Python の最適化ライブラリを用いた様々な構造最適化の実例

○藤田 慎之輔[*1]　寒野 善博[*2]　大崎 純[*3]

キーワード：構造最適化　重量言語と軽量言語　Python　最適化ライブラリ

1　はじめに

近年，建築構造物の設計プロセスに最適化手法を導入した研究は盛んに行われているが，まだ実務の世界において一般的に使われるまでには至っていない。最適化手法を理解し，建築構造デザインへ応用するためには，数学的な素養とともにプログラミング技術が不可欠となる。プログラミング言語は，大きく分けて重量言語と軽量言語の 2 つに分類される (表 1)。

表 1　重量言語と軽量言語

	代表的な言語	計算速度	コンパイル	言語ルール
重量言語	Fortran, C++ など	速い	必要	煩雑
軽量言語	Perl, Python, Ruby など	遅い	不要	簡潔

建築の分野で最適化手法を扱う研究者のほとんどは，重量言語により最適化プログラムを開発している。しかし，科学技術計算の知識に乏しい初学者にとってそれらの重量言語は習得に時間を要し，このことは建築の世界において最適化手法を扱うことのできる実務家が不足している一因となっている。

一方で軽量言語は，初学者が短期間で習得可能であるため，教育用のプログラミング言語として有用であると考えられる。中でも Python は，非常に分かりやすい言語ルールと高い汎用性を持ち，加えてフリーでオープンソースであることから近年利用者が急増しており，膨大な数のサードパーティライブラリが存在し，最適化手法も豊富なライブラリにより容易に利用することができる。一般に軽量言語は計算が遅く，これは科学技術計算において致命的な欠点となりうるが，Fortran や C 言語で作られた Python ライブラリを利用することでこの欠点は回避することができるため，教育用だけでなく，研究用，実務用としても十分活用可能な言語である。事実，NASA，Google，KEK などがプログラム開発に Python を取り入れている。

しかしながら，日本で Python を用いて建築構造物の最適化を行っている研究者は多くなく，最適化ライブラリもほとんど利用されていない。本論文では，Python の様々な最適化ライブラリを紹介し，それらを用いることで，建築構造に関する様々な最適化問題を容易に解くことができることを数値解析例を通じて示す。フリーかつオープンソースの Python の最適化ライブラリを活用することで，初学者が建築構造の世界における最適化手法に対してなじみやすい環境を構築することを目指す。

2　Python の最適化ライブラリの紹介

Python には，科学技術計算に関する膨大なライブラリが存在し，現在も世界中の技術者により開発が続けられ，その数は増え続けている。その中で，建築の構造最適化を行うにあたり有用な最適化手法に関するライブラリとしては，以下のものが挙げられる。

表 2　代表的な Python 最適化ライブラリ

最適化手法の種類とアルゴリズム		主な対応ライブラリ
数理計画法	滑降シンプレックス法	SciPy[1)]
	Powell 法	SciPy
	共役勾配法	SciPy
	準ニュートン法	SciPy
	Newton-CG 法	SciPy
	記憶制限準ニュートン法	SciPy
	COBYLA 法	SciPy, pyOpt[2)]
	逐次最小二乗法	SciPy, pyOpt
	逐次二次計画法	pyOpt
	拡張ラグランジュ関数法	pyOpt
	CONMIN 法	pyOpt
	KS 汎関数法	pyOpt
	フィルタ信頼領域法	pyOpt
	ペナルティ法	pyOpt
	改良型 Shor's r-algorithm	pyOpt
	線形計画問題一般	PuLP[3)] , PICOS*
	二次計画問題一般	PICOS
	二次制約二次計画問題一般	PICOS
	二次錐計画問題一般	PICOS
	半正定値計画問題一般	PICOS
	幾何計画問題一般	PICOS
	混合整数計画問題一般	PICOS
発見的手法	粒子群最適化法	pyOpt, PyGMO[4)]
	ハーモニーサーチ	pyOpt, PyGMO
	遺伝的アルゴリズム	pyOpt, PyGMO, DEAP[5)]
	蟻コロニー/蜂コロニー最適化	pyOpt, PyGMO
	力まかせ探索	SciPy
	ベイズンホッピング	SciPy, PyGMO
	差分進化	SciPy, PyGMO
	擬似焼きなまし法	SciPy, PyGMO
	モンテカルロ法	PyGMO
	共分散行列適応進化戦略	PyGMO

※ PICOS[6)] 自身は最適化ソルバーではなく，Python 経由で CPLEX, Gurobi Optimizer, MOSEK などの外部ソルバーを利用可能にするモデラーである

上記ライブラリはすべてライセンスフリーである。ここに紹介したライブラリ以外にも数多くの最適化ライブラリが存在し，数理計画法から発見的手法まで様々なアルゴリズムが網羅されている。SciPy，pyOpt，PyGMO，DEAP，PuLP，PICOS のいずれについても，公式サイトに詳細な documentation が用意されているので参考にされたい [7–12)]。SciPy の Newton-CG 法のソルバーである fmin_ncg のように，ピュアな Python コードで書かれているため計算速度の遅いものも中には存在するが，ほぼ全てのソルバーは Fortran または C 言語で記述されており，高速に実行可能である。

なお，各ライブラリの公式 HP によれば，最適化アルゴリズムを格納した関数群の呼称は，Optimizer あるいは method などとそれぞれのライブラリごとに異なるが，本論文ではすべて「ソルバー」という呼称に統一する。

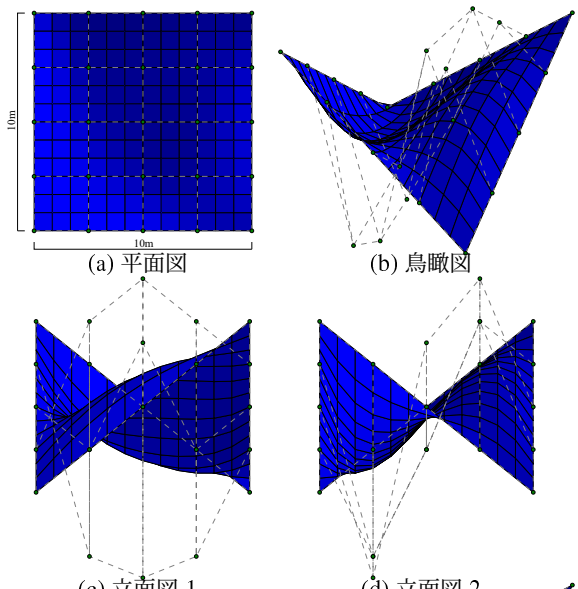
3　SciPy による最適化例—曲面積最小化問題—

曲面をテンソル積ベジエ曲面で記述し，曲面の外周を除いた制御点座標を設計変数とし，曲面積を最小化する次の無制約単一目的最適化問題を考える。

$$\text{minimize} \quad A(\boldsymbol{q}_z^*) = \int_0^1 \int_0^1 \sqrt{EG - F^2}\,du\,dv \qquad (1)$$

ここで $u, v \in [0, 1]$ はテンソル積ベジエ曲面のパラメータ，$\boldsymbol{q}_z^*$ は外周を除いた制御点 z 座標，E, F, G は曲面の第 1 基本量である [13]。膜構造の設計の初期段階では，等張力条件を満足する膜曲面形状が設計の基本曲面となるが，これは極小曲面と等価な形状となることが知られている。問題 (1) を解くことで，テンソル積ベジエ曲面の形状表現の自由度の範囲内で近似的に極小曲面を創生することができる。

例として，10m×10m の正方形平面を有し，4 つの直線で囲まれた境界を持ち，ランダムな形状を有するテンソル積ベジエ曲面 (図 1) を初期形状として最適化を行う。ベジエ曲面の制御ネットは 5×5，設計変数の数は 9 である。SciPy は，無制約最適化手法の多くのソルバーを有するため，問題 (1) を解くのに適している。本論文では，SciPy の fmin (Powell 法)，fmin_cg (共役勾配法)，fmin_bfgs (準ニュートン法) の 3 つのソルバーを用いて最適化を行った。

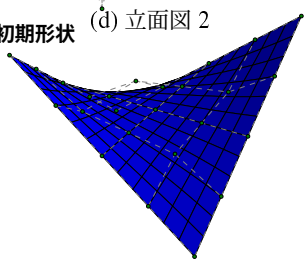


(a) 平面図　(b) 鳥瞰図　(c) 立面図 1　(d) 立面図 2

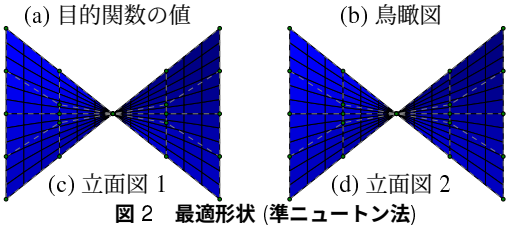
図 1　初期形状

	初期形状	最適形状
$A(\boldsymbol{q}_z^*)$[m^2]	140.67	118.86

(a) 目的関数の値



(b) 鳥瞰図



(c) 立面図 1　(d) 立面図 2

図 2　最適形状 (準ニュートン法)

準ニュートン法により最適化を行った結果を図 2 に示す。また，3 つのソルバーそれぞれの最適化過程における目的関数値の履歴を図 3 に示す。いずれのソルバーを用いても得られた解はほぼ同一の形状となった。目的関数の値もすべてのソルバーで $A = 118.86\text{m}^2$ となった。

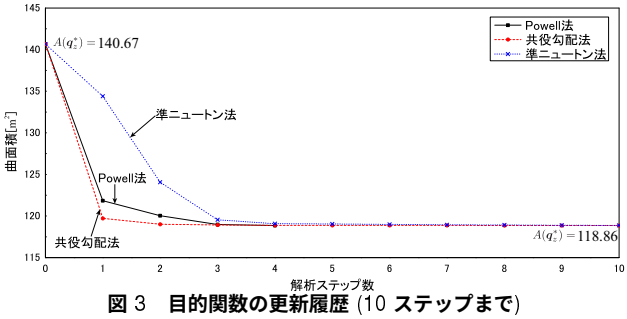


図 3　目的関数の更新履歴 (10 ステップまで)

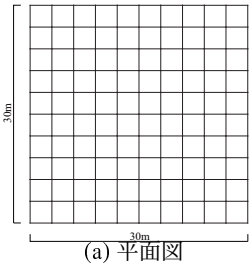
なお，最適化計算部分のプログラムは

optimize. **ソルバー名 (目的関数, 設計変数, オプションパラメータ)**

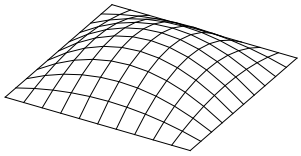
のように簡潔に記述される。ソルバー名の部分を，fmin, fmin_cg, fmin_bfgs のように記述するだけで様々なソルバーを利用することが可能である。

4　pyOpt による最適化例—ラチスシェルの構造最適化—

図 4 に示すような 30m×30m の正方形平面を有するラチスシェルを対象とした最適化を，非線形計画問題に対する豊富なソルバーを有するライブラリである pyOpt を用いて行う。



(a) 平面図



(b) 鳥瞰図

(c) 立面図

断面 (一様)	ヤング率 [GPa]	単位体積重量 [kN/m^2]	境界条件	外力
○-267.4x9	210	77	4隅固定	自重のみ

(d) 形状諸元

図 4　初期形状

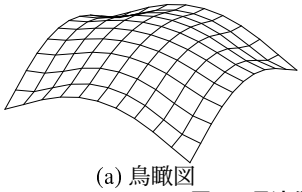
4.1　ひずみエネルギー最小化問題

ラチスシェルの古典的な構造最適化問題としては，ひずみエネルギー最小化による剛性最大化曲面の創生がある。節点の鉛直方向座標を設計変数としたひずみエネルギー最小化問題は次のように定式化される [14]。

$$\text{minimize} \quad f(\boldsymbol{r}_z^*) = \frac{1}{2}\boldsymbol{d}^\top \boldsymbol{K}\boldsymbol{d} \qquad (2a)$$

$$\text{subject to} \quad g_1(\boldsymbol{r}_z^*) = \sum_{k=1}^{m} l_k \le \bar{L} \qquad (2b)$$

ここで $\boldsymbol{r}_z^*$ は支点を除く節点の z 座標を並べたベクトル，f はひずみエネルギー，$\boldsymbol{d}$ は節点変位ベクトル，$\boldsymbol{K}$ は弾性剛性マトリクスである。制約 (2b) は総部材長制約であ



(a) 鳥瞰図

(b) 立面図

	$f(\boldsymbol{r}^*)$ [kNm]	$g_1(\boldsymbol{r}^*)$ [m]
初期形状	15.12	691.7
最適形状	2.665	691.7

(c) 物理諸量

図 5　最適化結果 (PSQP)

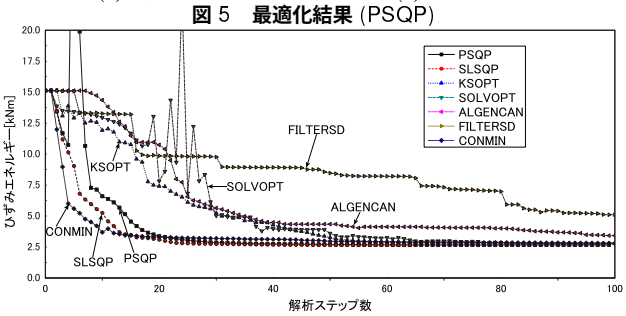


図 6　目的関数の更新履歴 (100 **ステップまで**)

り，m は部材数，l_k は k 番目の部材の部材長，$\bar{L}$ は総部材長の上限値であり，ここでは初期形状での値を指定する。断面一定の場合，制約 (2b) は体積制約と同義である。問題 (2) のようなひずみエネルギー最小化問題を解く際には，節点変位を求めるために大規模疎行列である $\boldsymbol{K}$ を含む連立方程式を繰り返し解く必要があるが，SciPy のスパースソルバーなどのライブラリを利用することで高速に計算できる。

pyOpt の PSQP (逐次二次計画法)，SLSQP (逐次二次計画法)，KSOPT (KS 汎関数法)，SOLVOPT (改良型 Shor's r–アルゴリズム)，ALGENCAN (拡張ラグランジュ関数法)，FILTERSD (フィルター信頼領域法)，CONMIN (CONMIN 法) の 7 つのソルバーを用いて最適化を行った (PSQP と SLSQP はいずれも逐次二次計画法のソルバーであるが，二次計画問題を解く際の解法や前処理操作などがそれぞれ異なる)。PSQP により最適化を行った結果を図 5 に示す。また，7 つのソルバーのそれぞれの最適化過程における目的関数値の履歴を図 6 に示す。いずれのソルバーを用いても得られた解はほぼ同一の形状となった。目的関数値もすべてのソルバーの解で $f = 2.665$kNm となった。ただし，ソルバーによって収束性は大きく異なり，ALGENCAN と FILTERSD を用いた場合には 100 ステップでは収束せず，多くの繰り返し計算を必要とした。

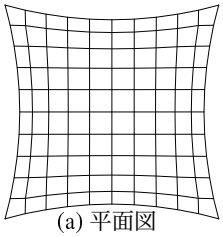
4.2　部材長一様化制約の考慮

最適化問題 (2) に部材長一様化制約を導入することで施工性を考慮できる。部材長を一様化する場合，設計変数は節点の鉛直方向座標だけでは不十分であり，平面方向にも形状を修正する必要がある。そのような問題は次のように定式化される [14]。

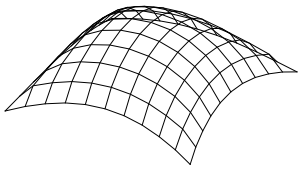
$$\text{minimize} \quad f(\boldsymbol{r}^*) = \frac{1}{2}\boldsymbol{d}^\top \boldsymbol{K}\boldsymbol{d} \tag{3a}$$

$$\text{subject to} \quad g_1(\boldsymbol{r}^*) = \sum_{k=1}^{m} l_k \leq \bar{L} \tag{3b}$$

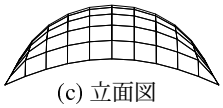
$$g_2(\boldsymbol{r}^*) = \sum_{k=1}^{m} (l_k - l_{\text{ave}})^2 \leq \varepsilon \tag{3c}$$



(a) 平面図



(b) 鳥瞰図



(c) 立面図

	$f(\boldsymbol{r}^*)$ [kNm]	$g_1(\boldsymbol{r}^*)$ [m]	$g_2(\boldsymbol{r}^*)$ [m^2]	$l_{\max}$ [mm]	$l_{\min}$ [mm]
初期形状	15.12	691.7	7.757	3697	3000
最適形状	0.5455	691.7	1.0×10^{-4}	3146	3144

(d) 物理諸量

図 7　最適形状 (PSQP)

ここで $\boldsymbol{r}^*$ は支点を除く節点の x, y, z 座標を並べたベクトルである。g_2 は部材長偏差で，この値が 0 になった時に部材長は一様化される。解析上は，ε に十分小さな値を指定することで部材長の一様性を確保する。l_{ave} は平均部材長である。

部材長一様性確保のための指定値を $\varepsilon = 1.0 \times 10^{-4}$ とし，前述の 7 つのソルバーを用いて最適化を行ったが，最適解が得られたのは，PSQP, SLSQP, ALGENCAN, FILTERSD のみで，その他のソルバーについては実行可能解が得られなかった。これは，制約 (3c) は非線形性が強く，かつ非常に厳しい幾何学的制約であり実行可能領域が限定されたためと考えられる。PSQP により最適化を行った結果を図 7 に示す。ここで，図中 (d) における $l_{\max}, l_{\min}$ はそれぞれ初期形状/最適形状における部材長さの最大値および最小値を表す。3 つのソルバーについては解が得られなかったが，解を得ることのできた 4 つのソルバーについては，いずれも得られた解はほぼ同一の形状となった。目的関数値もすべて $f = 0.5455$kNm となった。初期形状において，$l_{\max}$ と $l_{\min}$ の差はおよそ 697mm であったが，最適形状においてはその差は 2mm となっており，部材長がほぼ完全に一様化されていることが確認できる。

なお，最適化計算部分のプログラムは

```
optmethod = ソルバー名 ()
optmethod.setOption(ソルバー依存のオプションパラメータ)
opt_prob = Optimization('目的関数名', 目的関数)
optmethod(opt_prob, 各ソルバー共通のオプションパラメータ)
```

のように簡潔に記述される。ソルバー名の部分を PSQP, SLSQP, KSOPT, SOVLOPT, ALGENCAN, FILTERSD, CONMIN のように記述するだけで様々なソルバーを利用することが可能である。

5　PICOS による最適化例—平面骨組のトポロジー最適化—

部材断面の離散性を考慮した平面骨組のトポロジー最適化問題は，既往の研究では発見的手法によるものや混合整数計画法 (MIP) を用いた手法が提案されている。後者の方法による場合，平面骨組のコンプライアンス最小化問題は，次のような混合整数二次錐計画問題 (MISOCP)

として定式化できることが知られている[15]。

$$\text{minimize} \quad 2\sum_{e=1}^{m}\sum_{l=1}^{3} w_{el} \tag{4a}$$

$$\text{subject to} \quad w_{e1} + a_e \geq \left\| \begin{bmatrix} w_{e1} - a_e \\ \sqrt{\dfrac{2l_e}{E}} s_{e1} \end{bmatrix} \right\|, \ e = 1, \cdots, m \tag{4b}$$

$$w_{e2} + I_e \geq \left\| \begin{bmatrix} w_{e2} - I_e \\ \sqrt{\dfrac{6l_e}{E}} s_{e2} \end{bmatrix} \right\|, \ e = 1, \cdots, m \tag{4c}$$

$$w_{e3} + I_e \geq \left\| \begin{bmatrix} w_{e3} - I_e \\ \sqrt{\dfrac{2l_e}{E}} s_{e3} \end{bmatrix} \right\|, \ e = 1, \cdots, m \tag{4d}$$

$$\boldsymbol{p} = \sum_{e=1}^{m}\sum_{l=1}^{3} q_{el}\boldsymbol{b}_{el} \tag{4e}$$

$$\sum_{e=1}^{m} l_e a_e \leq \bar{V} \tag{4f}$$

$$a_e = \sum_{j=1}^{r} \bar{a}_j t_{ej}, \ e = 1, \cdots, m \tag{4g}$$

$$I_e = \sum_{j=1}^{r} \bar{I}_j t_{ej}, \ e = 1, \cdots, m \tag{4h}$$

$$t_{ej} \in \{0, 1\}, \quad e = 1, \cdots, m; \quad j = 1, \cdots, r \tag{4i}$$

$$\sum_{j=1}^{r} t_{ej} \leq 1, \ e = 1, \cdots, m \tag{4j}$$

ここで，E はヤング係数，$l_e, a_e, I_e (e{=}1, \cdots, m)$ は部材の自然長，断面積，断面 2 次モーメントである。r は断面の種類数であり，部材の断面性能は r 個の断面性能の組 $(\bar{a}_1, \bar{I}_1), \cdots, (\bar{a}_r, \bar{I}_r)$ から選択される。$\bar{V}$ は体積の上限値であり，設計変数は $t_{ej}, w_{el}, s_{el} (e{=}1, \cdots, m \ j{=}1, \cdots, r \ l{=}1, 2, 3)$ である。t_{ej} は部材選択のための 0-1 変数，w_{el} は最適化のための補助変数，s_{el} は一般化応力である。

例として，図 8 のように，スパン中央に 100kN の集中荷重が作用する 4m×2m の単純支持の骨組が，4 スパン×2 スパンのグランドストラクチャとして構成されている場合の最適トポロジーを考える。

ヤング係数は E=210GPa, 体積の上限値は $\bar{V}$=0.00275m^3 とする。選択する部材は円形鋼管とする。鋼管の外形を R_e，内径を $r_e = cR_e$ とすると，e 番目の断面性能の組は

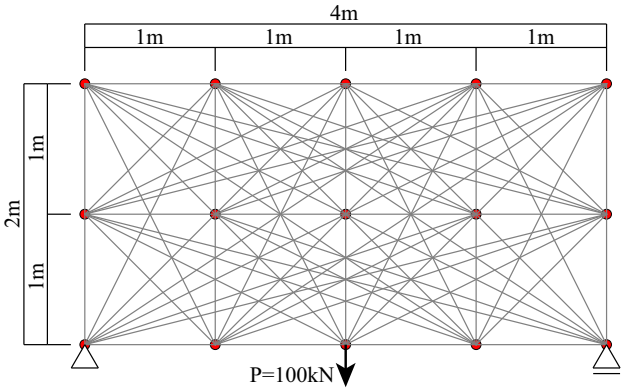


図 8　中央集中荷重を受ける単純支持骨組のグランドストラクチャ

次式で計算される。

$$\bar{a}_e = \pi(1 - c^2)R_e^2, \quad \bar{I}_e = \frac{\pi}{4}(1 - c^4)R_e^4 \tag{5}$$

このとき，$\bar{I}_e$ は $\bar{a}_e$ の関数として次式で表される。

$$\bar{I}_e = \bar{\gamma}\bar{a}_e^2 \tag{6a}$$

$$\bar{\gamma} = \frac{1}{4\pi}\frac{1 + c^2}{1 - c^2} \tag{6b}$$

本論文では，$c = 0.9$ とし，断面積のリスト $\bar{\boldsymbol{a}} = (\bar{a}_1, \cdots, \bar{a}_r)$ が次のケースの場合について最適化を行う。

ケース 1 $(r = 1)$: $\bar{\boldsymbol{a}} = \begin{pmatrix} 0.0010 \end{pmatrix}$
ケース 2 $(r = 1)$: $\bar{\boldsymbol{a}} = \begin{pmatrix} 0.0020 \end{pmatrix}$
ケース 3 $(r = 2)$: $\bar{\boldsymbol{a}} = \begin{pmatrix} 0.0010 & 0.0020 \end{pmatrix}$
ケース 4 $(r = 3)$: $\bar{\boldsymbol{a}} = \begin{pmatrix} 0.0010 & 0.0015 & 0.0020 \end{pmatrix}$
ケース 5 $(r = 6)$: $\bar{\boldsymbol{a}} = \begin{pmatrix} 0.0010 & 0.0012 & \cdots & 0.0018 & 0.0020 \end{pmatrix}$
ケース 6 $(r = 11)$: $\bar{\boldsymbol{a}} = \begin{pmatrix} 0.0010 & 0.0011 & \cdots & 0.0019 & 0.0020 \end{pmatrix}$

凸計画問題のモデラーである PICOS を用いて各ケースについてトポロジー最適化を行った。PICOS では MISOCP のソルバーをいくつか指定することになるが，本研究では MOSEK を指定した。最適化結果を図 9 に示す。ここで，図中の部材の線幅は，断面積の大きさを表す。最適解はいずれも体積制約を満足しつつ，コンプライアンスの小さい部材配置が得られており，問題 (4) を PICOS を用いて解くことができた。ただし，ケース 1，5，6 の最適解において，部材の交差が発生していることがわかる。実際の構造物を考えたときには部材の交差は許容されないので，これらの最適解は現実的ではない。

そこで，MISOCP として定式化した問題 (4) に，部材の交差を回避する条件を考慮することを考える。図 8 のグランドストラクチャにおいて交差する部材の組数を q，部材番号の組を

$$\boldsymbol{C}_q = \begin{pmatrix} \begin{pmatrix} C_{11} & C_{12} \end{pmatrix} & \cdots & \begin{pmatrix} C_{q1} & C_{q2} \end{pmatrix} \end{pmatrix} \tag{7}$$

とおく。このとき，問題 (4) に次の制約を加えることで，部材の交差を回避することができる。

$$\sum_{j=1}^{r} t_{\alpha j} + \sum_{j=1}^{r} t_{\beta j} \leq 1, \qquad \begin{matrix} \alpha = C_{k1}, & \beta = C_{k2} \\ k = 1, \cdots, q \end{matrix} \tag{8}$$

制約 (8) を考慮した最適化結果を図 10 に示す。コンプライアンスは，交差条件を考慮しない最適解よりはやや大きいものの小さく抑えられており，部材の交差が生じていないことが確認できる。なお，最適化計算部分のプログラムは

```
Problem=pic.Problem()
Problem.add_variable(設計変数)
Problem.add_constraint(制約条件)
Problem.add_objective(目的関数)
Problem.solve(solver=ソルバー名, オプションパラメータ)
```

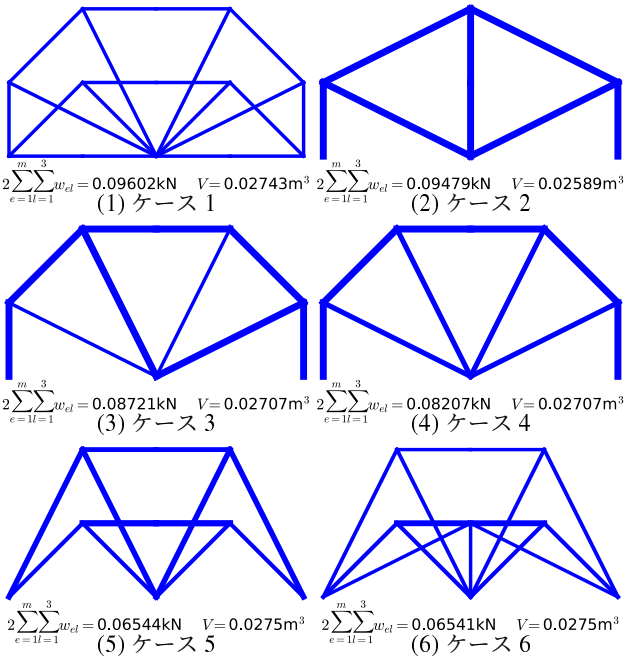


図 9　最適形状 (部材交差許容)

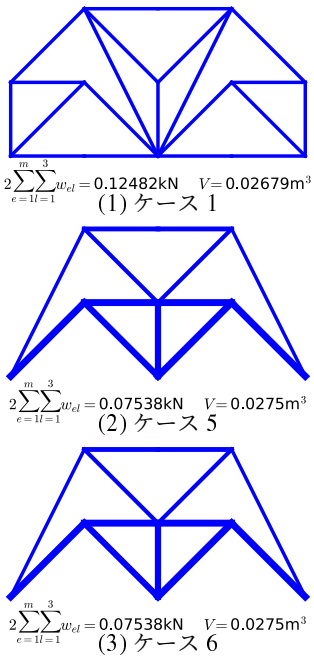


図 10　最適形状 (部材交差回避)

のように簡潔に記述される。ソルバー名の部分を，mosek, gurobi, cplex のように記述するだけで様々なソルバーを利用することが可能である。

6　まとめ

本論文では，連続変数問題から離散変数問題に至るまで様々な種類の構造最適化問題を，Python の最適化ライブラリのみを利用することで容易に解くことが可能であることを数値解析例を通じて示した。3 節では，SciPy を用いた最適化により，極小曲面を生成することができた。4 節では，pyOpt を用いた最適化により，剛性の高いラチスシェルが得られ，部材長の一様化も実現された。5 節では，PICOS を用いた最適化により，高い剛性を有する平面骨組のトポロジーが得られた。

Python は軽量言語であり，シンプルで分かりやすい言語ルールを有するため，初学者にも習得が容易であり，プログラムのソースコードを非常に簡潔なものとすることができる。本論文で紹介したもの以外にも数多くのライブラリが存在し，あらゆる最適化手法のアルゴリズムを幅広く網羅しており，その多くがフリーかつオープンソースで改変自由である。

Python では，SciPy や pyOpt のように一つのライブラリの中に様々な最適化ソルバーを有するライブラリを利用することで，同一のインターフェースで複数の最適化手法を手軽に比較検討することができる。ソルバーによっては問題 (3) の最適解が得られなかったことからも分かるとおり，最適化手法は，解く問題の規模や種類によって解析の安定性や解の収束性は大きく異なるため，複数の最適化手法を手軽に比較検討できるこの性質は，特に研究や開発を行う上で非常に重要である。

Python はその手軽さから，科学技術計算教育用のプログラミング言語として近年アメリカで急速に利用者が増えている言語であるが，教育用途のみならず，研究用，実務用としても十分に活用できるものであると考えられる。

[参考文献]

1）T. E. Oliphant: Guide to NumPy, Createspace Independent Publishing Platform, 2015
2）R. E. Perez, P. W. Jansen, and J. R. R. A. Martins: pyOpt: A Python-based object-oriented framework for nonlinear constrained optimization, Struct. Multidisc. Optim., Vol. 45, pp 101–118, 2012
3）S. Mitchell: An Introduction to pulp for Python Programmers, The Python Papers Monograph, Vol. 1, 2009
4）D. Izzo: PyGMO and PyKEP: Open Source Tools for Massively Parallel Optimization in Astrodynamics (the case of interplanetary trajectory optimization), Proceedings of the International Conference on Astrodynamics Tools and Techniques - ICATT, 2012
5）F. A. Fortin, F. M. De Rainville, M. A. Gardner, M. Parizeau and C. Gagne: DEAP: Evolutionary Algorithms Made Easy, Journal of Machine Learning Research, No. 13, pp. 2171–2175, 2012
6）G. Sagnol: Picos, a python interface to conic optimization solvers, Technical report 12–48. ZIB: http://picos.zib.de, 2012
7）SciPy documentation: http://docs.scipy.org/doc/scipy/reference
8）pyOpt documentation: http://www.pyopt.org/contents.html
9）PyGMO documentation: http://esa.github.io/pygmo/documentation/index.html
10）DEAP documentation: https://deap.readthedocs.io/en/master
11）PuLP documentation: http://pythonhosted.org/PuLP
12）PICOS documentation: http://picos.zib.de
13）小林昭七：曲線と曲面の微分幾何，裳華房, 1995
14）藤田慎之輔, 大崎純：パラメトリック曲面で定義されたラチスシェルの部材長一様化と剛性最大化を目的とした形状最適化，日本建築学会構造系論文集, No. 685, pp. 495–502, 2013
15）Y. Kanno: Mixed-integer second-order cone programming for global optimization of compliance of frame structure with discrete design variables, Struct. Multidisc. Optim., Vol. 53, pp. 1–16, 2015

*1　東京工業大学未来産業技術研究所助教博士 (工学)
*2　東京工業大学未来産業技術研究所准教授博士 (工学)
*3　京都大学大学院工学研究科建築学専攻教授博士 (工学)

Various Structural Optimizations Using Python Optimization Libraries

○ Shinnosuke FUJITA [*1]　Yoshihiro KANNO [*2]　Makoto OHSAKI [*3]

Keywords：Structural optimization　CPL&SPL　Python　Optimization library

In recent years, script programming languages (SPLs) are becoming popular in various engineering applications. These languages are easy to learn and use, because they are processed by interpreter and no compilation is necessary. However, in scientific computing, it is still common to develop a software using Fortran, C++, etc., which are categorized as compiler programming languages (CPLs), because SPLs usually have serious disadvantage in the computation speed. Although CPLs generally have advantage in computation, they are difficult to learn for the beginner such as undergraduate students who are not familiar with scientific computing. To attract more students to scientific computing, SPL is very effective for educational purpose. Python is an SPL, which has many license-free libraries for scientific computing, because the license of Python has no strict restriction for use and redistribution. Owing to the variety of libraries for numerical analysis, linear algebra, computer graphics, structural analysis, and optimization, Python is becoming one of the most popular language for structural optimization.

In this paper, we demonstrate that very simple programs coded by Python can solve various optimization problems. The various structural optimization problems are solved by simple Python prgramming using SciPy, pyOpt, and PICOS. First, we consider the shape optimization problem which minimizes the area of the tensor product Bézier surface. Optimal solution of the problem is able to be found easily by using SciPy. Next, we consider the shape optimization problem which minimizes the strain energy of the latticed shell with 4 corner fixed supports. Optimal solution of the problem is also able to be found easily by using pyOpt. Finally, we consider the topology optimization problem which minimizes the compliance of the 2D frame structure. Optimal solution of the problem is also able to be found easily by using MOSEK through PICOS.

The conclusions obtained in this paper are as follows:

- Python is suitable language for scientific education because of the characteristics shown below:
  - Simple cording rule
  - Compilation is unnecessary
  - Type declaration is unnecessary
  - Powerful visualization libraries
  - Many applications support Python scripting
- Easy to solve various optimization problem using Python
  - A whole range of optimization method are supported in Python libraries
  - Calculation speed is not slow by using scientific library compiled in CPL

Python is useful language not only for education but also for scientific research. Additionally, Python is open-source software and anyone can freely modify or customize it. So Vast number of libraries for scientific computing are created by programmers, engineers, or scientist around the world. Python is a language with high future growth potential.

*1　Assistant Professor, Laboratory for Future Interdisciplinary Research of Science and Technology, Tokyo Institute of Technology, Dr. Eng.
*2　Associate Professor, Laboratory for Future Interdisciplinary Research of Science and Technology, Tokyo Institute of Technology, Dr. Eng.
*3　Professor, Department of Architecture and Architectural Eng., Kyoto Univ., Dr. Eng.