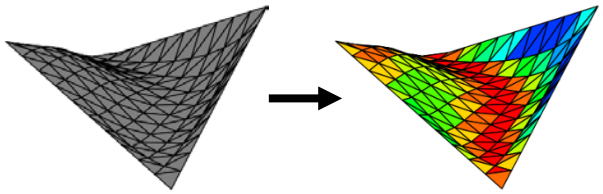
強化学習を導入した平面骨組構造物の最適設計

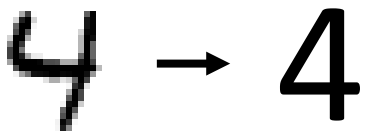
林 和希（京都大学）
大崎 純（京都大学）

機械学習 = 入力と出力の関係を学習

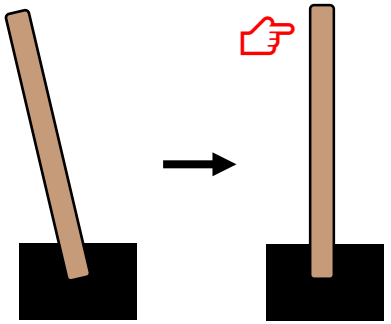
Type	教材	
1. 教師なし学習	なし	
2. 教師あり学習	正しい出力値	
3. 強化学習	出力の評価関数	← 研究対象



パネルのクラスタリング



手書き文字の識別

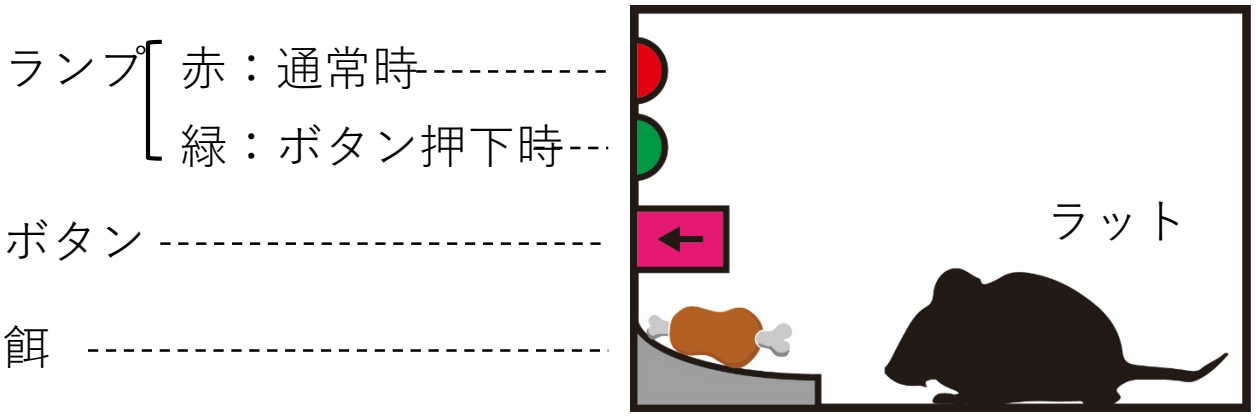


Poll balancing問題

強化学習

- Skinnerの提唱したオペランド学習に由来
- ”スキナー箱(Skinner box)”
  ・・・ボタンを押すとラットに餌が与えられる

⇒ラットはボタンを押す動作を繰り返す (強化)

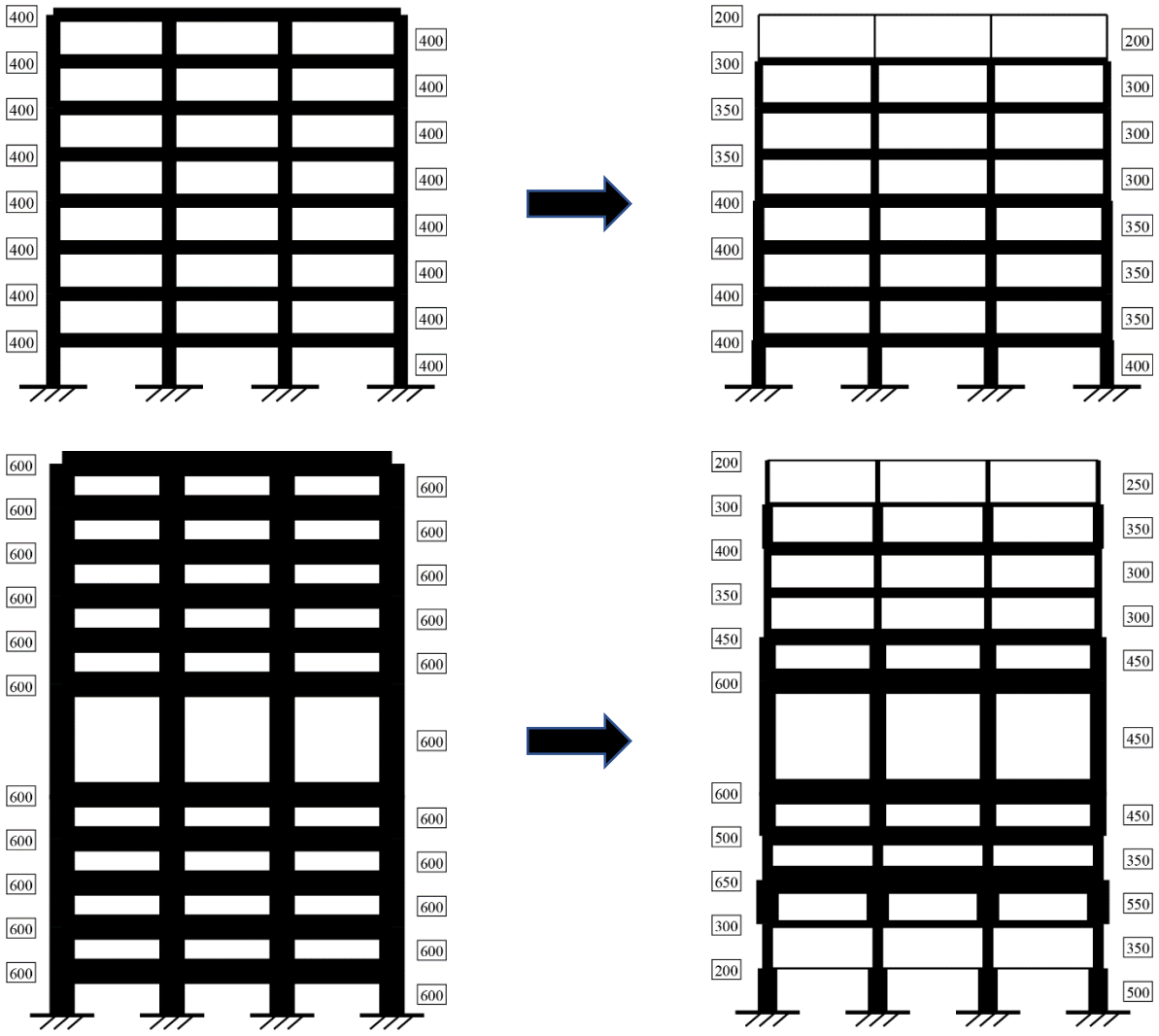


Skinner, B. F. (1938). *The Behavior of organisms: An experimental analysis*. New York: Appleton-Century.

なぜ構造最適化×強化学習？

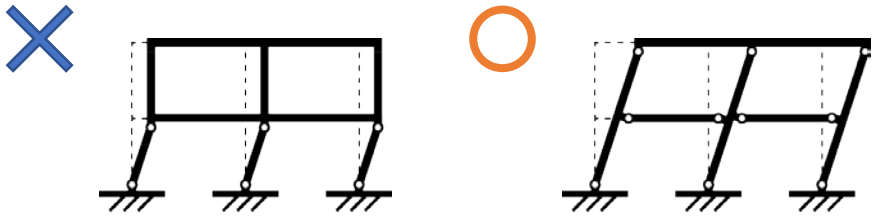
1. 非線形かつ困難な問題への適用
2. 予期せぬ優良解の生成
3. 計算負荷の低減

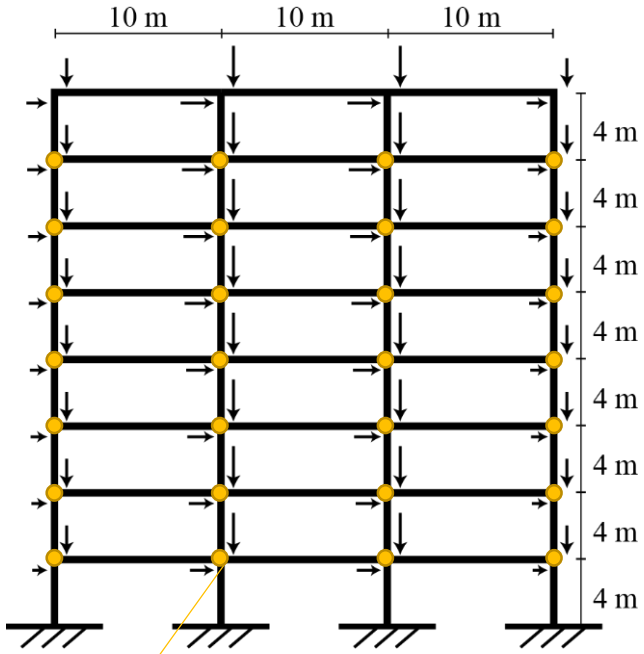
例題1：骨組の断面最適化



骨組断面の最適化

1. 部材総体積の最小化
2. 応力制約
3. 崩壊メカニズムの制約



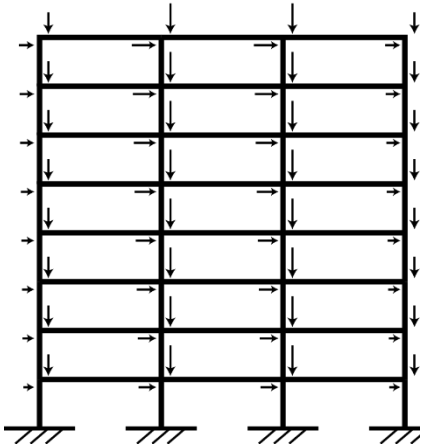


$$\text{minimize } V(\mathbf{A})$$ 1. total structural volume

$$\text{subject to } \sigma_i(\mathbf{A}) \le \bar{\sigma}_i \quad (i = \text{all members})$$ 2. stress constraint

$$\beta_j(\mathbf{A}) \ge 1.5 \quad (j \in \text{middle layer nodes})$$ 3. column-to-beam strength ratio

断面が離散変数の場合



$$\text{minimize } V(\mathbf{A})$$

$$\text{subject to } \sigma_i(\mathbf{A}) \le \bar{\sigma}_i \quad (i = \text{all members})$$

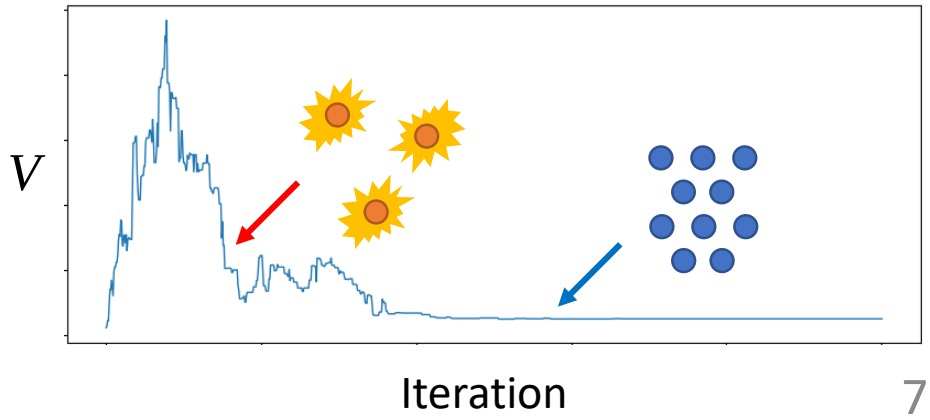
$$\beta_j(\mathbf{A}) \ge 1.5 \quad (j \in \text{middle layer nodes})$$

初期断面

index	Column □	Beam H
200	200x200	194x150
250	250x250	244x175
300	300x300	294x200
350	350x350	340x250
400	400x400	400x200
450	450x450	450x200
500	500x500	500x250
550	550x550	550x250
600	600x600	600x250
650	650x650	650x250

焼きなまし法 (SA):

粒子の冷却過程を模擬したメタヒューリスティクス

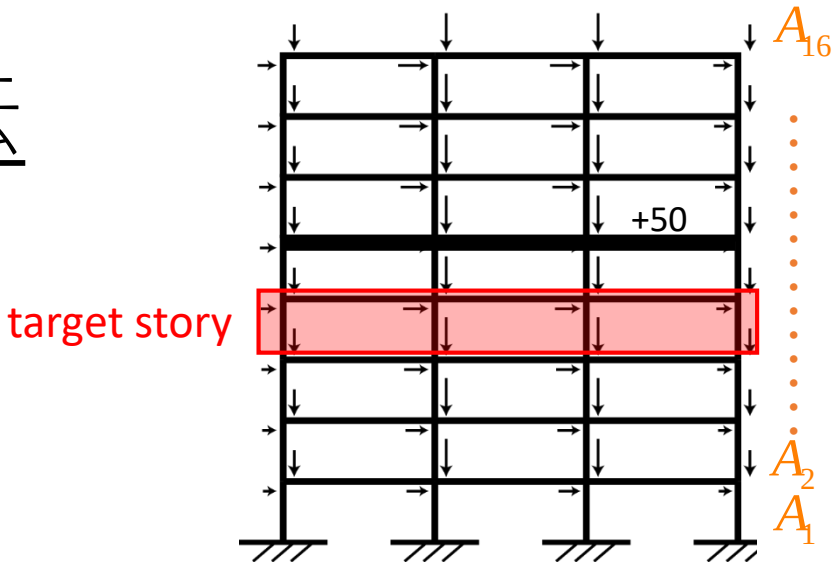


変数の更新方法

$$\text{minimize } V(\mathbf{A})$$

$$\text{subject to } \sigma_i(\mathbf{A}) \le \bar{\sigma}_i$$

$$\beta_j(\mathbf{A}) \ge 1.5$$



初期断面

index	Column □	Beam H
200	200x200	194x150
250	250x250	244x175
300	300x300	294x200
350	350x350	340x250
400	400x400	400x200
450	450x450	450x200
500	500x500	500x250
550	550x550	550x250
600	600x600	600x250
650	650x650	650x250

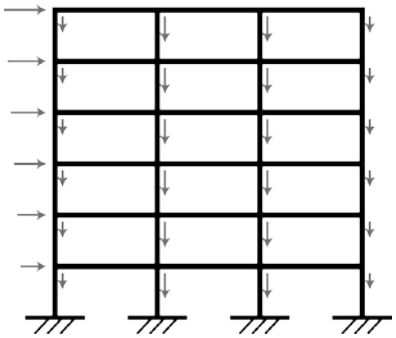
変数の更新方法:

1. target storyを一層上に移動
2. target storyの柱又は梁断面を無作為に変更 (-50, 0, +50)
3. 目的・制約関数を計算
4. 変更後の解を受容又は棄却

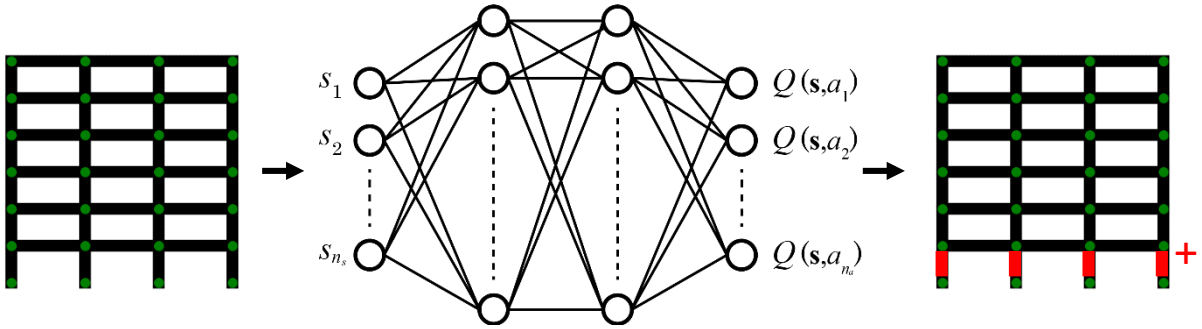
“賢く”変数を更新できないか

提案手法の手順

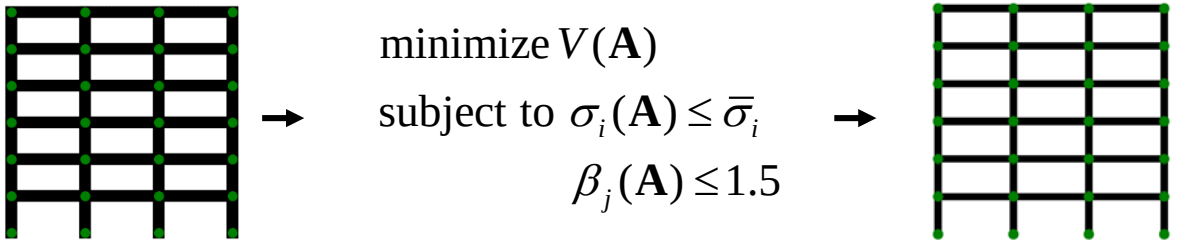
1. 骨組モデルを仮定



2. 強化学習
   (New!)

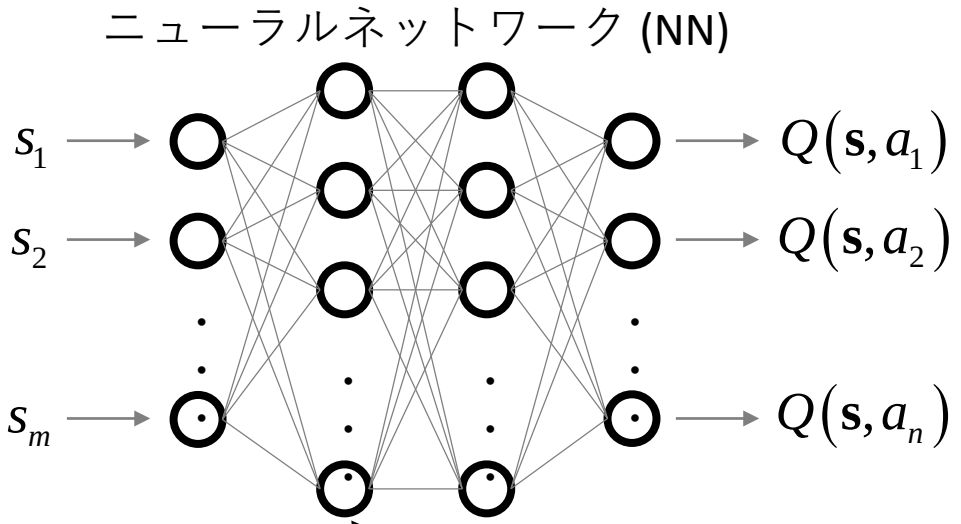


3. 最適化

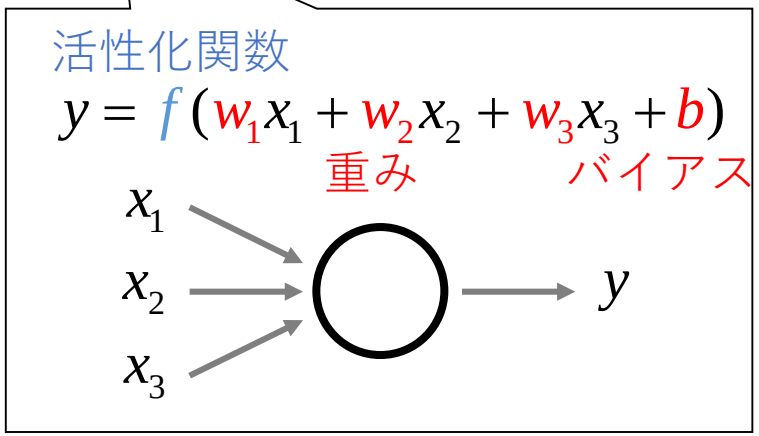


Deep Q-Network(DQN)

- input：状態値
- output：行動価値

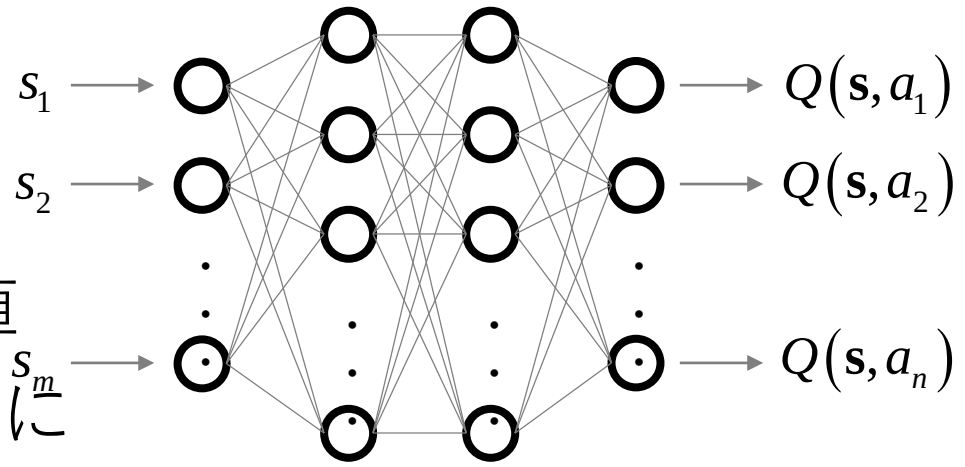


- 各ニューロンが入力値$\mathbf{x}$を$(\mathbf{w},\mathbf{b})$とfを用いてyに変換
- 特徴量抽出に人間の介在を殆ど必要としない (≒ AI)



Deep Q-Network(DQN)

- NNは実際の報酬を観測するにつれてより正確な行動価値Qを推定できるように



Q-learning (Watkins, 1989)

$$Q(\mathbf{s},a)=Q(\mathbf{s},a)+\alpha\left(r(\mathbf{s}')+\gamma\max_a Q(\mathbf{s}',a)-Q(\mathbf{s},a)\right)$$

観測された報酬 + 次状態以降の累積報酬の推定値 ［現状態以降の累積報酬の推定値］

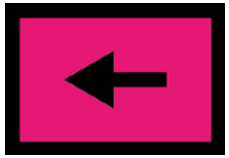
- Deep Q-Network

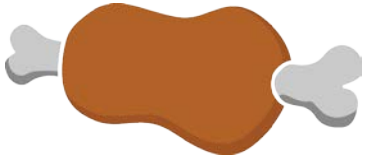
$$\text{minimize } F(\mathbf{w},\mathbf{b})=\left(r(\mathbf{s}')+\gamma\max_a Q(\mathbf{s}',a)-Q(\mathbf{s},a)\right)^2$$

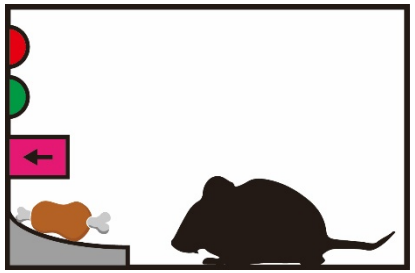
強化学習に必要なもの

エージェント

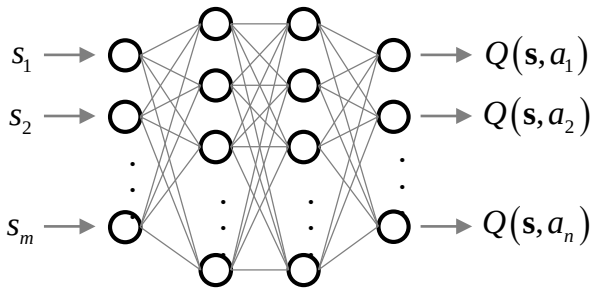
1. 状態s　2. 行動a　3. 報酬r　4. 環境F(s,a) = {s’,r}



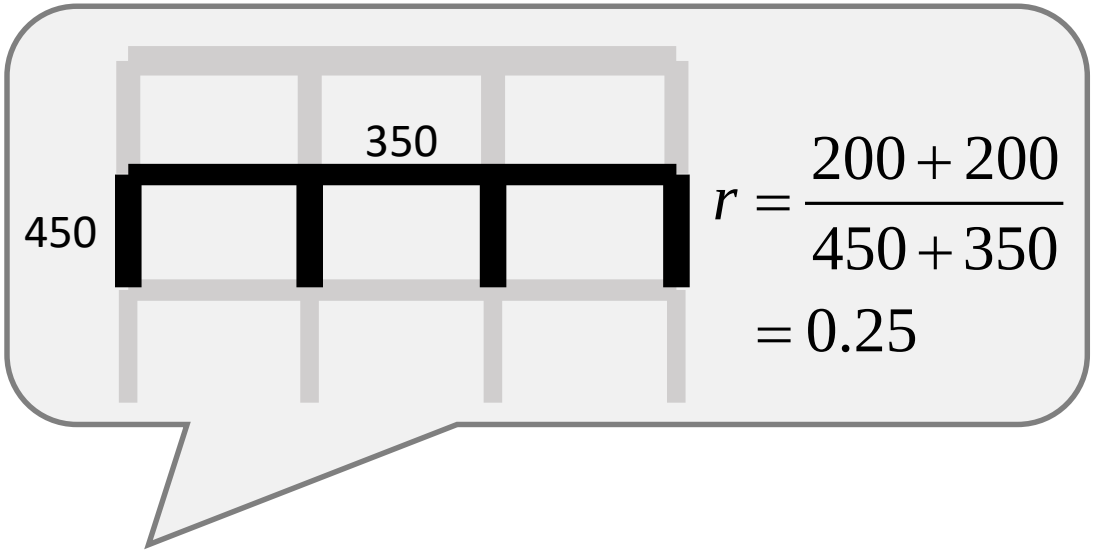




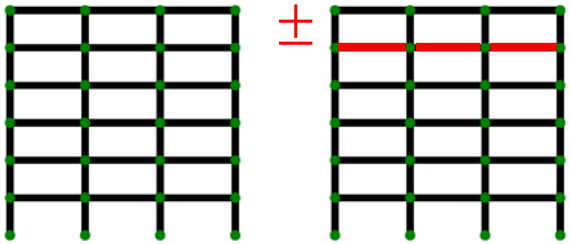
強化学習に必要なもの



エージェント



1. 状態s　2. 行動a　3. 報酬r　4. 環境F(s,a) = {s’,r}



s

a

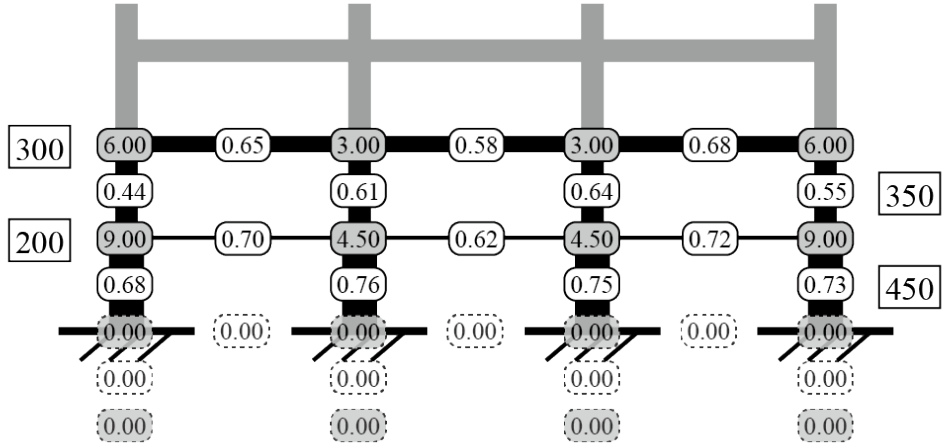
$$r = \frac{\sum \text{min. indices}}{\sum \text{sect. indices}}$$

Python coding

$$F(s,a) = \{s', r\}$$

※ $r = 0$ if not satisfy constraints

状態: 22 入力



$$s_a = \begin{pmatrix} 0 & 450/650 & 350/650 \\ 0 & 200/650 & 300/650 \end{pmatrix} \quad s_b = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \quad s_c = \begin{pmatrix} 0 & 0.76 & 0.64 \\ 0 & 0.72 & 0.68 \end{pmatrix} \begin{matrix} \\ \text{— column} \\ \text{— beam} \end{matrix}$$

$$s_d = \begin{pmatrix} 0 & 0 & 4.50 & 3.00 \end{pmatrix} \quad s_e = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

6 s_a: target storyとその上下の層の柱梁断面のインデックス

4 s_b: 断面が上下限なら1となる{0,1}値

6 s_c: 同一層の柱または梁ごとの応力の検定比の最大値

4 s_d: 同じ高さにある節点ごとの、柱梁耐力比の最小値

2 s_e: 現在のtarget storyが最下階か最上階なら1となる{0,1}値

行動: 5 出力

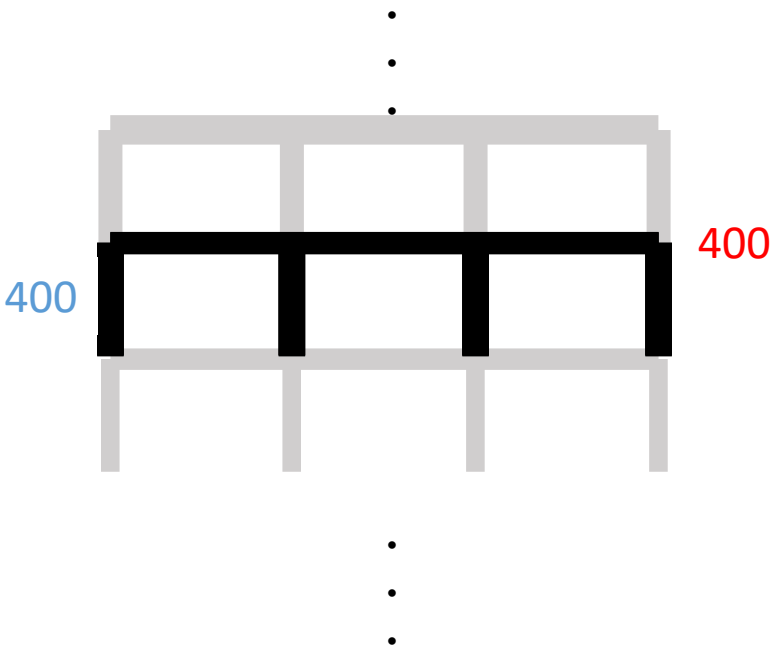
a_1: 柱断面を1段階大きくする

a_2: 梁断面を1段階大きくする

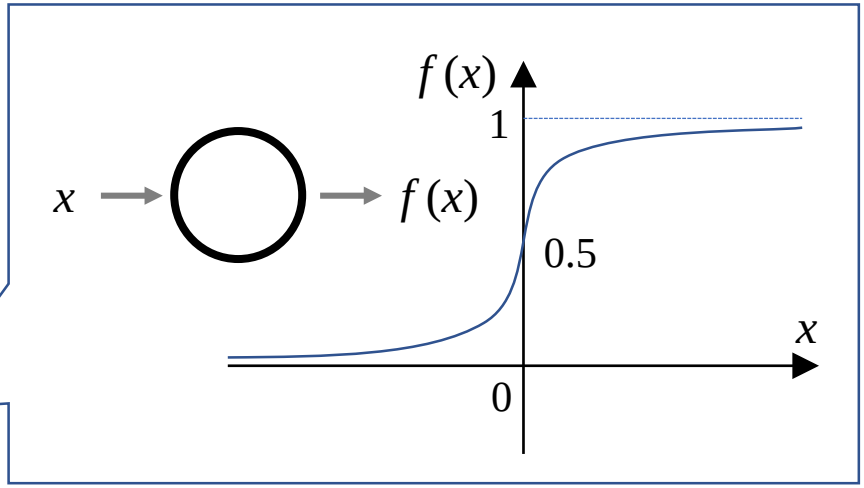
a_3: 柱断面を1段階小さくする

a_4: 梁断面を1段階小さくする

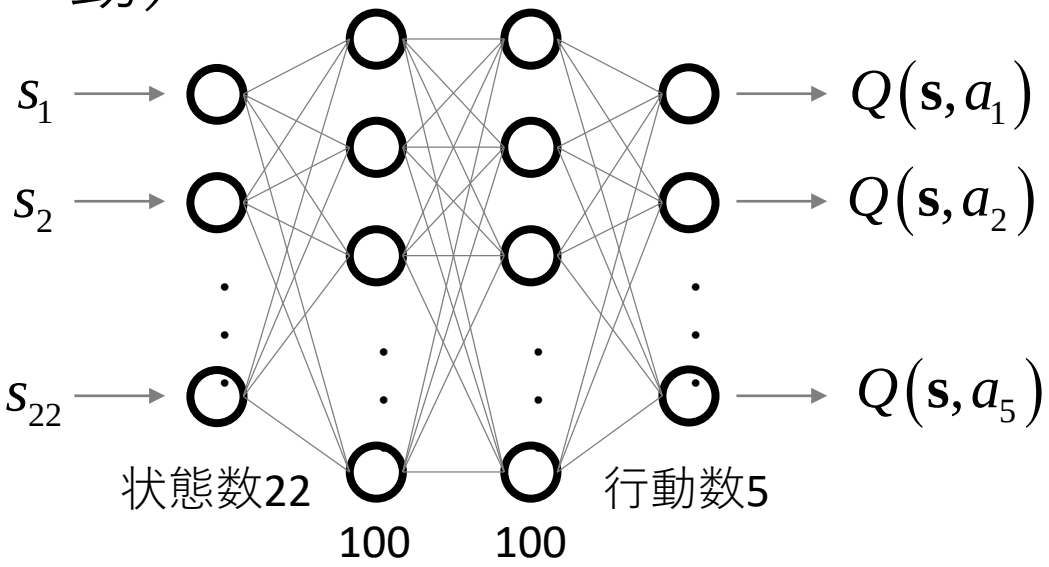
a_5: 現在の断面をそのまま維持



NNの設定



- 最適化手法：Adam
- 活性化関数：Sigmoid
- 方策：ε-greedy方策（ε=0.2の確率でランダム行動）

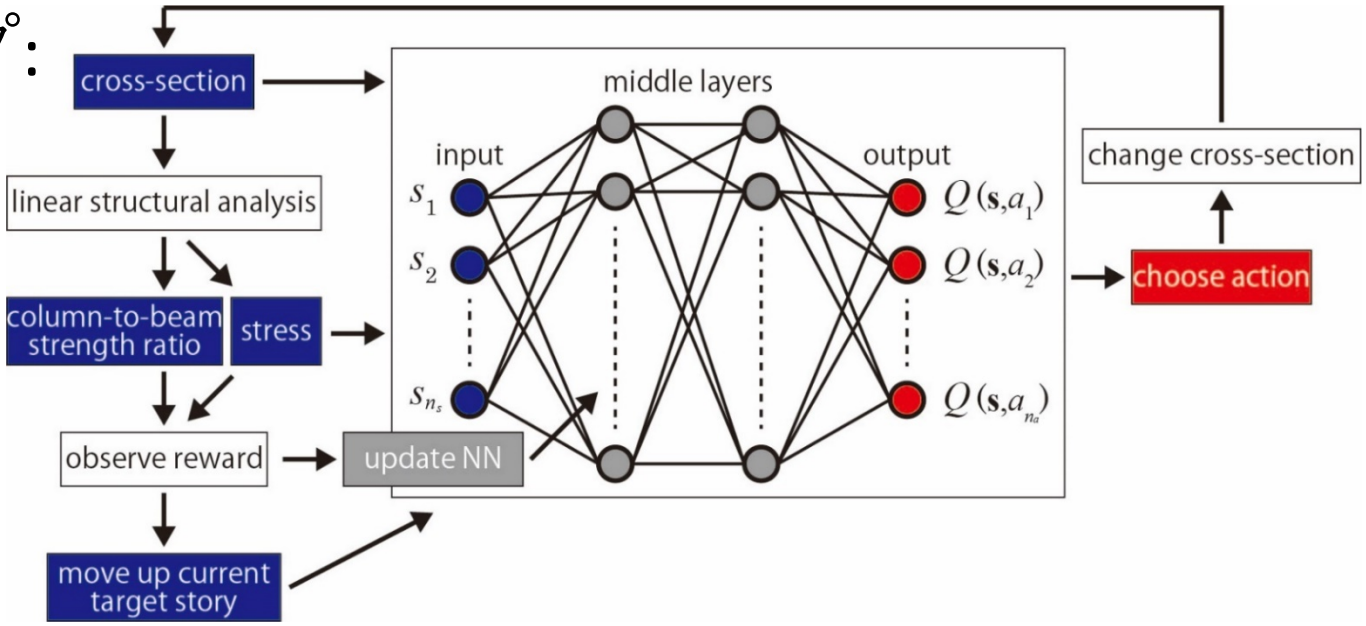


元の行動価値：10^{16}次元

(22+1)×100
+(100+1)×100
+(100+1)×5
=12905次元に縮約

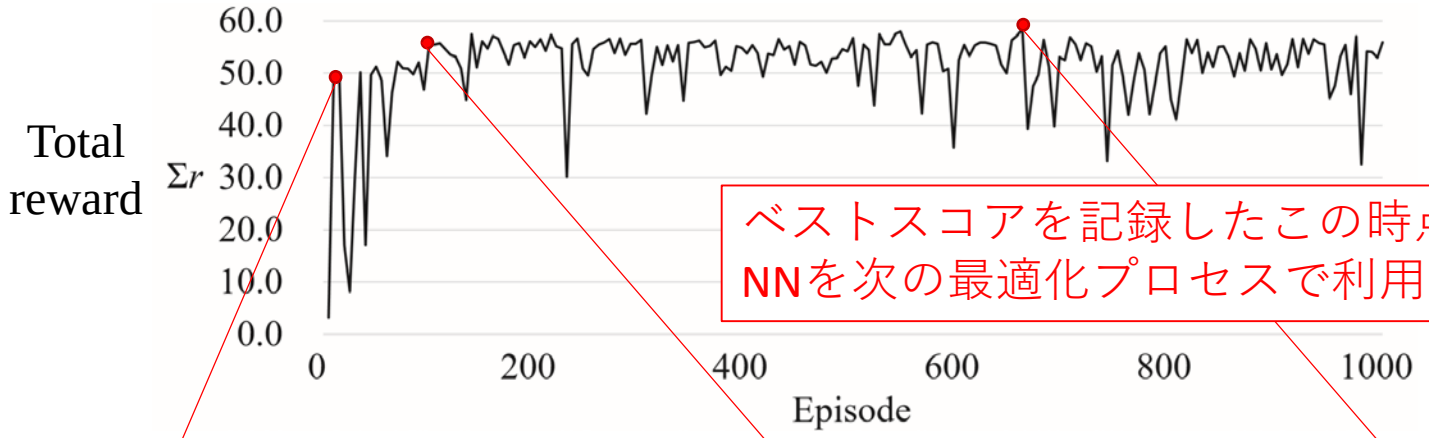
学習ワークフロー

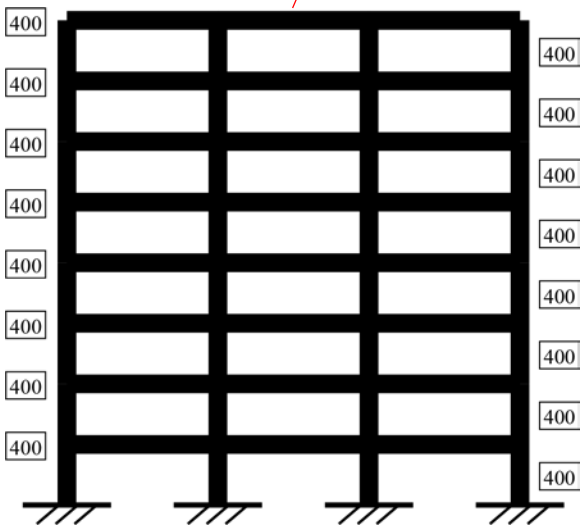
- 1ステップ:



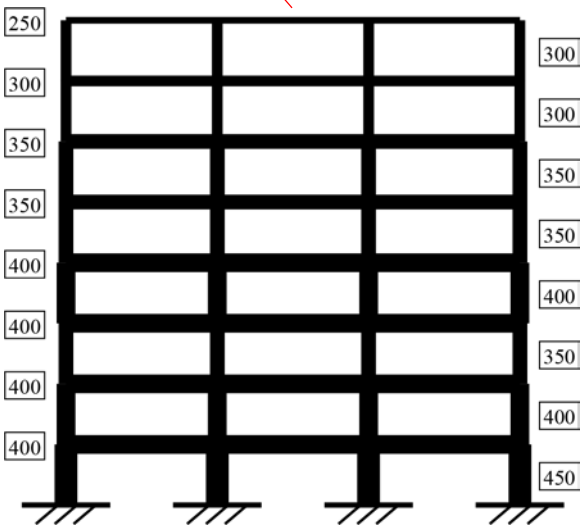
- 100ステップ繰り返す (= 1エピソード)
- 1000エピソード繰り返し学習させる (≒ 30分)

学習過程での累積報酬の履歴

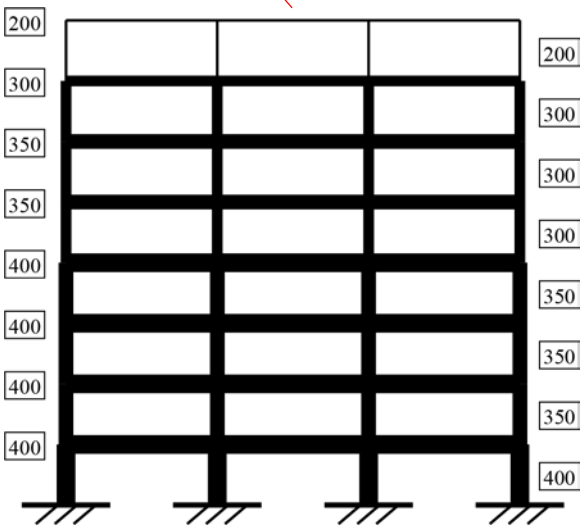




$V = 6.579\ [\mathrm{m}^3]$



$V = 5.546\ [\mathrm{m}^3]$



$V = 4.852\ [\mathrm{m}^3]$

提案するSA+DQNアルゴリズム

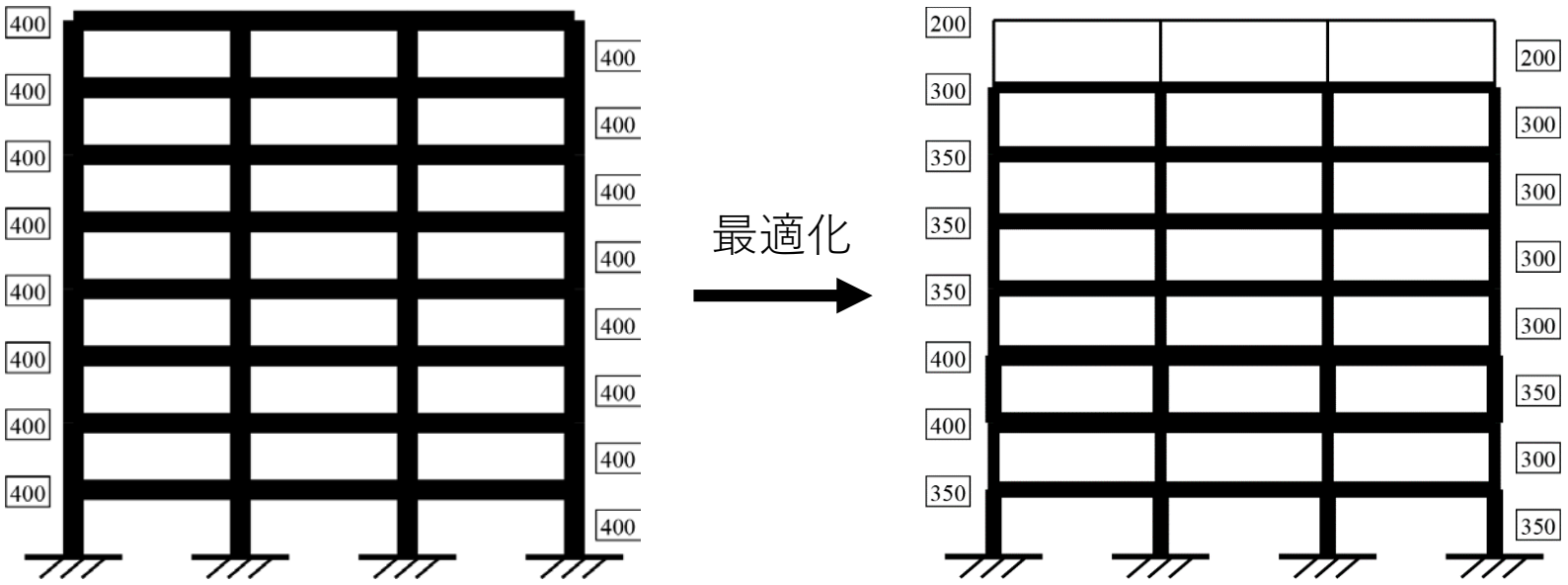
input: x: initial solution, T: initial temperature, c: cooling rate, n_c: cooling interval, n_i: maximum number of iteration
output: x_b: the best solution

1 $x_b \longleftarrow x$
2 $j \longleftarrow 0$
3 **for** $i \leftarrow 1$ **to** n_i **do**
4 $\quad y \longleftarrow$ neighborhood solution of x generated by ~~randomly~~ RL agent
5 $\quad$ **if** $f(y) < f(x_b)$ **then**
6 $\quad\quad x_b \longleftarrow y$
7 $\quad$ **else if** $f(y) < f(x)$ **then**
8 $\quad\quad x \longleftarrow y$
9 $\quad$ **else**
10 $\quad\quad$ **if** $\exp\left(\frac{f(x)-f(y)}{T}\right) \geq$ random $[0, 1]$ **then** $x \longleftarrow y$
11 $\quad j \longleftarrow j + 1$
12 $\quad$ **if** $j = n_c$ **then**
13 $\quad\quad T \longleftarrow cT$
14 $\quad\quad j \longleftarrow 0$
15 **return** x_b

最適化結果

SAにDQNを援用した場合としなかった場合で10回ずつ最適化 ⇒ SA+DQNの方が良い結果を得た

	Max.	Median	Min.	Average	Std. dev.
SA+DQN	4.935	4.701	4.468	4.676	0.155
SA only	5.334	4.860	4.657	4.920	0.207

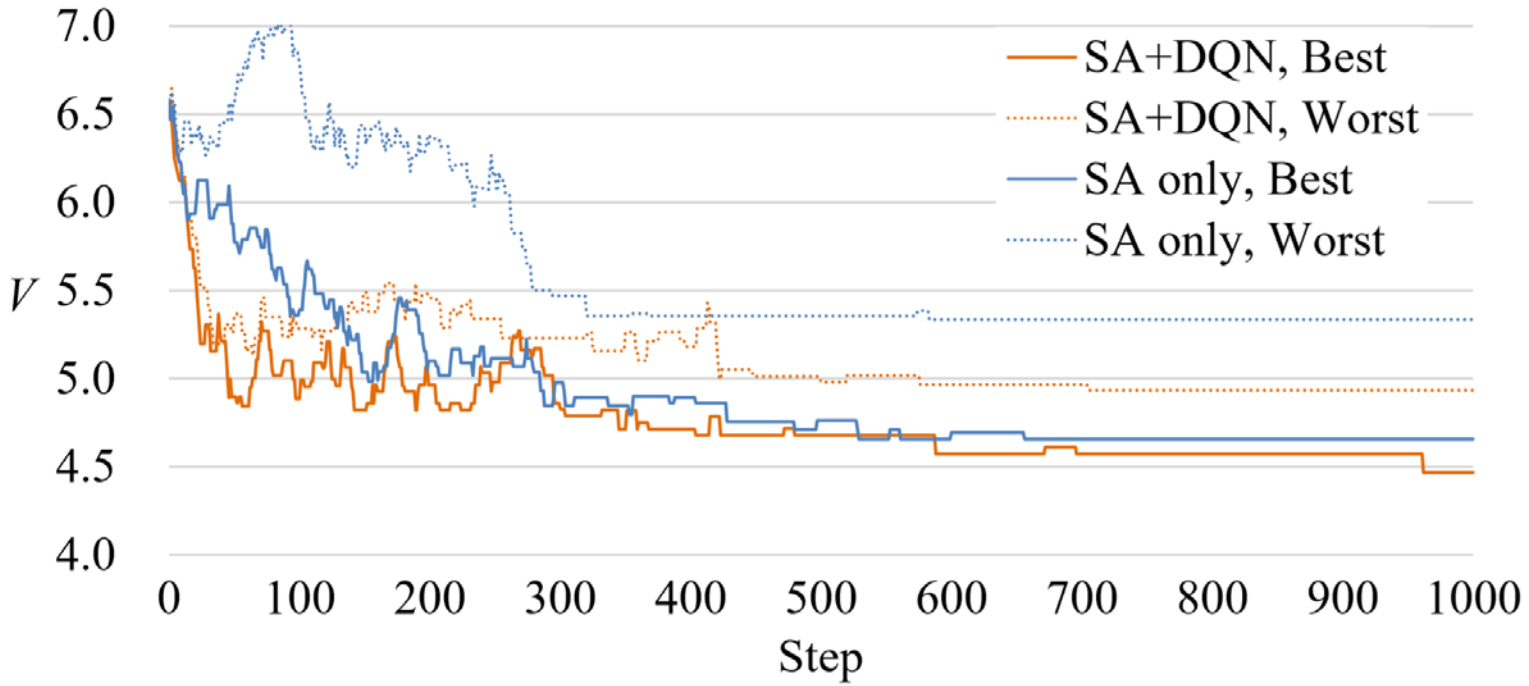


$V = 6.579\ [\mathrm{m}^3]$

$V = 4.468\ [\mathrm{m}^3]$

最適化結果

SA+DQNでは最適化の初期ステップにおいて急速に目的関数が減少 ⇒ “賢く”最適化している



10個ずつ得られた解で最良・最悪の解についての最適化途中の目的関数の履歴

SAパラメータに対する頑強性

SAのみと比較して、SA+DQNではSAの温度パラメータに依存せず安定した最適化結果を得た

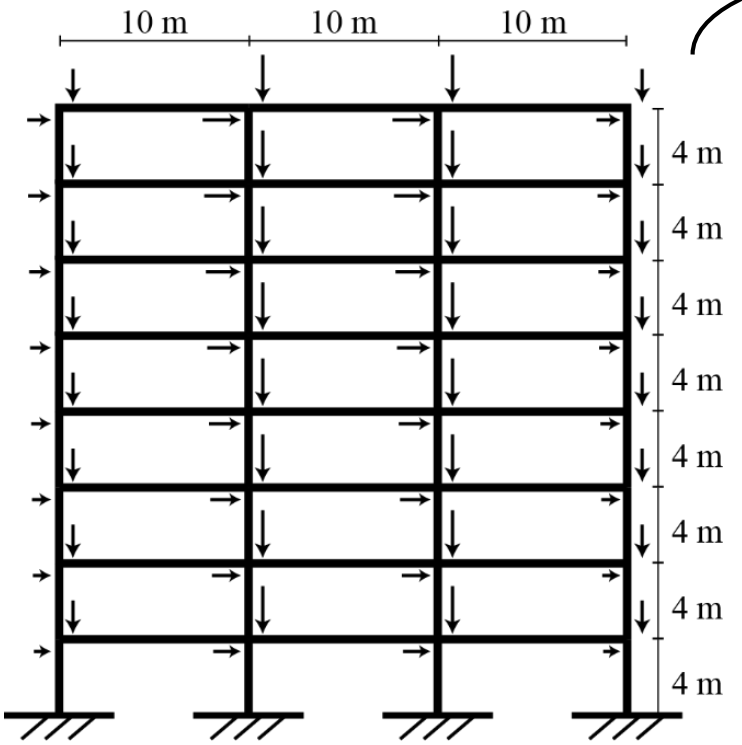
10回ずつ最適化して得られた目的関数(部材総体積)Vの中央値

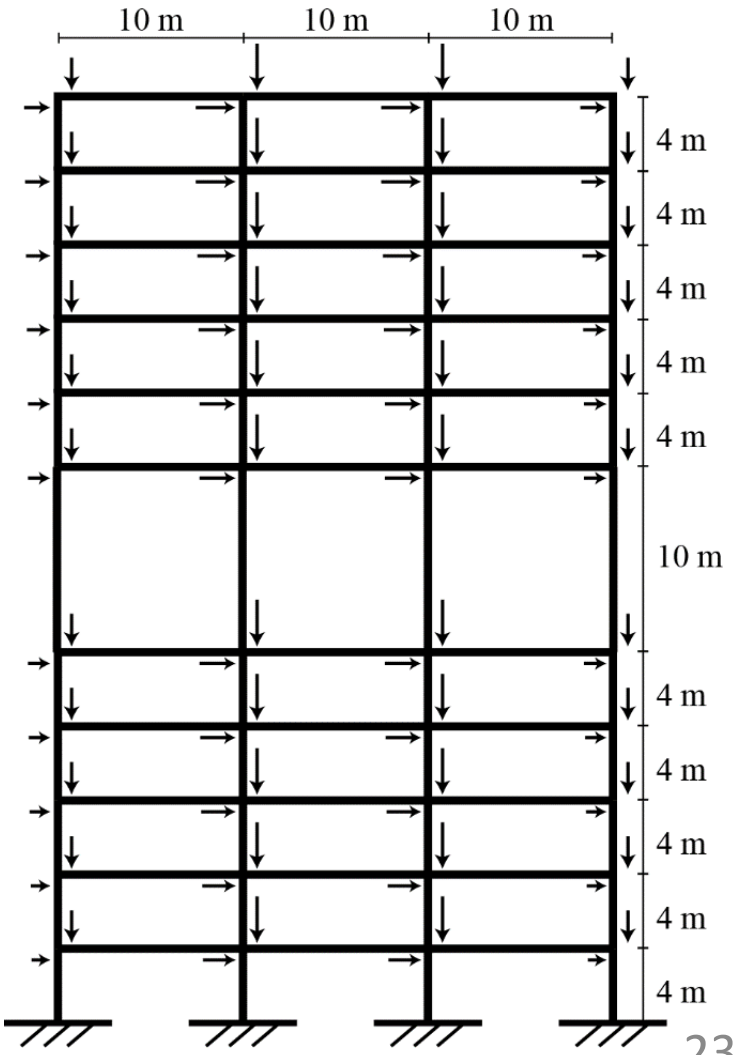
	SA+DQN			**SA only**		
	$c = 0.8$	$c = 0.9$	$c = 0.95$	$c = 0.8$	$c = 0.9$	$c = 0.95$
$p_0 = 0.2$	4.725	4.672	4.782	4.861	4.794	4.808
$p_0 = 0.8$	4.677	4.701	4.697	5.023	4.860	4.804
$p_0 = 0.99$	4.782	4.762	4.815	5.141	5.695	6.453

p_0 : 改悪解を受理する初期確率, c : 冷却率

異なる骨組モデルにも適用

学習済みのNNを再学習無でそのまま最適化に使用



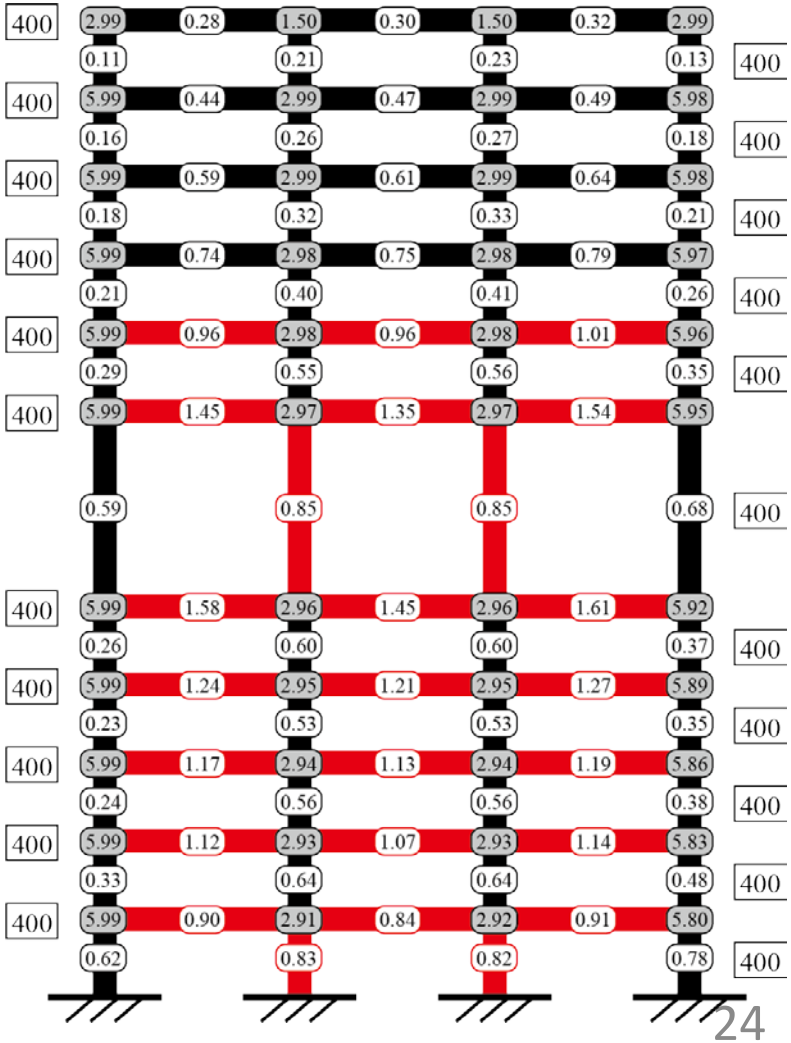


初期断面が制約を超過する場合

“400” の断面を初期解に割り当てると一部の部材で応力制約を超過

↓

ペナルティ項などを導入しないと制約を満たす解を得ることが困難になる



最適化結果

- SA+DQNは制約を満たす解に到達できる

10個の最適化結果の部材総体積
(制約を満たさない“400”の初期断面を与えた場合)

	Max.	Median	Min.	Average	Std. dev.
SA+DQN	11.000	10.482	10.175	10.505	0.293
SA only	No feasible solution found				

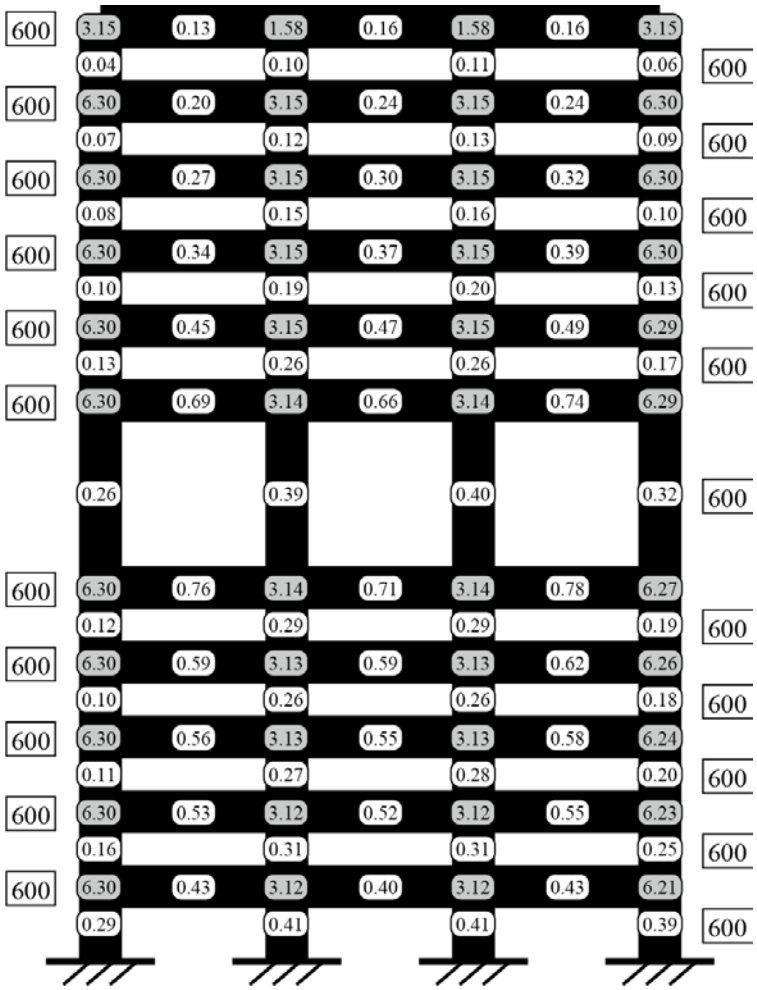
10個中9個の解が制約を満たす解

- 制約を満たす初期解から最適化しても、SA+DQNの方がわずかに良い結果が得られた

10個の最適化結果の部材総体積
(制約を満たす“600”の初期断面を与えた場合)

	Max.	Median	Min.	Average	Std. dev.
SA+DQN	10.975	10.333	9.775	10.349	0.303
SA only	10.850	10.504	10.084	10.450	0.206

最優良解

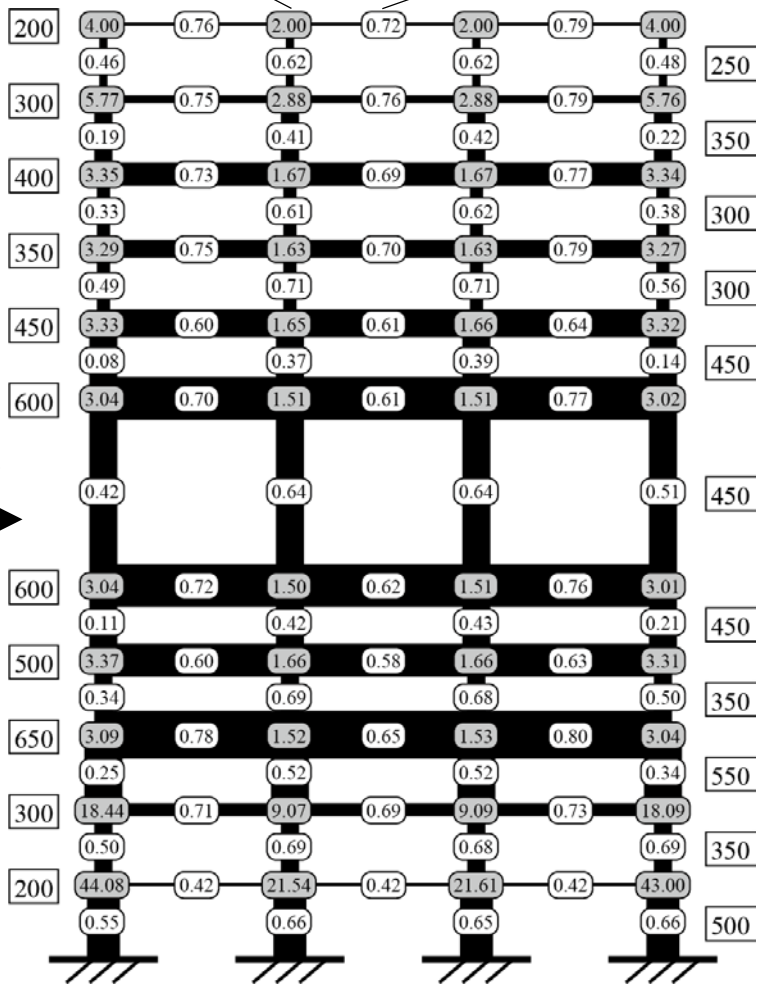


$V = 17.208\ [\mathrm{m}^3]$

最適化

1.5以上ならOK　　0.8以下ならOK

柱梁耐力比　　応力検定比



$V = 9.775\ [\mathrm{m}^3]$

結論

- 構造要件を満たしながら部材総体積を減らすための骨組断面の変更方策を強化学習で学習
- 学習済エージェントとメタヒューリスティクスを併用し、より良い最適化結果を効率的に得た
- 学習済エージェントを再学習なしで異なる骨組モデルにも適用できることを確認

K. Hayashi and M. Ohsaki, Reinforcement learning for optimum design of a plane frame under static loads, Engineering with Computers, published online. DOI: 10.1007/s00366-019-00926-7