

動的計画法を用いたラーメン構造の形状最適化

○林 和希*¹ 大崎 純*²

キーワード：ラーメン構造 建築構造最適化 多目的最適化 動的計画法 価値反復 焼きやまし法

1. 序

一貫構造計算プログラムは当初整形な建物を対象としていたが、コンピュータの高性能化に伴い、複雑な立体骨組モデルの解析も可能となっている[1]。しかし、幾何学的に不整形な骨組の可能性も検討する必要がある初期設計プロセスでの形状探索手法は十分に確立されていないといえる。

形状最適化は、節点位置を移動させることで目的に適った形状を得るための手法である。形状最適化を利用すれば、建築設計の初期において不整形な形も含めて多様な構造形態を考慮することができる。

形状最適化問題の解法としては、主に数理計画法を用いた手法が数多く提案されてきた[2]。しかし、数理計画法は目的関数・制約関数の勾配を必要とするため、変数が離散的である場合や、関数の非線形性が強い場合には適用が困難になる。

一方で、形状を最適化する際に、目的関数・制約関数の勾配情報を必要としない局所探索に基づくメタヒューリスティクスを用いた手法も提案されてきた[3]。しかし、これらの確率的近似解法は、一般に、近傍探索において変数の変化と目的関数・制約関数の変化の関係が十分に考慮されていない。

本報では、コンプライアンスの最小化と節点位置に応じた報酬の最大化を目的としたラーメン構造の形状最適化問題に対して、メタヒューリスティクスの一種である焼きやまし法[4]をベースとした新たな最適化手法を提案する。節点位置をグリッド空間を用いて離散化し、動的計画法[5]により各節点位置の状態価値を学習する。学習結果を利用して、近傍解を生成する際に用いる節点移動の方策を決定する。

本手法の特徴は、メタヒューリスティクスによる最適化において、エージェントを用いて変数の変更プロセスを制御することにある。エージェントの最適方策は目的関数のグリッド空間内に報酬を定義するだけで容易に学習させることができ、意匠的に計画したヴォリュームに対応した報酬を設定することで、デザイン要件を満たす構造形態が得られる。

2. 動的計画法

動的計画法 (Dynamic Programming: DP)は、Bellman方程式を解いて最適化問題を解く手法である[6]。まず、動

的計画法で用いる用語について説明する。

あるタスクについて学習を行う主体をエージェントと呼ぶ。エージェントは、ある状態 s において方策 π にしたがって行動 a を選択し、次状態 s' と報酬 r を観測する。エージェントの目的は、このような過程を繰り返して得られる累積報酬の最大化であり、動的計画法によって累積報酬を最大化するエージェントの最適方策を獲得することができる。方策は、現在の状態から累積することを期待する累積報酬に応じた価値関数と呼ばれる関数 (収益) に基づいて改善される。

現状態 s において方策 π に従ったときに得られる期待収益を状態価値関数 $V^\pi(s)$ とし、次式で定義する。

$$V^\pi(s) = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s \right\} \quad (1)$$

ここで、 t は現在のステップ、 $\gamma \in [0,1]$ は割引率と呼ばれ、1に近いほど累積報酬を、0に近いほど即時報酬を重視するパラメータである。また、 E_π は π に関して平均値をとるオペレータである。

状態 s で行動 a をとったとき、次状態 s' に遷移する確率を $p_{ss'}^a$ 、また、そのときに得られる報酬を $R_{ss'}^a$ とする。

Bellman 方程式は、任意の方策 π と状態 s に対して、 s の価値と可能な後続状態群との間に成立する最適性の必要条件であり、任意の状態価値関数について次のように定式化できる[6]。

$$V^\pi(s) = \sum_a \pi(s,a) \sum_{s'} p_{ss'}^a \left[R_{ss'}^a + \gamma V^\pi(s') \right] \quad (2)$$

ここで、 $\pi(s,a)$ は状態 s で行動 a を選択する確率、 $\gamma V^\pi(s')$ は次状態で期待される割引された状態価値である。エージェントの目的は収益の最大化であるから、方策空間から次式を満たす方策 π^* を探せばよい。

$$V^*(s) = \max_{\pi} V^\pi(s), \quad \pi^* = \arg \max_{\pi} V^\pi(s) \quad (3)$$

方策 π^* を最適方策、 $V^*(s)$ を最適状態価値関数という。

最適方策 π^* と最適状態価値関数 $V^*(s)$ は、Bellman方程式(2)を元にした価値反復[5]と呼ばれる反復近似解法で求めることができる。価値反復のアルゴリズムを以下に示す。なお、許容可能な状態の集合を \mathbf{S} 、最適状態価値関数の計算における変化量の最大値の許容誤差を θ とする。

Algorithm 1 : 価値反復のアルゴリズム

$$V(s) \leftarrow 0 \quad (s \in S) \quad (4a)$$

$$\text{while } \Delta > \theta \text{ do} \quad (4b)$$

$$\quad \Delta \leftarrow 0 \quad (4c)$$

$$\quad \text{for } s \in S \text{ do} \quad (4d)$$

$$\quad \quad v \leftarrow V(s) \quad (4e)$$

$$\quad \quad V(s) \leftarrow \max_a \sum_{s'} p_{ss'}^a (R_{ss'}^a + \gamma V(s')) \quad (4f)$$

$$\quad \quad \Delta \leftarrow \max \{ \Delta, |v - V(s)| \} \quad (4g)$$

$$\pi^*(s) \leftarrow \operatorname{argmax}_a \sum_{s'} p_{ss'}^a (R_{ss'}^a + \gamma V^*(s')) \quad (4h)$$

3. 動的計画法の適用例：グリッド空間でのマス目の価値

動的計画法の適用例として、あらかじめグリッド空間内のマス目に与えた報酬の条件下で得られる累積報酬を最大化するように、エージェントが移動すべき方向を DP により学習する。マス目の位置を状態とし、エージェントは上下左右の4方向のみに1マスずつ移動でき、グリッドの外へ出ようとする行動を選択した場合には直前のマス目に留まるものとする。ここで与える報酬は、意匠設計者がデザイン性を考慮して敷地内に計画する建築のヴォリュームから設定するものとして考えることができる。例えば、5×5グリッドの敷地に対してL字型のプランを提案する場合、図1のように報酬を与えれば良い。

1	1	0	0	0
1	1	0	0	0
1	1	0	0	0
1	1	1	1	1
1	1	1	1	1

図1: 報酬空間の例

図1の報酬設定から、各マス目の状態価値を価値反復により学習する。まず、各マス目に対応した状態価値関数 $V(s)$ を全て0に初期化する。続いて、状態価値関数を各状態 $s \in S$ について式(4f)で更新するが、その際、次式を仮定する[7]。

$$\sum_{s'} p_{ss'}^a R_{ss'}^a = r_s \quad (5)$$

r_s はマス目 s に設定した報酬の値である。行動 a や次状態 s' に依らず、移動前のマス目 s に対応した報酬 r_s を与える。したがって、式(4f)は次のように書き換えられる。

$$V(s) \leftarrow r_s + \max_a \sum_{s'} \gamma p_{ss'}^a V(s') \quad (6)$$

さらに、環境の条件として $p_{ss'}^a = 1.0$ を仮定する。すなわち、

「下へ移動する」という行動を選択した際には、次状態への遷移が他の方向へ逸れずに必ず下に向かう。以上より、式(6)は次のようにさらに簡略化できる。

$$V(s) \leftarrow r_s + \max_a \sum_{s'} \gamma V(s') \quad (7)$$

更新式(7)を(4f)式の代わりに用いて価値反復のアルゴリズムを実行し、その状態価値関数の変化が十分に小さくなるまで繰り返す。 $\gamma = 0.9, \theta = 0.001$ に設定した場合、(4b)から(4g)の手順を10回繰り返すことで収束し、図2のように最適状態価値関数の近似解をプロットできる。

1.00	1.00	0.90	0.81	0.73
1.00	1.00	0.90	0.81	0.81
1.00	1.00	0.90	0.90	0.90
1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.00	1.00	1.00

図2: 図1の報酬を元に学習した最適状態価値 (最大値を1.0にスケール)

以上のようにして学習した状態価値を元に、行動を次式で確率的に決定する。

$$\pi(s, a) = (V(s'))^k \quad (8)$$

$k (> 0)$ は次状態の価値に応じた行動の偏重度合いを調整するパラメータである。例えば、 $k = 2.0$ に設定すると、(8)の方策に基づいて図2において中央のマスから左または下に移動する確率は

$$\frac{1.0^2}{1.0^2 + 1.0^2 + 0.9^2 + 0.9^2} = 0.276 \quad (9)$$

4. 最適化問題

以下では、境界条件、荷重条件、部材断面と節点の接続関係を固定したラーメン構造について、 n 個の自由節点の平面的な位置 $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n\}$, $\mathbf{X}_i = (x_i, y_i)$ を z 座標を固定させながら移動し、単一荷重に対するコンプライアンスの最小化と、節点が位置するマス目の報酬の合計値の最大化を目的とする多目的最適化問題を定式化する。

節点変位の自由度を n_d , 外力ベクトルを $\mathbf{p} \in \mathbb{R}^{n_d}$ として、変位ベクトル $\mathbf{u}(t) \in \mathbb{R}^{n_d}$ は次の剛性方程式を解くことで計算できる。

$$\mathbf{K}(t)\mathbf{u}(t) = \mathbf{p} \quad (10)$$

ここで、 $\mathbf{K}(t) \in \mathbb{R}^{n_d \times n_d}$ は全体剛性行列である。節点が重複せず、 $\mathbf{K}(t)$ が正則なとき、コンプライアンス F_1 は次式で表せる。

10 個の最適解のうち、最も目的関数の小さい解の形状を図 5 右に示す。青い部材ほど圧縮応力の大きい部材である。初期形状は自重に対して最も効率よく抵抗できる形状であるため、コンプライアンスが最も小さくなる ($F_1 = 0.184$) が、どの自由節点も報酬のあるマス目内に位置していない ($F_2 = 0.0$)。最適解ではコンプライアンスは $F_1 = 0.431$ と大きくなったが、20 個中 18 個の自由節点が 1 の報酬を獲得しており、結果として目的関数値が初期形状より小さくなっている。つまり、デザイン意図に従って設定した報酬に応じた構造形態が得られたことがわかる。

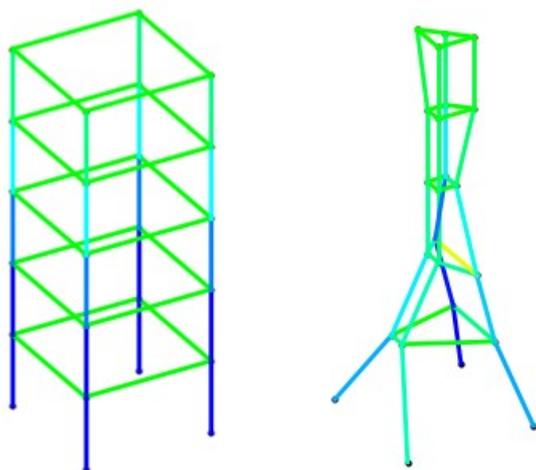


図 5: 初期形状 (左, $F = 0.184$) と最適形状 (右, $F = -1.431$)

提案手法の有効性を確認するため、節点移動の方策を学習せずに、節点を上下左右に一樣な確率で移動させて近傍解を生成した場合との最適解の比較を行った。それぞれの場合について、ランダムシードを変更して得られた 10 個の最適解の目的関数値を表 1 に示す。学習を行った方が目的関数値の最大値・中央値・最小値・平均ともに学習を行わなかった場合に比べて小さくなった。したがって、提案手法に基づいて節点移動の方策を学習した方がより良い最適解が得られる。

表 1: 方策学習の有無別の最適解 10 個の目的関数 F

	max	median	min	mean	std.dev.
学習無	-0.291	-0.765	-1.310	-0.812	0.345
学習有	-0.927	-1.214	-1.431	-1.117	0.179

さらに、方策の学習を行った方が良い最適化結果が得られる理由について検証するために、最も小さい目的関数値をとったケースの目的関数値の履歴を学習の有無別に図 6 に示す。学習有りの場合、序盤から終盤まで目的関数値を小さく抑えられている。このことは、節点移動の方策を学

習することで、解の探索効率が改善できることを示している。

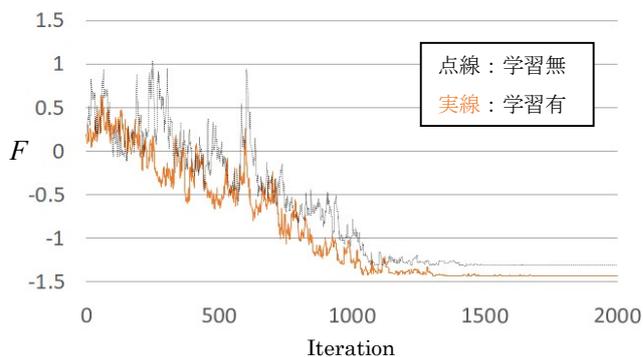


図 6: 方策学習の有無別の最優良解の最適化履歴

6. 結

本報では、コンプライアンス最小化とデザイン指標に基づく節点位置の報酬最大化を目的とする、ラーメン構造物の形状最適化問題において、焼きなまし法の近傍生成段階で動的計画法による学習結果を用いることで効率的に解空間を探索する新たな最適化手法を提案した。

動的計画法は強化学習の理論的基礎をなす手法である。今後はさらに発展的な強化学習手法を用いて、より複雑な建築構造最適化問題への応用を試みる。

謝辞

本研究の一部は JSPS 特別研究員奨励費 18J21456、科研費 16H03014、18K18898 により助成を受けた。ここに記して謝意を表する。

【参考文献】

- 1) 中川佳久, 國光修五: 不整形な建物の耐震設計における構造計算ソフトウェアの対応, 2017 年度日本建築学会大会 (中国) 構造部門 (応用力学) パネルディスカッション資料, 2017 年 9 月
- 2) Ohsaki, M. Optimization of Finite Dimensional Structures. CRC Press, 2010.
- 3) Papadrakakis M., Tsompanakis J., Lagaros N. Structural shape optimization using evolution strategies. Eng. Optim., Vol. 31, pp. 515-540, 1999.
- 4) Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P. Optimization by simulated annealing. Science, Vol. 220, pp. 671-680, 1983.
- 5) Bellman, R. Dynamic Programming. Princeton University Press, 1957.
- 6) Sutton, R. S., Barto, A. G. Reinforcement Learning: An Introduction. MIT Press, 1998.
- 7) Russell S. J., Norvig P. Artificial intelligence: a modern approach. Pearson Education, 2003.

*1 京都大学大学院工学研究科 博士後期課程 修士(工学)

*2 京都大学大学院工学研究科 教授 博士(工学)