

# 軸力密度法による位相と形状の同時最適化手法 -部材長と節点位置の制約方法の分析-

○林 和希\*1 大崎 純\*2

キーワード：建築構造最適化 軸力密度法 トラス 位相最適化 形状最適化

## 1. 序

トラスの位相最適化については多くの研究がなされており、数理計画法・発見的手法（ヒューリスティクス）とともに、様々な手法が確立されてきた[1,2]。位相を最適化するために、密に接続されたモデル（グランドストラクチャ）から疎な最適トラスを求めるグランドストラクチャ法[3]が用いられることが多い。

位相に加えて形状も最適化する問題は、節点位置と断面積を変数とし、最適化後に十分に細い部材を除去する方法によって解くことができる[4]。しかし、最適形状を高精度で求めるためには、節点・部材ともに密なグランドストラクチャを必要とし、計算負荷が大きい。一方、疎なグランドストラクチャを用いて節点の移動範囲を大きくすると、極めて短い部材で接続され近接した節点（melting nodes）が発生し、解析が困難になる[5,6]。

このような問題点を解決するため、疎なトラスから始まり、節点や部材を発見的手法によって繰り返し追加する進化的手法が開発された[7,8]。しかし、その解は理論的に定義された最適性条件をどれも満たさないため、解の理論的最適性を保証できない。また、Achtziger[6]は陰関数を用いた最適化手法を開発したが、節点位置が重ならないための制約が必要となる。

本報では、著者らの提案した軸力密度を設計変数とする方法[10]を元に、複雑なラティスシェル構造に部材長や節点位置の制約を課して、良好な解形状を得る方法とその適用例を報告する。

## 2. 軸力密度法

軸力密度法(force density method)を用いて、節点座標と固定節点の反力を軸力密度の関数として定式化する。

トラスの部材数を  $m$ 、節点数を  $n$  とする。部材  $k$  が節点  $j$  と  $k$  ( $j < k$ ) で接続されているとき、接続行列  $\mathbf{C} \in \mathbb{R}^{m \times n}$  の  $[i, j]$  成分  $C_{ij}$  を次のように定義する。

$$C_{ij} = -1, \quad C_{ik} = 1 \quad (i = 1, \dots, m) \quad (1)$$

部材  $i$  の軸力  $N_i$  の長さ  $L_i$  に対する比  $q_i = N_i / L_i$  を軸力密度という。  $q_i$  を並べたベクトルを  $\mathbf{q} \in \mathbb{R}^m$  とし、接続行列  $\mathbf{C}$  を用いて軸力密度行列  $\mathbf{Q}$  を以下のように定義する。

$$\mathbf{Q} = \mathbf{C}^T \text{diag}(\mathbf{q}) \mathbf{C} \quad (2)$$

$\mathbf{Q}$  は  $x, y, z$  方向成分に対して同一であり、それらを重ね合わせて、自由節点座標と固定節点座標（以下、自由座標、固定座標）に分けて並べた行列を  $\tilde{\mathbf{Q}} \in \mathbb{R}^{3n \times 3n}$  とする。

トラスの形状最適化では、荷重作用点の座標は固定させるため、固定座標として扱う。自由座標数を  $n_{\text{free}}$ 、固定座標数を  $n_{\text{fix}}$  とし、 $\tilde{\mathbf{Q}}$  から自由座標成分  $\tilde{\mathbf{Q}}_{\text{free}} \in \mathbb{R}^{n_{\text{free}} \times n_{\text{free}}}$ 、自由・固定座標間のリンク成分  $\tilde{\mathbf{Q}}_{\text{link}} \in \mathbb{R}^{n_{\text{free}} \times n_{\text{fix}}}$  を抽出する。自由座標成分、固定座標成分の座標のベクトルを  $\mathbf{X}_{\text{free}} \in \mathbb{R}^{n_{\text{free}}}$ 、 $\mathbf{X}_{\text{fix}} \in \mathbb{R}^{n_{\text{fix}}}$  とすると、次式が成り立つ。

$$\tilde{\mathbf{Q}}_{\text{free}} \mathbf{X}_{\text{free}} = -\tilde{\mathbf{Q}}_{\text{link}} \mathbf{X}_{\text{fix}} \quad (3)$$

この連立1次方程式を  $\mathbf{X}_{\text{free}}$  について解くと、指定した軸力密度で釣り合う形状の自由座標が得られる。したがって、 $\mathbf{X}_{\text{free}}$  と  $\mathbf{X}_{\text{fix}}$  を並べたベクトル  $\mathbf{X}$  は  $\mathbf{q}$  の関数である。節点  $j$  と  $k$  で接続された部材の長さ  $L_i$  は次式で得られる。

$$L_i = \|\mathbf{X}_k - \mathbf{X}_j\| \quad (4)$$

また、固定座標の反力  $\mathbf{R} \in \mathbb{R}^{n_{\text{fix}}}$  は次式で表される。

$$\mathbf{R} = \tilde{\mathbf{Q}}_{\text{link}}^T \mathbf{X}_{\text{free}} + \tilde{\mathbf{Q}}_{\text{fix}} \mathbf{X}_{\text{fix}} \quad (5)$$

## 3. 最適化問題

以下では、総体積制約下でコンプライアンスを最小化する問題を考える。この問題に対する最適解は、静定トラスとなり、各部材の応力の絶対値が一定となる [1]。したがって、応力の絶対値を  $\bar{\sigma}$ 、ヤング係数を  $E$  とすると、部材総体積  $V$  とコンプライアンス  $F$  は、軸力密度を用いて次のように求められる。

$$V = \sum_{i=1}^m \frac{|q_i| L_i^2}{\bar{\sigma}}, \quad F = \sum_{i=1}^m \frac{\bar{\sigma} |q_i| L_i^2}{E} \quad (6)$$

以上より、部材体積とコンプライアンスを軸力密度の関数として表現できた。

トラスの総体積制約下でのコンプライアンスを最小化する問題の最適解は静定トラスであり、存在する部材の軸力  $N_k$  は設計に依存しないことを用いると、部材総体積とコンプライアンスの積は以下のように一定値となる [7]。

$$VF = \frac{1}{E} \left( \sum_{i=1}^m q_i L_i^2 \right)^2$$

$$= \frac{1}{E} \left( \sum_{i=1}^m N_i L_i \right)^2 \quad (7)$$

したがって、 $\bar{\sigma}$  を指定値としてコンプライアンスを最小化し、最適解を得た後で部材体積を計算すればよい。固定させた荷重作用点に反力  $\bar{\mathbf{R}}$  を指定することで荷重条件を満たす解が得られるため、最適化問題は以下のように表される。

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

$$\text{subject to } \sum_{i \in \mathfrak{R}} (R_i(\mathbf{q}) - \bar{R}_i)^2 \leq 0$$

$$q_i^L \leq q_i \leq q_i^U \quad (i=1, \dots, m) \quad (8)$$

$\mathfrak{R}$  は荷重に対応したインデックスであり、 $q_k^U$ 、 $q_k^L$  は  $q_k$  の上・下限値、 $c$  は十分に小さな値である。式(6)では不連続関数であった変数の絶対値  $|q_i|$  を  $\sqrt{q_i^2 + c}$  にスムージングしている。また、制約関数を二乗ノルムでまとめてプログラム上で制約関数を読み込む回数を減らしている。

#### 4. 部材長と節点位置の追加制約

本手法では節点位置に制約を設ける必要がないため、最適化前後で形状に著しい変化が見られることが多い。そこで、大幅な形状変化を避けるための制約条件式を最適化問題(8)に追加することを考える。

まず、部材長の制約を以下のように加える。

$$\text{追加制約 1: } L_i^2 \leq \bar{L}_i^2 \quad (9)$$

次に、 $z$  座標が  $x, y$  座標の関数として定義されるような曲面に限定し、以下のような制約式を導入する。

$$\text{追加制約 2: } z_i = f(x_i, y_i) \quad (10)$$

これらの制約条件式も、式(8)と同様に軸力密度に関して連続微分可能な関数となるため、新たな不連続性は生じない。

#### 5. 感度解析

逐次 2 次計画法の利用に際して、解析効率の向上・計算時間の短縮を目的として目的関数と制約関数の 1 次の感度係数を計算する。

まず、目的関数を  $q_i$  で微分すると次式となる。

$$\frac{\partial \tilde{F}(\mathbf{q})}{\partial q_i} = \frac{\bar{\sigma} q_i L_i^2}{E \sqrt{q_i^2 + c}} + \sum_{i=1}^m \left( \frac{\bar{\sigma} \sqrt{q_i^2 + c}}{E} \cdot \frac{\partial L_i^2}{\partial q_i} \right) \quad (11)$$

式(11)は部材長の 2 乗の感度係数の項を含んでいる。式(4)より、 $L_i^2$  を  $q_i$  で微分して、感度係数は以下のように得られる。

$$\frac{\partial L_i^2}{\partial q_i} = 2(\mathbf{X}_j - \mathbf{X}_k) \cdot \frac{\partial(\mathbf{X}_j - \mathbf{X}_k)}{\partial q_i} \quad (12)$$

式(12)より、さらに座標ベクトルの感度係数が必要であり、それらは次式で求められる。

$$\frac{\partial \mathbf{X}_{\text{free}}}{\partial q_i} = -\tilde{\mathbf{Q}}_{\text{free}}^{-1} \left( \frac{\partial \tilde{\mathbf{Q}}_{\text{free}}}{\partial q_i} \mathbf{X}_{\text{free}} + \frac{\partial \tilde{\mathbf{Q}}_{\text{link}}}{\partial q_i} \mathbf{X}_{\text{fix}} \right) \quad (13)$$

次に、荷重作用点の反力の 2 乗ノルムを  $q_i$  で偏微分すると、次式を得る。

$$\frac{\partial \sum_{i \in \mathfrak{R}} (R_i(\mathbf{q}) - \bar{R}_i)^2}{\partial q_i} = 2 \sum_{i \in \mathfrak{R}} (R_i(\mathbf{q}) - \bar{R}_i) \cdot \frac{\partial R_i(\mathbf{q})}{\partial q_i} \quad (14)$$

反力ベクトル  $\mathbf{R}$  の感度係数は、式(5)より、以下のように求められる。

$$\frac{\partial \mathbf{R}}{\partial q_i} = \frac{\partial \tilde{\mathbf{Q}}_{\text{link}}^T}{\partial q_i} \mathbf{X}_{\text{free}} + \tilde{\mathbf{Q}}_{\text{link}}^T \frac{\partial \mathbf{X}_{\text{free}}}{\partial q_i} + \frac{\partial \tilde{\mathbf{Q}}_{\text{fix}}^T}{\partial q_i} \mathbf{X}_{\text{fix}} \quad (15)$$

#### 6. 最適化手法

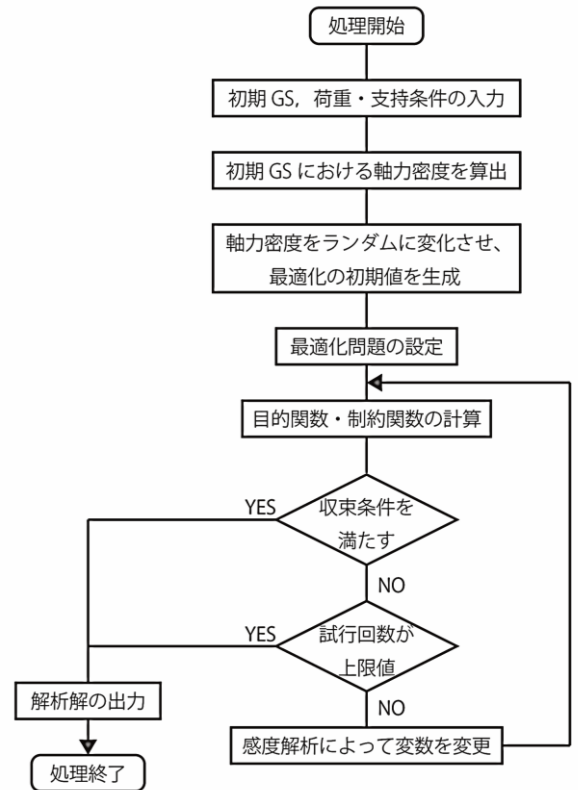


図 1: 最適化プログラムのフローチャート

最適化には逐次2次計画法のライブラリである NLOpt の SLSQP [11,12]を用いる。Grasshopper SDK を援用した C# によるプログラミングを行い、本手法をパッケージ化した Grasshopper コンポーネントを作成し、3D モデリングソフトである Rhinoceros の Grasshopper プラグイン上で解析を行う。

以下の例では、単位は重要ではないため省略する。ヤング係数、体積の上限値はそれぞれ 1.0, 10, スムージングのためのパラメータ  $c$  は  $1.0 \times 10^6$  とする。式(7)より、 $FV$  の値が一定なので、以降は  $FV/10$  の値を比較する。初期形状における基準解での値を  $\tilde{q}_i$  とし、 $\tilde{q}_i \pm 10$  の範囲で変数をランダムに 100 種類与える。それぞれについて最適化問題を解いて最も優良な解を選択する。なお、目的関数の計算回数の上限値は 150000 とし、上限値に達した場合はその時点での解析解を出力する。

## 7. 数値例題

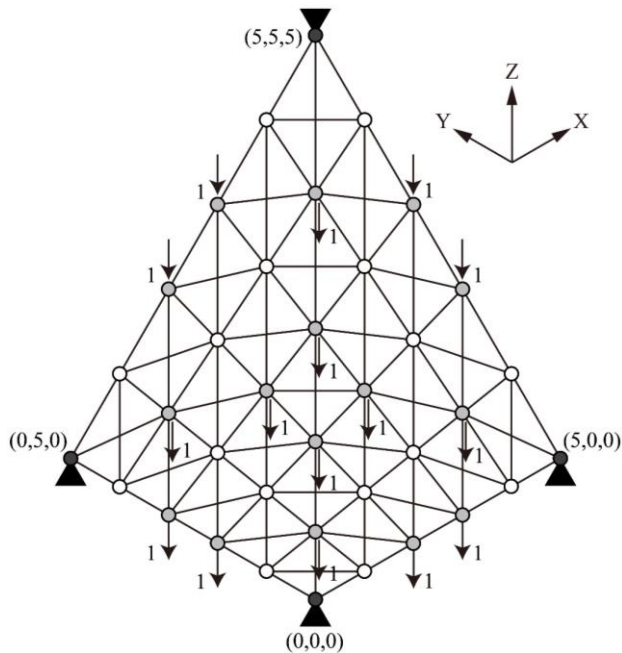


図 2：初期 GS として入力する 5×5 モデル

(0,0,0), (5,0,0), (5,5,5), (0,5,0) の 4 頂点で張られる極小曲面上に各節点をもつ 5×5 グリッドのラティスシェルを考える。部材数は 110 本である。4 頂点をピン支持し、(0,0,0), (5,5,5) を通る  $xy$  平面に垂直な面に関して対称になるように、鉛直下向きに大きさ 1 の荷重を 16 個の節点に作用させる。図 2 に初期形状を示しており、白色で塗りつぶされた 16 個の節点が最適化の途中で移動する自由節点である。

まず、100 種類のランダムシードに対して得られた初期解を用いて最適化問題(8)を解き、それらの最適解の中で最も優良な解を図 3 に示す。赤色は圧縮、青色は引張を表している。上部の支持点(5,5,5)から引張材が伸びており、下部の 3 つの支持点付近では圧縮材によるアーチが形成され

ている。複数の重複した節点が見られるが、それに伴う melting nodes による数値困難は発生しなかった。

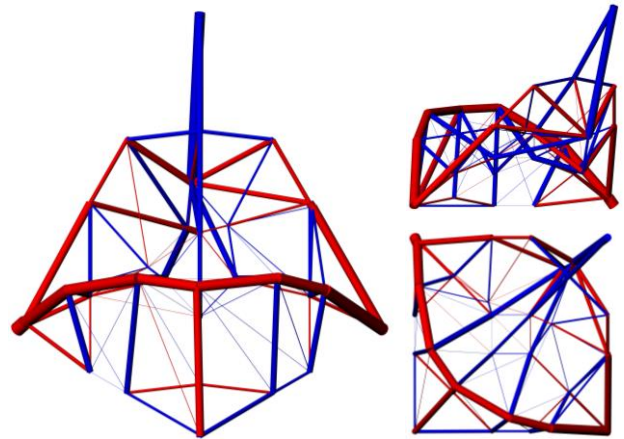


図 3：最優良解(追加制約なし,  $FV/10 = 1765.254$ )

次に、平面的に見て正方形グリッドを構成する(斜材でない) 60 部材の各部材長が 2 以下となるように、式(9)を用いて制約を与える。最適化問題(8)に式(9)の追加制約を課した問題を同様に 100 回解き、最も優良な解を図 4 に示す。

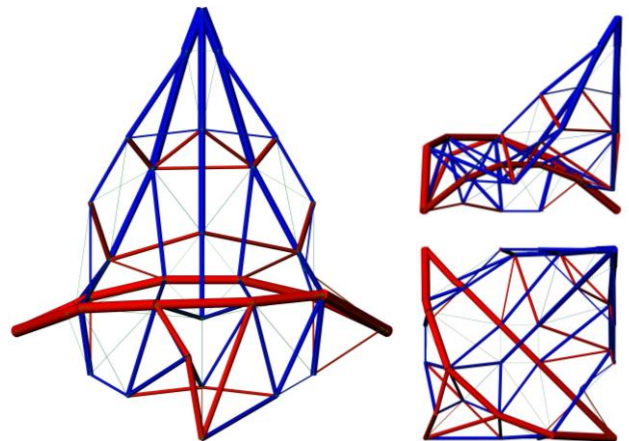


図 4：最優良解( $L < 2$ ,  $FV/10 = 1546.117$ )

部材長に制約を課さなかった場合(図 3)と比較すると、上部の支持点(5,5,5)から扇状に引張材が伸びており、重複した節点は少ない。より多くの部材で荷重を負担しようとした構造であることがわかる。また、節点に対称制約を課さなかったにも関わらず、アイソメトリックにはある程度の面対称性が確認できる。初期 GS の形状を比較的維持できているだけでなく、目的関数値のより小さな良好な解形状でもある。

さらに、部材長が 2 以下となる制約を課した最適化問題に、 $z$  座標を 2 次曲面  $z = xy/5$  に制約する条件を式(10)のように追加した場合の最優良解を、図 5 に示す。

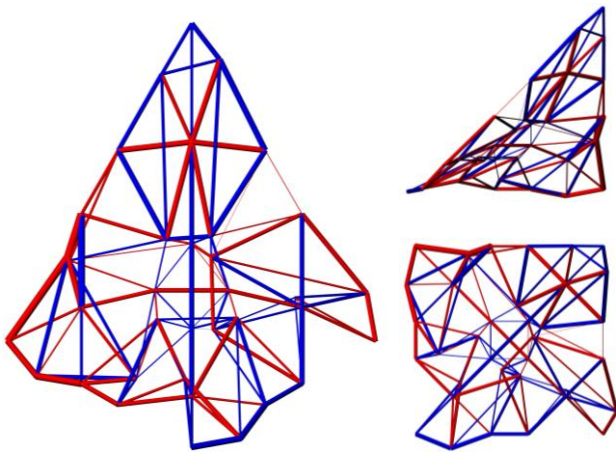


図 5 : 最優良解( $L < 2$ ,  $z = xy/5$ ,  $FV/10 = 17821.113$ )

曲面  $z = xy/5$  上に各節点が位置していることが確認できるが、 $FV/10$  の値が 10 倍に増えてしまった上に、前の 2 例に比べて軸力を他の部材にスムーズに伝達できないような滑らかでない部材配置となっているため、より多くのランダムシードについて最適化問題を解いて更なる優良解が得られる可能性は高いと考えられる。

5×5 モデルに適用した以上の 3 つの最適化問題について、100 回の試行で得られたコンプライアンスに関する主要値を表 1 に示す。

表 1 : コンプライアンス  $F$  の主要値

	normal	$\bar{L}_i = 2.0$	$\bar{L}_i = 2.0$ & $z$
max	$7.214 \times 10^{16}$	$7.214 \times 10^{16}$	$7.214 \times 10^{16}$
median	$2.719 \times 10^3$	$2.557 \times 10^3$	$2.174 \times 10^5$
min	$1.765 \times 10^3$	0.834 ※	$1.782 \times 10^4$
average	$7.216 \times 10^{14}$	$7.216 \times 10^{14}$	$7.215 \times 10^{14}$
st.dev	$7.178 \times 10^{15}$	$7.178 \times 10^{15}$	$7.178 \times 10^{15}$

注 : ※は制約条件を満たしておらず、不良な解である。

部材長に制約を課した場合では、課さなかった場合に比べて良好な目的関数値が得られた。しかし、 $z$  座標を曲面上に制約した場合、目的関数値のオーダーが他の 2 例に比べて大きい。

表 2 : 解の収束結果と最適化の所要時間

	normal	$\bar{L}_i = 2.0$	$\bar{L}_i = 2.0$ & $z$
converge	65	55	97
maxeval	32	12	0
failure	3	33	3
time (s)	1438.392	605.865	51.727

表 2 に、解の収束性に関するデータをまとめる。“converge”、“maxeval”、“diverge”は、それぞれ収束解、

目的関数を呼び出す回数が上限値に達した解、発散・制約条件を満たさない等により解析に失敗した解の個数を、“time”は 1 回の最適化に要する時間を示している。複雑な 5×5 モデルに対して比較的短い解析時間で最適化が終了した。制約条件が厳しくなればなるほど目的関数を最大回数(150000 回)呼び出すまでに解が改善されなくなり、また 1 回のサイクルに要する時間が短くなっている。

## 8. まとめ

本研究で得られた結論を以下に示す。

- (1) 複雑な立体トラスモデルについても、著者らが提案した軸力密度を設計変数とする位相・形状の同時最適化手法を用い、良好な収束解が得られることを確認した。
- (2) 本手法では、設計変数の数は部材数に一致し、節点位置を連続変数とすることができるので、最適化の計算負荷が既往の手法と比べて小さい。
- (3) 長さや節点位置に関する制約条件は、最適化における形状の大幅な変化を制御するのに有効であるが、有料でない局所最適解に収束する可能性が高くなる。
- (4) 本手法をパッケージ化した Grasshopper コンポーネントを作成し、Rhinoceros 上で作成したモデルの構造最適化を実行できた。

## [参考文献]

- 1) M. Ohsaki, Optimization of Finite Dimensional Structures, CRC Press, 2010.
- 2) M. P. Bendsøe and O. Sigmund, Topology Optimization: Theory, Methods and Applications, Springer, 2003.
- 3) W. R. Dorn, R. Gomory and H. Greenberg. Automatic design of optimal structures, J. de Mecanique; Vol. 3: 25–52, 1964.
- 4) W. Dobbs and L. P. Felton, Optimization of truss geometry, J. Struct. Div. ASCE, Vol. 95(ST10), pp. 2105–2119, 1969.
- 5) M. Ohsaki, Simultaneous optimization of topology and geometry of a regular plane truss, Comput. & Struct., Vol. 66(1), pp. 69–77, 1998.
- 6) W. Achtziger, On simultaneous optimization of truss geometry and topology, Struct. Multidisc. Optimiz., Vol. 33, pp. 285–304, 2007.
- 7) J. J. McKeown, Growing optimal pin-jointed frames, Struct. Opt., Vol. 15, pp. 92–100, 1998.
- 8) T. Hagishita and M. Ohsaki, Topology optimization of trusses by growing ground structure approach, Struct. Multidisc. Optimiz., Vol. 37, No. 4, pp. 377–393, 2009.
- 9) J. Y. Zhang and M. Ohsaki, Tensegrity Structures: Form, Stability, and Symmetry, Mathematics for Industry 6, Springer, 2015.
- 10) 林 和希, 大崎 純, 軸力密度を設計変数とする平面トラスの位相と形状の同時最適化, コロキウム構造形態の解析と創生 2016, 日本建築学会, pp. 141–146, 2016.
- 11) D. Kraft, A software package for sequential quadratic programming, Technical Report DFVLR-FB 88-28, 1988.
- 12) S. G. Johnson, The NLOpt nonlinear-optimization package, <http://ab-initio.mit.edu/nlopt>

\*1 京都大学大学院工学研究科 修士課程

\*2 京都大学大学院工学研究科 教授 博士(工学)