{5}$$

Then, matrices $(\mathbf{Q}^x_{\mathrm{free}}, \mathbf{Q}^y_{\mathrm{free}}, \mathbf{Q}^z_{\mathrm{free}})$, $(\mathbf{Q}^x_{\mathrm{link}}, \mathbf{Q}^y_{\mathrm{link}}, \mathbf{Q}^z_{\mathrm{link}})$, and $(\mathbf{Q}^x_{\mathrm{fix}}, \mathbf{Q}^y_{\mathrm{fix}}, \mathbf{Q}^z_{\mathrm{fix}})$ are combined to $\tilde{\mathbf{Q}}_{\mathrm{free}} \in \mathbb{R}^{n_{\mathrm{free}}\times n_{\mathrm{free}}}$, $\tilde{\mathbf{Q}}_{\mathrm{link}} \in \mathbb{R}^{n_{\mathrm{free}}\times n_{\mathrm{fix}}}$, and $\tilde{\mathbf{Q}}_{\mathrm{fix}} \in \mathbb{R}^{n_{\mathrm{fix}}\times n_{\mathrm{fix}}}$, respectively.

If the force densities of all members and fixed nodal coordinates $\mathbf{X}_{\mathrm{fix}} \in \mathbb{R}^{n_{\mathrm{fix}}}$ are specified, then the free nodal coordinates $\mathbf{X}_{\mathrm{free}} \in \mathbb{R}^{n_{\mathrm{free}}}$ are obtained from the following system of linear equations:

$$\tilde{\mathbf{Q}}_{\mathrm{free}}\mathbf{X}_{\mathrm{free}} = -\tilde{\mathbf{Q}}_{\mathrm{link}}\mathbf{X}_{\mathrm{fix}} \tag{6}$$

Therefore, $\mathbf{X}_{\mathrm{free}}$ is a function of $\mathbf{q}$.

2.2. Objective and constraint functions

Consider a problem for minimizing compliance under total structural volume. The solution to the optimization problem is a statically determinate truss with the same absolute value of axial stress $\bar{\sigma}$ for all members [8, 20]. Hence, the cross-sectional area of member i is expressed as

$$A_i = \frac{|q_i|L_i}{\bar{\sigma}} \tag{7}$$

If member i connects nodes j and k, the square of L_i is given as

$$L_i^2 = (\mathbf{X}_k - \mathbf{X}_j)^{\mathrm{T}}(\mathbf{X}_k - \mathbf{X}_j) \tag{8}$$

where $\mathbf{X}_j \in \mathbb{R}^3$ and $\mathbf{X}_k \in \mathbb{R}^3$ are the position vectors of nodes j and k, respectively.

Let Young's modulus E and a very small positive number c be given. Then the compliance can be expressed as

$$F = \sum_{i=1}^{m} \frac{\bar{\sigma}L_i^2\sqrt{q_i^2 + c}}{E} \tag{9}$$

In (9), the absolute value of force density $|q_i|$ is substituted by $\sqrt{q_i^2 + c}$ for the purpose of smoothness of the objective function.

Fig. 1 illustrates the variation of $f_i(q_i) = \bar{\sigma}L_i^2|q_i|/E$ with and without smoothing. If $|q_i|$ is used for the objective function without smoothing, the sensitivity coefficient is discontinuous at $q_i = 0$, which causes difficulty of convergence. On the other hand, the sensitivity coefficient becomes continuous around 0 by introducing smoothing parameter c, which should be small enough to neglect its effect on the compliance value.

Since the product of the total structural volume and the compliance is independent of $\bar{\sigma}$, the total structural volume can be calculated after minimizing the compliance with arbitrary positive value of $\bar{\sigma}$ [19].

Define $\mathbf{R} \in \mathbb{R}^{n_{\mathrm{fix}}}$ as the vector of reaction forces corresponding to $\mathbf{X}_{\mathrm{fix}}$, which is obtained from

$$\mathbf{R} = \tilde{\mathbf{Q}}^{\mathrm{T}}_{\mathrm{link}}\mathbf{X}_{\mathrm{free}} + \tilde{\mathbf{Q}}_{\mathrm{fix}}\mathbf{X}_{\mathrm{fix}} \tag{10}$$

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

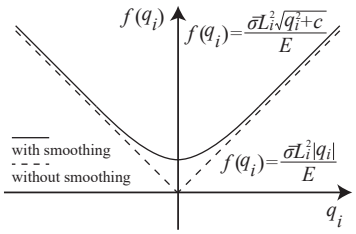


Figure 1: Sensitivity coefficient with and without smoothing.

In the optimization problem, the loading condition must be considered by prescribing reaction forces at loaded nodes as

$$\sum_{i\in\mathbf{I}_R}(R_i-\bar{P}_i)^2=0 \tag{11}$$

where $\mathbf{I}_R$ is the set of indices of reaction forces to be specified and $\bar{P}_i$ is the specified load value.

We further add constraint on nodal locations using a design surface. If z-coordinate of a point on the surface can be expressed as an explicit function of x- and y-coordinates, the constraint can be expressed as

$$z_i=f(x_i,y_i)\ \ (i=1,...,n) \tag{12}$$

Thus, the optimization problem can be formulated as

$$\text{minimize } F(\mathbf{q})=\sum_{i=1}^{m}\frac{\bar{\sigma}L_i^2\sqrt{q_i^2+c}}{E} \tag{13a}$$

$$\text{subject to } \sum_{i\in\mathbf{I}_R}(R_i-\bar{P}_i)^2=0 \tag{13b}$$

$$z_i=f(x_i,y_i)\ \ (i=1,...,n) \tag{13c}$$

$$0.0\le x_i\le 1.0\ \ (i=1,...,n) \tag{13d}$$

$$0.0\le y_i\le 1.0\ \ (i=1,...,n) \tag{13e}$$

$$q_k^{\mathrm{L}}\le q_k\le q_k^{\mathrm{U}}\ \ (k=1,...,m) \tag{13f}$$

where q_k^{L} and q_k^{U} are the lower and upper bounds for q_k, respectively. Necessity of constraints (13d) and (13e) will be explained in the next sub-section.

2.3. Explicit expression of Bézier surface

According to constraint (13c), the design surface need to be expressed as an explicit function. A tensor product Bézier surface of order $M\times N$ can be expressed with parameters $u,v\in[0.0,1.0]$ as

$$\mathbf{S}(u,v)=\sum_{i=0}^{M}\sum_{j=0}^{N}B_i^M(u)B_j^N(v)\mathbf{P}_{ij} \tag{14}$$

where $\mathbf{P}_{ij}\in\mathbb{R}^3$ is the position vector of a control point to define the design surface. The functions $B_i^M(u)$ and $B_j^N(v)$ are the Bernstein basis polynomials in u- and v-directions, respectively, which are written as

$$B_i^M(u)=\binom{M}{i}u^i(1-u)^{M-i} \tag{15a}$$

$$B_j^N(v)=\binom{N}{j}v^j(1-v)^{N-j} \tag{15b}$$

If $\mathbf{P}_{ij}$ is described as

$$\mathbf{P}_{ij}=\left(i/M,\ j/N,\ b_{ij}\right)^{\mathrm{T}}\ \ (i=0,...,M,\ j=0,...,N,\ b_{ij}\in\mathbb{R}) \tag{16}$$

then z-coordinate of any arbitrary point on the tensor product Bézier surface can be expressed as an explicit function of its x- and y-coordinates, since the following equation is satisfied [21]:

$$\mathbf{S}(u,v)=\left(u,\ v,\ \sum_{i=0}^{M}\sum_{j=0}^{N}B_i^M(u)B_j^N(v)b_{ij}\right)^{\mathrm{T}} \tag{17}$$

If the surface is scaled to satisfy $x=u$ and $y=v$, then (17) can be re-written as

$$z=f(x,y)=\sum_{i=0}^{M}\sum_{j=0}^{N}B_i^M(x)B_j^N(y)b_{ij} \tag{18}$$

From (18), every control point $\mathbf{P}_{ij}$ must be placed at equal intervals in x- and y-directions and its x- and y-coordinates must be within the range of $[0.0,1.0]$. The latter condition requires constraints (13d) and (13e) in the optimization problem.

2.4. Sensitivity analysis

To reduce computational cost for solving (13), sensitivity coefficients of objective and constraint functions are analytically obtained in this sub-section. The objective function (13a) is differentiated with respect to q_l as

$$\frac{\partial F(\mathbf{q})}{\partial q_l}=\frac{\bar{\sigma}q_lL_l^2}{E\sqrt{q_l^2+c}}+\sum_{i=1}^{m}\left(\frac{\bar{\sigma}\sqrt{q_i^2+c}}{E}\cdot\frac{\partial L_i^2}{\partial q_l}\right) \tag{19}$$

From (8), the sensitivity coefficient of L_i^2 with respect to q_l is obtained as

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

$$\frac{\partial L_i^2}{\partial q_l} = 2(\mathbf{X}_k - \mathbf{X}_j)^{\mathrm{T}} \cdot \frac{\partial(\mathbf{X}_k - \mathbf{X}_j)}{\partial q_l} \tag{20}$$

Differentiation of Eq. (6) with respect to q_l leads to

$$\tilde{\mathbf{Q}}_{\mathrm{free}} \frac{\partial \mathbf{X}_{\mathrm{free}}}{\partial q_l} + \frac{\partial \tilde{\mathbf{Q}}_{\mathrm{free}}}{\partial q_l} \mathbf{X}_{\mathrm{free}} = -\frac{\partial \tilde{\mathbf{Q}}_{\mathrm{link}}}{\partial q_l} \mathbf{X}_{\mathrm{fix}} \tag{21}$$

Therefore, sensitivity coefficient of objective function with respect to design variable can be analytically obtained from Eq. (19) to (21).

Next, Eq. (13b) is differentiated as

$$\sum_{i \in \mathbf{I}_R} \left(2(R_i - \bar{P}_i) \frac{\partial R_i}{\partial q_l} \right) = 0 \tag{22}$$

Sensitivity coefficients of reaction forces are computed from (10) as

$$\frac{\partial \mathbf{R}}{\partial q_l} = \frac{\partial \tilde{\mathbf{Q}}_{\mathrm{link}}^{\mathrm{T}}}{\partial q_l} \mathbf{X}_{\mathrm{free}} + \tilde{\mathbf{Q}}_{\mathrm{link}}^{\mathrm{T}} \frac{\partial \mathbf{X}_{\mathrm{free}}}{\partial q_l} + \frac{\partial \tilde{\mathbf{Q}}_{\mathrm{fix}}}{\partial q_l} \mathbf{X}_{\mathrm{fix}} \tag{23}$$

Sensitivity coefficient of $f(x_k, y_k)$ is derived by differentiating Eq. (18) at $(x, y) = (x_k, y_k)$ as

$$\frac{\partial f(x_k, y_k)}{\partial q_l} = \sum_{i=0}^{M} \sum_{j=0}^{N} \left(\frac{\partial B_i^M(x_k)}{\partial q_l} B_j^N(y_k) + B_i^M(x_k) \frac{\partial B_j^N(y_k)}{\partial q_l} \right) b_{ij} \tag{24}$$

From (15a) and (15b), differentiation of Bernstein basis polynomial with respect to q_l with $u = x_k$ and $v = y_k$ leads to

$$\frac{\partial B_i^M(x_k)}{\partial q_l} = \binom{M}{i} \left(i x_k^{i-1} (1 - x_k)^{M-i} - (M - i) x_k^i (1 - x_k)^{M-i-1} \right) \frac{\partial x_k}{\partial q_l} \tag{25a}$$

$$\frac{\partial B_j^N(y_k)}{\partial q_l} = \binom{N}{j} \left(j y_k^{j-1} (1 - y_k)^{N-j} - (N - j) y_k^j (1 - y_k)^{N-j-1} \right) \frac{\partial y_k}{\partial q_l} \tag{25b}$$

Sensitivity coefficients of nodal coordinates in Eqs. (20), (23), and (25) are derived by solving the system of linear equations (21) for $\partial \mathbf{X}_{\mathrm{free}} / \partial q_l$.

3. System architecture

In this section, the process of developing the optimizer in Grasshopper is described. In optimizing trusses, initial ground structure, support and loading conditions, and additional constraints must be translated to numerical data to be incorporated to the optimization program. It takes much time and is prone to mistakes if we conduct the translation manually. Instead, we developed a Grasshopper component to automatically extract information which is necessary for solving the optimization problem.

Fig. 2 shows the Grasshopper component coded in C# with the aid of Grasshopper SDK. This component is capable of handling geometry classes of Rhinoceros directly. However, we dare to set *supporting condition Sp* as a text input instead of *Point* class, to facilitate releasing the boundary condition; for example, the user only need to write "3, xy" when releasing the third node in z-direction. If the component succeeds to collect the data from the required fields, it triggers an initialization *method*, a code block containing a series of statements. The component converts software-specific geometry information into numerical data through the framework of RhinoCommon API so that the optimization algorithm is able to handle it. The user is also able to check the numerical data because it is saved in text format in the local storage. Note that the initialization *method* does not include optimization procedure, because the optimization should start after finishing all the input setting. We further added six optional inputs. L, x, y, z are positive real numbers to control shape change from the initial solution. Though *design surface S* is also an optional input for shape controlling, it can be specified by *Surface* class, which is one of basic geometry types of Rhinoceros. Once the component collects the *Surface* geometry, it retrieves locations of its controlling points in order to calculate Eqs. (18) and (24) in the optimization process. I is a random seed to randomize initial force densities; by altering this value, the user is able to obtain different solutions from the same initial ground structure.

Once the component is double-clicked, it starts calling an optimization program compiled in FORTRAN, where SNOPT ver. 7.2, an SQP solver is incorporated [22]. The FORTRAN program randomizes the initial force densities based on the prescribed random seed, conducts the optimization, and returns an optimal solution back to the Grasshopper component. Owing to geometry processing libraries and a graphical interface of Rhinoceros, the optimal solutions can be easily visualized, which offers users real-time feedback of the op-

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

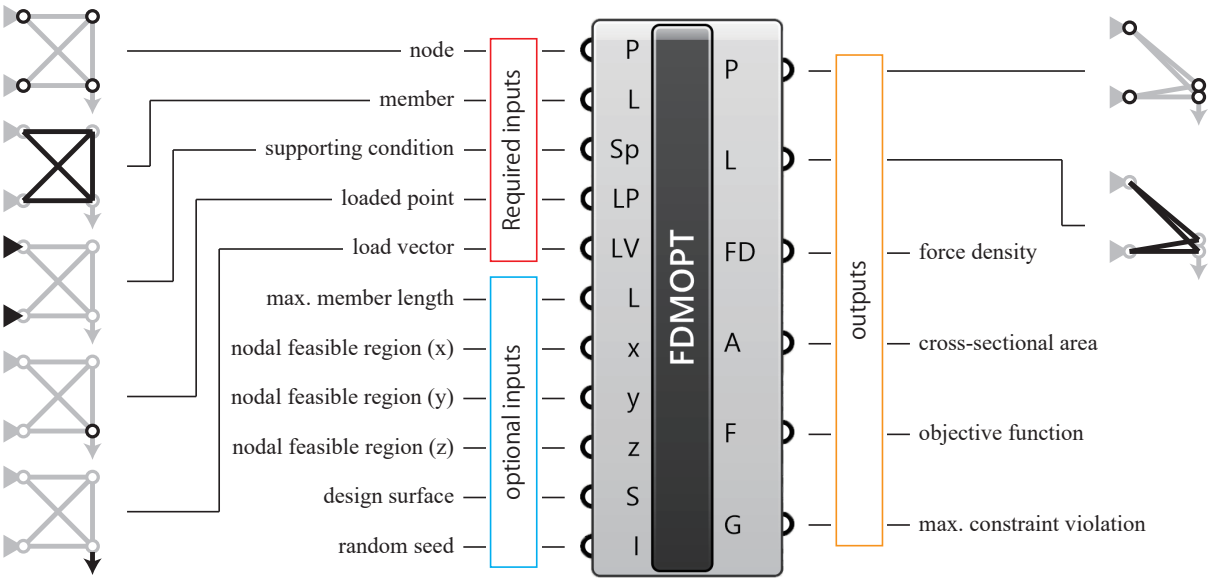


Figure 2: Compiled Grasshopper component.

timal geometry and topology of trusses. The whole optimization workflow is illustrated in Fig. 3. This framework allows the user to conduct optimization with the proposed method, review the result, and change the conditions of the initial model for design exploration easily and interactively.

4. Numerical examples

In this section, we demonstrate our proposed method through numerical examples. In the optimization process, Young's modulus E, the absolute value of axial stress $\bar{\sigma}$, and the smoothing approximation factor c are first set to be 1.0 N/mm^2, 1.0 N/mm^2, and 1.0 $\times$ 10^{-6} N^2/m^2, respectively. Note that the cross-sectional areas and the objective function value are re-scaled such that $E = 2.0 \times 10^5$ [N/mm^2], $\bar{\sigma} = 200$ [N/mm^2], and the maximum volume $\bar{V} = 0.01$ [m^3] after obtaining the optimal solutions.

Let q_{init} $(i = 1, ..., m)$ and $\bar{r}$ (> 0) denote initial force density of the i th member for the initial model and the random seed. Then every force density is randomized within the following range before starting optimization:

$$q_i^{\text{start}} \in [q_i^{\text{init}} - \bar{r}, q_i^{\text{init}} + \bar{r}] \tag{26}$$

Empirically, the order of $\bar{r}$ should be less than that of initial force densites to stabilize the optimization. Considering the maximum absolute value of initial force density is around 10.0 N/m in the following examples, $\bar{r}$ is set to be 1.0 N/m. After randomizing the initial force densites, the lower and upper bounds are defined as

$$q_i^{\text{L}} = q_i^{\text{start}} - \bar{d} \tag{27a}$$
$$q_i^{\text{U}} = q_i^{\text{start}} + \bar{d} \tag{27b}$$

A small value is preferrable for $\bar{d}$ in view of convergence; however relatively large value compared with $\bar{r}$ and initial force densities must be assigned in order to ensure wide variable range. We set $\bar{d} = 1.0 \times 10^2$ [N/m] in the following examples.

Optimization is conducted 100 times for the 100 different initial sets of initial force densities to select the best solution with the least objective function value. If the maximum violation of constraints is more than 1.0×10^{-4}, the solution is rejected as infeasible. We use a PC with Intel Core i9-7900X [3.30 GHz/10 Core] processor in the following examples.

4.1. Semi-cylindrical latticed shell

The first example is a semi-cylindrical double-layer latticed shell with 46 nodes and 225 members, as shown in Fig. 4. The bottom four corner nodes are pin-supported and all the upper nodes are subjected to downward unit loads 1.0 N. Note again that these 25 supported or loaded nodes cannot move in the optimization process. A semi-cylindrical quadratic Bézier surface is introduced as a design surface which interpolates the 21 free nodes.

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

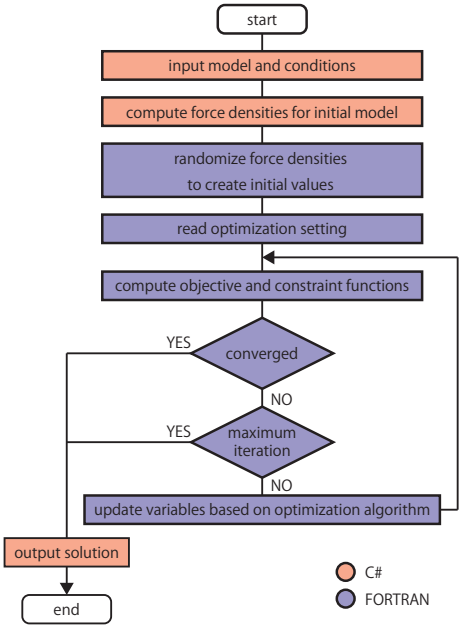


Figure 3: Overall flowchart conducted in the optimizer.

After 100 trials, 73 valid solutions are obtained. The maximum, median, minimum, average values and standard deviation of F for the 73 valid solutions are listed in Table 1. It takes 115 seconds for each trial on average. The optimal solution with the least value of $F = 73.104$[N·m] is shown in Fig. 5, and their nodal locations are listed in Table. 2. The pair of nodes 22 and 23 is coalescent within the range of 0.01 m to generate the simpler shape with 45 nodes.

We also conducted optimization without smoothing of objective function, and obtained only 21 feasible solutions out of 100 initial solutions. Therefore, smoothing of objective function is crucial to improve optimization performance.

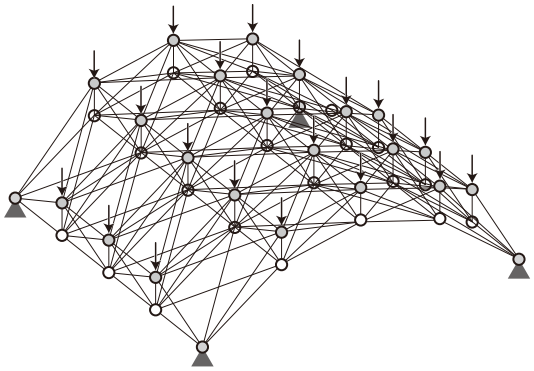
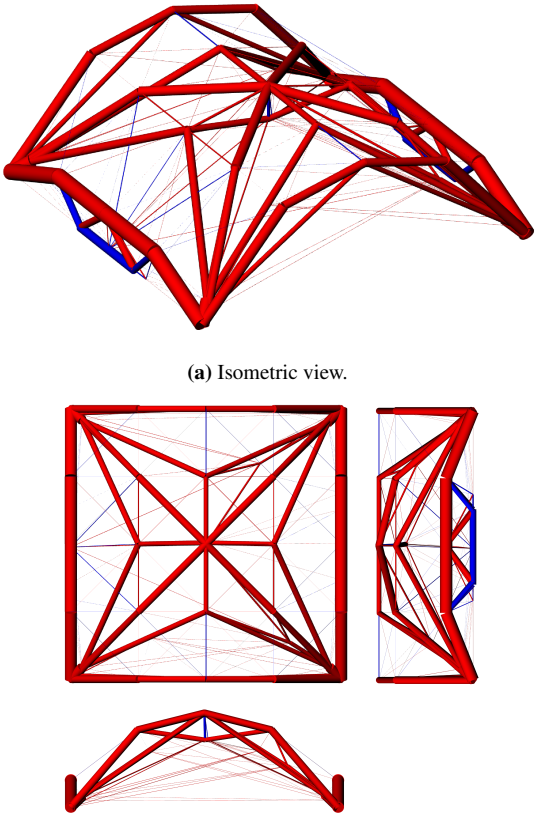


Figure 4: Initial ground structure of a semi-cylindrical lattices shell.



(a) Isometric view.

(b) Plan and elevations.

Figure 5: The optimal solution of semi-cylindrical latticed shell ($F = 73.104$[N·m]).

Table 1: Statistical results of F [N·m] for 73 converged solutions of semi-cylindrical latticed shell.

Max.	Median	Min.	Average	Std. dev.
112.109	75.084	73.104	76.548	5.885

Since the optimal solution to Problem (13) includes overlapped nodes and very thin members, and the configuration is obscure, we improve the optiimzation result with the following steps. First, overlapped nodes and members are unified in the optimal solution to Problem (13). Let m^* denote the number of members after unifying overlapped nodes and members, then the cross-sectional areas of members are optimized by solving the following problem:

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

$$\text{minimize} \quad F(\mathbf{A}) = \sum_{i=1}^{m^*} \frac{N_i^2 L_i}{E A_i} \tag{28a}$$

$$\text{subject to} \quad \sum_{i=1}^{m^*} A_i L_i \leq \bar{V} \tag{28b}$$

$$A_i^{\mathrm{L}} \leq A_i \leq A_i^{\mathrm{U}} (i = 1, ..., m^*) \tag{28c}$$

The lower bound A_i^{L} for A_i is 1.0×10^{-3} [m^2], and the member with $A_i \leq 1.0 \times 10^{-2}$ [m^2] is eliminated after optimization. Finite difference method is applied for computing sensitivity coefficients, because Problem (28) is solved only once.

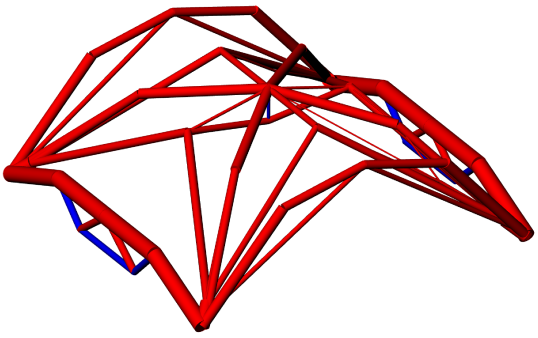
The improved solution is shown in Fig. 6. The compliance F slightly increased to 73.259 N·m. This implies that the influence of re-optimization is trivial and FDMopt produced a sufficiently converged solution. The number of members is reduced to 85. Although there is no constraint on symmetry, almost symmetric optimal shape is obtained.

4.2. Semi-spherical latticed shell

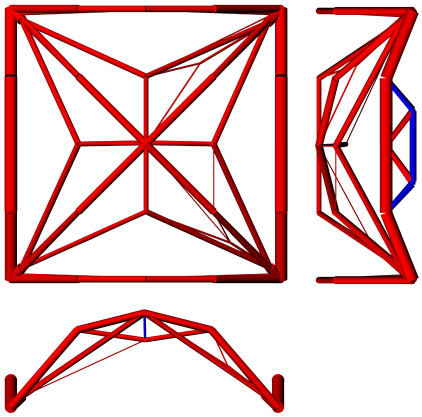
The second example is a semi-spherical double-layer latticed shell as shown in Fig. 7 with the same number of nodes and members as the first example. The bottom four corner nodes are pin-supported and all the upper nodes are subjected to downward unit loads. As well as the first example, a semi-spherical quadratic Bézier surface is introduced as a design surface which interpolates the 21 free nodes.

After 100 trials, 84 valid solutions are obtained. The maximum, median, minimum, average values and standard deviation of F for the 84 valid solutions are listed in Table 3. It takes 481 seconds for each trial on average. The optimal solution with the least value of F = 37.646 [N·m] is shown in Fig. 8, and their nodal locations are listed in Table. 4. The distance between nodes 7 and 12 is 0.007 m which is very small.

We further solve Problem (28) in the same manner as the first example, and the result is shown in Fig. 9. The compliance F becomes 37.824 [N·m], and the number of members is reduced from 225 to 83.



(a) Isometric view.



(b) Plan and elevations.

Figure 6: The solution after re-optimization of semi-cylindrical latticed shell (F = 73.259[N·m]).

5. Conclusion

We proposed an interactive and integrated design approach to truss design by developing a Grasshopper component for the simultaneous optimization of geometry and topology of trusses. The numerical difficulty due to melting nodes can be successfully avoided using force density as design variable, which contributes to generating optimal solutions with a variety of geometry and topology from a relatively sparse initial ground structure. Moreover, a tensor product Bézier surface is successfully incorporated as a design surface of the optimizer, which enables reflection of the user's shape preference.

We further introduced sensitivity analysis in the solution process of the optimization problem to reduce the computational cost. Owing to the sensitivity analysis and the less number of design variables compared with previous researches, the optimizer is able to yield a solution in a short time even for relatively complex models demonstrated in the examples.

Although the design surface is restricted to a tensor product Bézier surface whose control points are placed at equal intervals in x- and y-directions, this restriction is expected to be alleviated by using geometric transformation in future research.

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

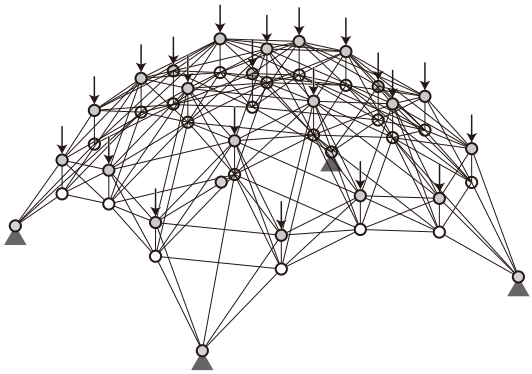
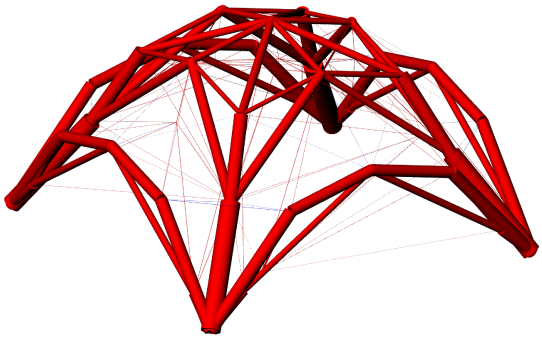
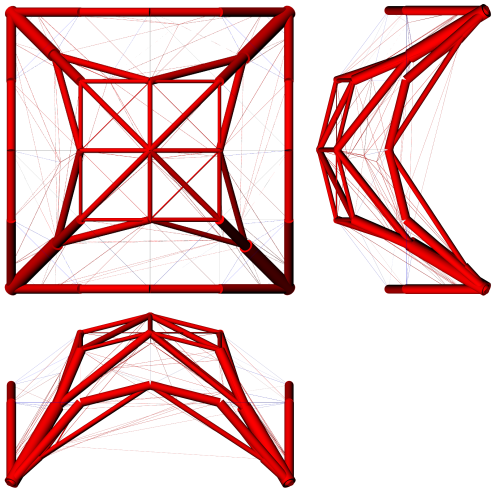


Figure 7: Initial ground structure of a semi-spherical latticed shell.



(a) Isometric view.



(b) Plan and elevations.

Figure 8: The optimal solution of semi-spherical latticed shell (F = 37.646[N·m]).

Acknowledgement

This work is supported by the "SPIRITS" program of Kyoto University, Grant-in-Aid for JSPS Research Fellow Grant Number 18J21456 and JSPS KAKENHI Grant Number 16H03014. The authors wish to express their appreciation to Professor Caitlin Mueller at Massachusetts Institute of Technology for her valuable comments.

References

[1] R. Razani, Behavior of fully stressed design of structures and its relationshipto minimum-weight design, AIAA Journal 3 (1965) 2262–2268.

[2] U. Kirsch, Optimal topologies of truss structures, Computer Methods in Applied Mechanics and Engineering 72 (1989) 15 – 28.

[3] M. P. Bendsøe, O. Sigmund, Topology Optimization: Theory, Methods and Applications, Springer, 2004.

[4] W. S. Dorn, R. E. Gomory, H. J. Greenberg, Automatic design of optimal structures, Journal de Mecanique 3 (1964) 25–52.

[5] K. Imai, L. A. Schmit Jr, Configuration optimization of trusses 107 (1981) 745–756.

[6] E. A. Sadek, Dynamic optimization of framed structures with variable layout, International Journal for Numerical Methods in Engineering 23 (1986) 1273–1294.

[7] M. Ohsaki, Simultaneous optimization of topology and geometry of a regular plane truss, Computers and Structures 66 (1998) 69–77.

[8] W. Achtziger, On simultaneous optimization of truss geometry and topology, Structural and Multidisciplinary Optimization 33 (2007) 285–304.

[9] M. Ohsaki, Optimization of finite dimensional structures, CRC Press, Hoboken, NJ, 2010.

[10] T. Hagishita, M. Ohsaki, Topology optimization of trusses by growing ground structure method, Structural and Multidisciplinary Optimization 37 (2009) 377–393.

[11] J. J. McKeown, Growing optimal pin-jointed frames, Structural optimization 15 (1998) 92–100.

[12] E. Ramm, K.-U. Bletzinger, R. Reitinger, Shape optimization of shell structures, Revue Européenne des Éléments Finis 2 (1993) 377–398.

[13] M. Ohsaki, T. Nakamura, M. Kohiyama, Shape optimization of a double-layer space truss described by a parametric surface, International Journal of Space Structures 12 (1997) 109–119.

[14] R. McNeel, Grasshopper - algorithmic modeling for rhino, http://www.grasshopper3d.com/, 2018. Accessed: 2018-07-13.

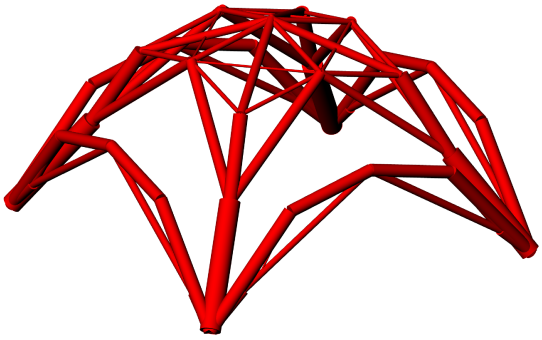
[15] R. McNeel, Rhinoceros - design, model, present, analyze, realize..., https://www.rhino3d.com/, 2018. Accessed: 2018-07-13.

[16] D. Rutten, Galapagos evolutionary solver, https://www.grasshopper3d.com/group/galapagos, 2018. Accessed: 2018-07-13.

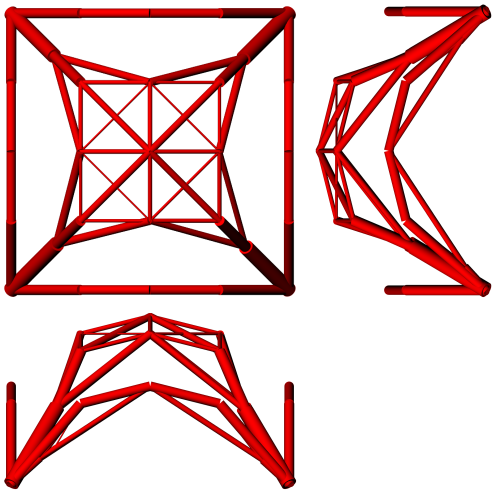
[17] Rechenraum e.U., Goat - free optimization solver component for rhino's grasshopper, https://www.rechenraum.com/en/goat.html, 2018. Accessed: 2018-07-13.

[18] E. Bradner, F. Iorio, M. Davis, Parameters tell the design story: Ideation and abstraction in design optimization, in: Proceedings of the Symposium on Simulation for Architecture and Urban Design, SimAUD '14, Society for Computer Simulation International, San Diego, CA, USA, 2014, pp. 26:1–26:8.

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository



(a) Isometric view.



(b) Plan and elevations.

Figure 9: The solution after re-optimization of semi-spherical latticed shell (F = 37.824[N·m]).

[19] M. Ohsaki, K. Hayashi, Force density method for simultaneous optimization of geometry and topology of trusses, Structural and Multidisciplinary Optimization 56 (2017) 1157–1168.
[20] W. S. Hemp, Optimum structures, Clarendon Press Oxford, 1973.
[21] G. Farin, Curves and Surfaces for CAGD: A Practical Guide, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th edition, 2002.
[22] P. E. Gill, W. Murray, M. A. Saunders, Snopt: An sqp algorithm for large-scale constrained optimization, SIAM JOURNAL ON OPTIMIZATION 12 (1997) 979–1006.

Table 2: Nodal locations of semi-cylindrical latticed shell [m].

	Initial solution			Best optimal solution		
Node	x	y	z	x	y	z
1	0.000	0.000	0.000	0.000	0.000	0.000
2	0.250	0.000	0.188	0.024	0.015	0.023
3	0.500	0.000	0.250	0.294	0.153	0.207
4	0.750	0.000	0.188	0.978	0.015	0.022
5	1.000	0.000	0.000	1.000	0.000	0.000
6	0.000	0.250	0.000	0.000	0.295	0.000
7	0.250	0.250	0.188	0.041	0.043	0.039
8	0.500	0.250	0.250	0.812	0.135	0.153
9	0.750	0.250	0.188	0.958	0.043	0.040
10	1.000	0.250	0.000	1.000	0.367	0.000
11	0.000	0.500	0.000	0.000	0.355	0.000
12	0.250	0.500	0.188	0.000	0.489	0.000
13	0.500	0.500	0.250	0.502	0.505	0.250
14	0.750	0.500	0.188	0.514	0.506	0.250
15	1.000	0.500	0.000	1.000	0.628	0.000
16	0.000	0.750	0.000	0.000	0.635	0.000
17	0.250	0.750	0.188	0.042	0.958	0.040
18	0.500	0.750	0.250	0.700	0.799	0.210
19	0.750	0.750	0.188	0.951	0.949	0.047
20	1.000	0.750	0.000	1.000	0.690	0.000
21	0.000	1.000	0.000	0.000	1.000	0.000
22	0.250	1.000	0.188	0.025	0.983	0.025
23	0.500	1.000	0.250	0.973	0.981	0.027
24	0.750	1.000	0.188	0.978	0.985	0.021
25	1.000	1.000	0.000	1.000	1.000	0.000
26	0.250	0.000	0.288	0.250	0.000	0.288
27	0.500	0.000	0.350	0.500	0.000	0.350
28	0.750	0.000	0.288	0.750	0.000	0.288
29	0.000	0.250	0.100	0.000	0.250	0.100
30	0.250	0.250	0.288	0.250	0.250	0.288
31	0.500	0.250	0.350	0.500	0.250	0.350
32	0.750	0.250	0.288	0.750	0.250	0.288
33	1.000	0.250	0.100	1.000	0.250	0.100
34	0.000	0.500	0.100	0.000	0.500	0.100
35	0.250	0.500	0.288	0.250	0.500	0.288
36	0.500	0.500	0.350	0.500	0.500	0.350
37	0.750	0.500	0.288	0.750	0.500	0.288
38	1.000	0.500	0.100	1.000	0.500	0.100
39	0.000	0.750	0.100	0.000	0.750	0.100
40	0.250	0.750	0.288	0.250	0.750	0.288
41	0.500	0.750	0.350	0.500	0.750	0.350
42	0.750	0.750	0.288	0.750	0.750	0.288
43	1.000	0.750	0.100	1.000	0.750	0.100
44	0.250	1.000	0.288	0.250	1.000	0.288
45	0.500	1.000	0.350	0.500	1.000	0.350
46	0.750	1.000	0.288	0.750	1.000	0.288

京都大学 KYOTO UNIVERSITY
A Self-archived copy in
Kyoto University Research Information Repository
https://repository.kulib.kyoto-u.ac.jp
京都大学学術情報リポジトリ KURENAI 紅 Kyoto University Research Information Repository

Table 3: Statistical results of F [N·m] for 84 converged solutions of semi-spherical latticed shell.

Max.	Median	Min.	Average	Std. dev.
42.588	37.706	37.646	37.799	0.569

Table 4: Nodal locations of semi-spherical latticed shell [m].

	Initial solution			Best optimal solution		
Node	x	y	z	x	y	z
1	0.000	0.000	0.000	0.000	0.000	0.000
2	0.250	0.000	0.188	0.104	0.000	0.093
3	0.500	0.000	0.250	0.158	0.000	0.133
4	0.750	0.000	0.188	0.906	0.000	0.086
5	1.000	0.000	0.000	1.000	0.000	0.000
6	0.000	0.250	0.188	0.000	0.085	0.077
7	0.250	0.250	0.375	0.163	0.163	0.273
8	0.500	0.250	0.438	0.766	0.177	0.325
9	0.750	0.250	0.375	0.845	0.155	0.261
10	1.000	0.250	0.188	1.000	0.082	0.075
11	0.000	0.500	0.250	0.177	0.457	0.394
12	0.250	0.500	0.438	0.165	0.168	0.278
13	0.500	0.500	0.500	0.503	0.467	0.499
14	0.750	0.500	0.438	0.747	0.480	0.439
15	1.000	0.500	0.250	0.824	0.234	0.325
16	0.000	0.750	0.188	0.000	0.892	0.096
17	0.250	0.750	0.375	0.154	0.846	0.261
18	0.500	0.750	0.438	0.350	0.760	0.410
19	0.750	0.750	0.375	0.837	0.837	0.273
20	1.000	0.750	0.188	1.000	0.908	0.083
21	0.000	1.000	0.000	0.000	1.000	0.000
22	0.250	1.000	0.188	0.089	1.000	0.081
23	0.500	1.000	0.250	0.603	0.956	0.282
24	0.750	1.000	0.188	0.905	1.000	0.086
25	1.000	1.000	0.000	1.000	1.000	0.000
26	0.250	0.000	0.288	0.250	0.000	0.288
27	0.500	0.000	0.350	0.500	0.000	0.350
28	0.750	0.000	0.288	0.750	0.000	0.288
29	0.000	0.250	0.288	0.000	0.250	0.288
30	0.250	0.250	0.475	0.250	0.250	0.475
31	0.500	0.250	0.538	0.500	0.250	0.538
32	0.750	0.250	0.475	0.750	0.250	0.475
33	1.000	0.250	0.288	1.000	0.250	0.288
34	0.000	0.500	0.350	0.000	0.500	0.350
35	0.250	0.500	0.538	0.250	0.500	0.538
36	0.500	0.500	0.600	0.500	0.500	0.600
37	0.750	0.500	0.538	0.750	0.500	0.538
38	1.000	0.500	0.350	1.000	0.500	0.350
39	0.000	0.750	0.288	0.000	0.750	0.288
40	0.250	0.750	0.475	0.250	0.750	0.475
41	0.500	0.750	0.538	0.500	0.750	0.538
42	0.750	0.750	0.475	0.750	0.750	0.475
43	1.000	0.750	0.288	1.000	0.750	0.288
44	0.250	1.000	0.288	0.250	1.000	0.288
45	0.500	1.000	0.350	0.500	1.000	0.350
46	0.750	1.000	0.288	0.750	1.000	0.288