Machine-specified Ground Structures for Topology Optimization of Binary Trusses

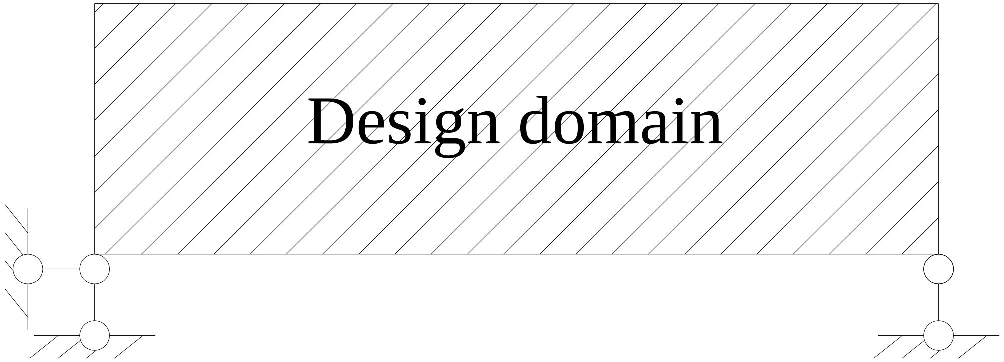
Shaojun Zhu, Makoto Ohsaki, Kazuki Hayashi, Xiaonong Guo

Ohsaki–Zhang Lab, Kyoto University

August 18, 2021

Background

Use AI to generate GSs?



Generate n nodes to make the design domain finite

***Number of members*:**

$n(n–1)/2$

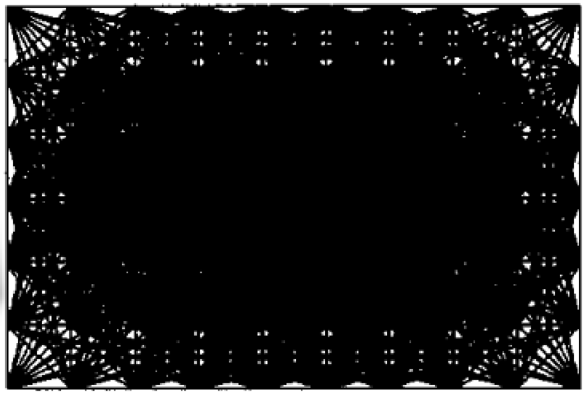
***Possible combinations*:**

$C^{n(n–1)/2}$

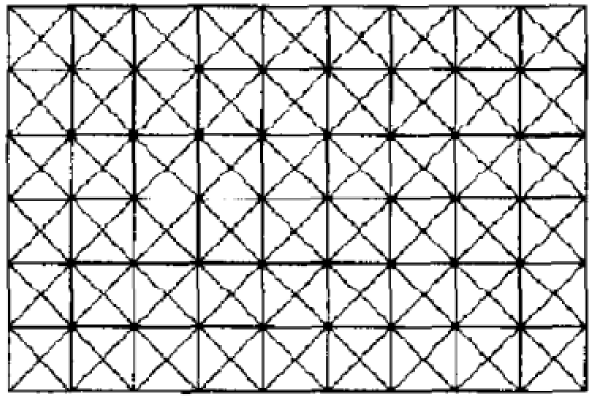
C: Number of sections

↓

High computational cost



Fully-connected ground structure (GS)



Human-specified GS

***Number of members*:**

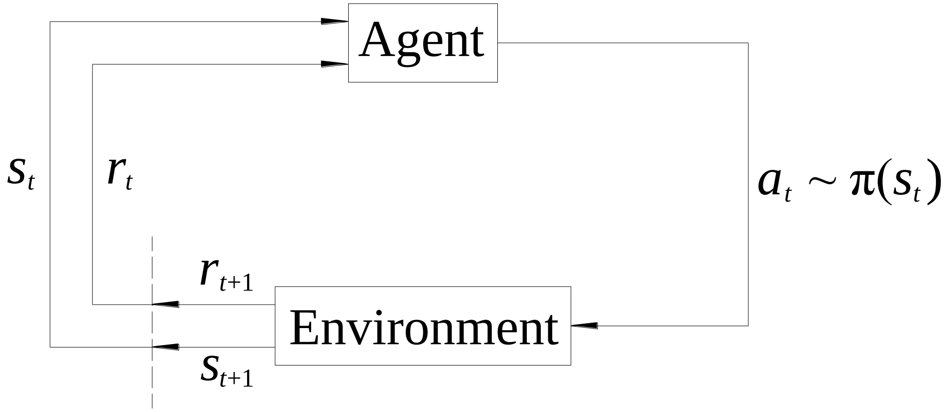
$2n - n_c < m < n(n–1)/2$

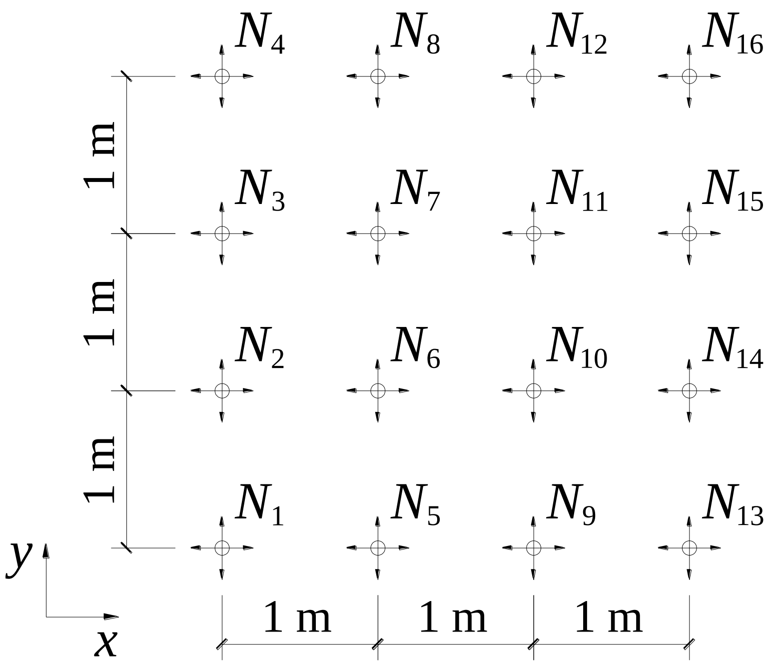
Kinematically stable

↓

Laborious to arrange

Reinforcement Learning (RL)





Task:

Generate a stable GS with fewest members

State:

Information on topology and nodal location

Action:

Select a node at a time
(2 selected nodes form a member)

Reward:

Positive for good actions
Negative for bad actions

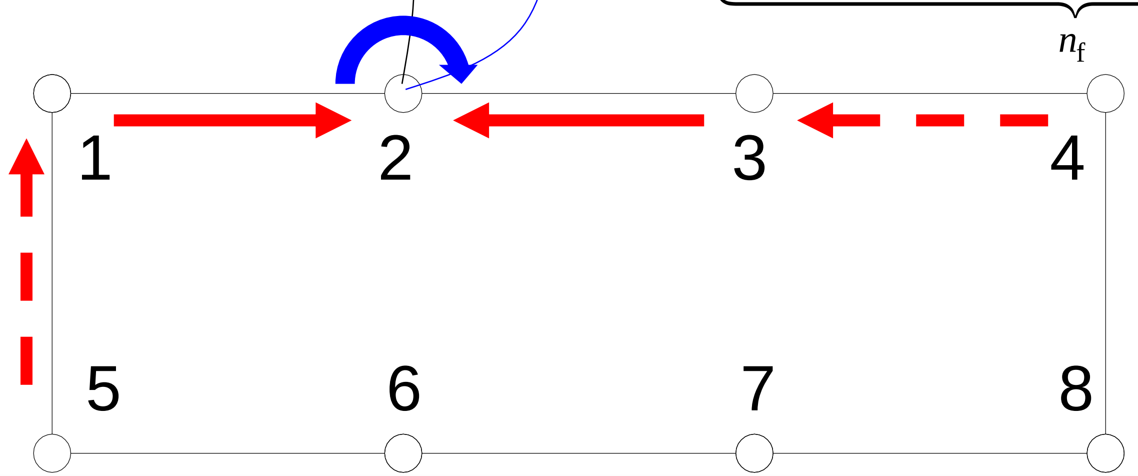
State description by Graph Embedding (GE)

nodal information vector

$$\mathbf{v}_i = \{x_i \quad y_i \quad \mathrm{Sel}_i\}$$

comprehensive information vector

$$\boldsymbol{\mu}_i = \underbrace{\{\mu_{i,1} \quad \mu_{i,2} \quad \mu_{i,3} \quad \cdots \quad \mu_{i,n_\mathrm{f}}\}}_{n_\mathrm{f}}$$



$$\hat{\mathbf{C}}_{8\times 8} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Neural network parameters (independent of *n*)

$$\boldsymbol{\mu}_i^{(T+1)} \leftarrow \mathrm{ReLU}\left(\boldsymbol{\theta}_1 \mathbf{v}_i + \boldsymbol{\theta}_3 \mathrm{ReLU}\left[\boldsymbol{\theta}_2\left(\hat{\mathbf{v}}\mathbf{C}_i^{(N)}\right)\right] + \boldsymbol{\theta}_4 \boldsymbol{\mu}_i^{(T)} + \boldsymbol{\theta}_6 \mathrm{ReLU}\left[\boldsymbol{\theta}_5\left(\hat{\boldsymbol{\mu}}^{(T)}\mathbf{C}_i^{(N)}\right)\right]\right)$$

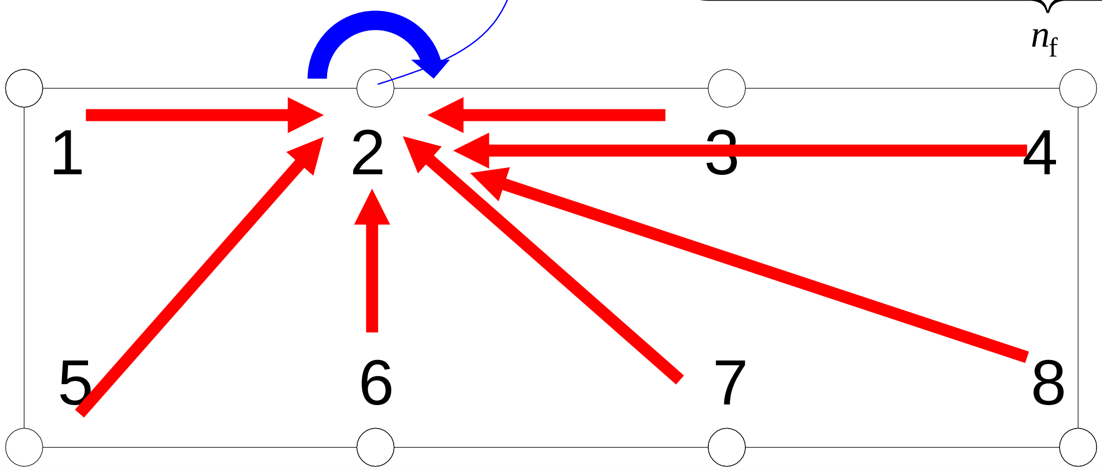
Information of the *i*th node

Information of adjacent nodes

Policy network

comprehensive information vector

$$\boldsymbol{\mu}_i = \{\underbrace{\mu_{i,1} \quad \mu_{i,2} \quad \mu_{i,3} \quad \cdots \quad \mu_{i,n_\mathrm{f}}}_{n_\mathrm{f}}\}$$



action probability

$$\boldsymbol{\pi} = \begin{Bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \end{Bmatrix} = \begin{Bmatrix} 10\% \\ 15\% \\ 15\% \\ 10\% \\ 10\% \\ 15\% \\ 15\% \\ 10\% \end{Bmatrix}$$

Neural network parameters (independent of *n*)

$$Q_i = \mathrm{ReLU}\left[\boldsymbol{\theta}_9^{\mathrm{T}} \mathrm{ReLU}\left(\boldsymbol{\theta}_7 \boldsymbol{\mu}_i^{(T)}; \boldsymbol{\theta}_8 \sum_{i=1}^{n} \boldsymbol{\mu}_i^{(T)}\right)\right]$$

$$\boldsymbol{\pi} = \mathrm{Softmax}\left[\mathbf{S} \cdot \mathbf{Q}(\hat{\boldsymbol{\mu}})\right] = \frac{e^{\mathbf{S} \cdot \mathbf{Q}(\hat{\boldsymbol{\mu}}, i)}}{\sum_{i' \in \mathcal{A}} e^{\mathbf{S} \cdot \mathbf{Q}(\hat{\boldsymbol{\mu}}, i')}}$$

Information of the *i*th node

Information of all nodes

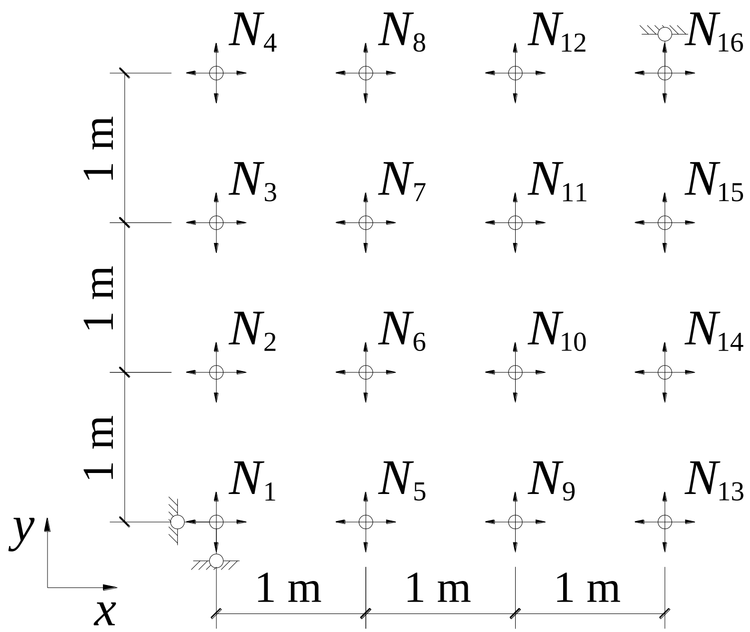
Learning using REINFORCE

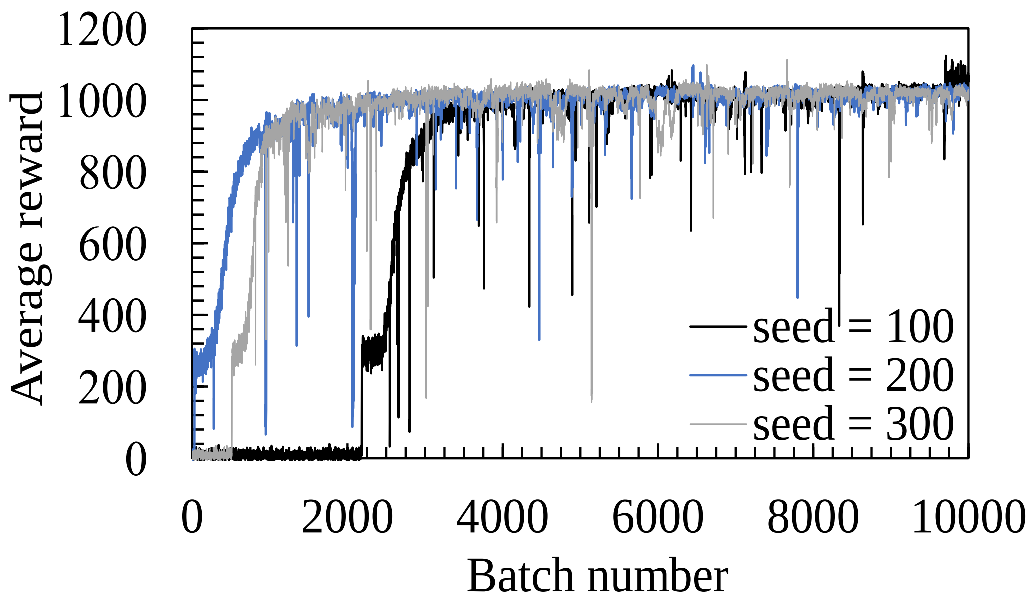
Estimated by Monte-Carlo sampling

Target: Maximize the total reward J

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \nabla_{\theta} J(\boldsymbol{\theta}) \qquad \nabla_{\theta} J(\boldsymbol{\theta}) = \left(\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_1}, \frac{\partial J(\boldsymbol{\theta})}{\partial \theta_2}, \ldots, \frac{\partial J(\boldsymbol{\theta})}{\partial \theta_{n_p}} \right)^{\mathrm{T}}$$

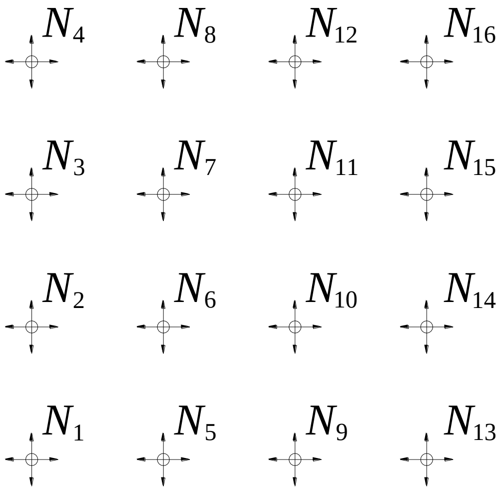
Effect: Probability of good actions ↑

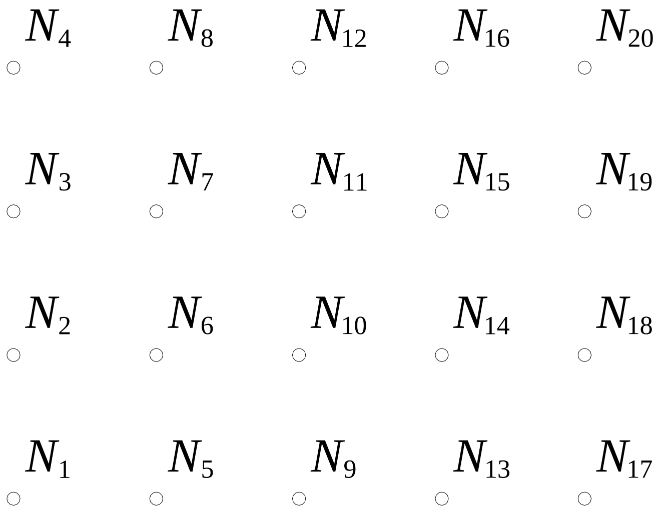




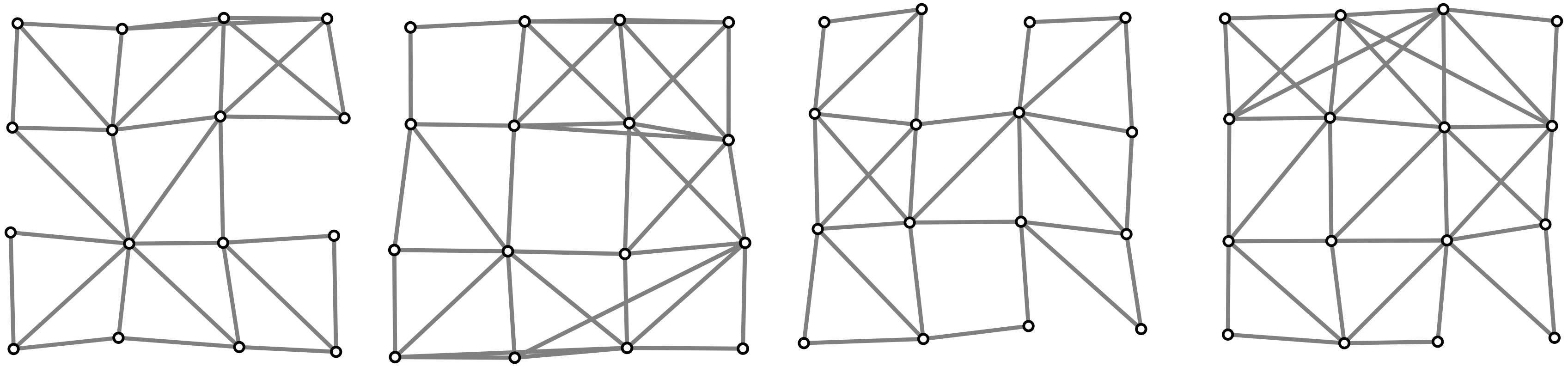
Testing of agent

Initial node-sets

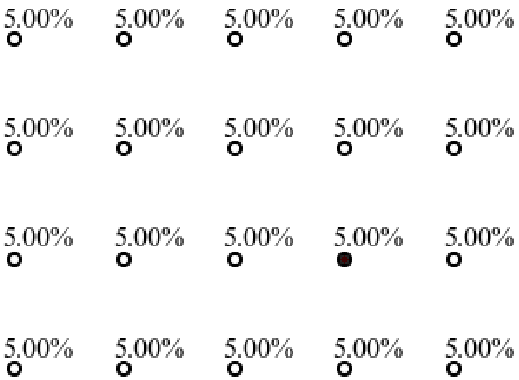


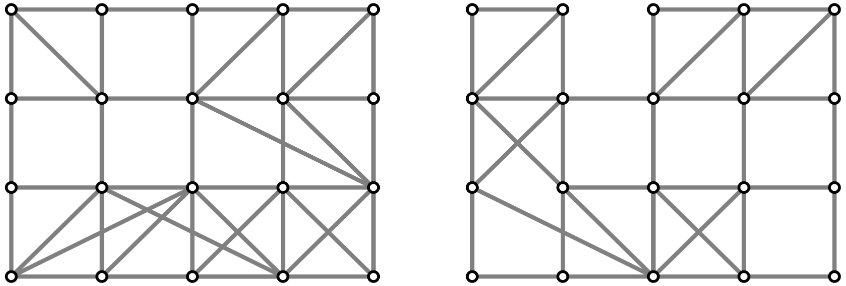


Generated trusses



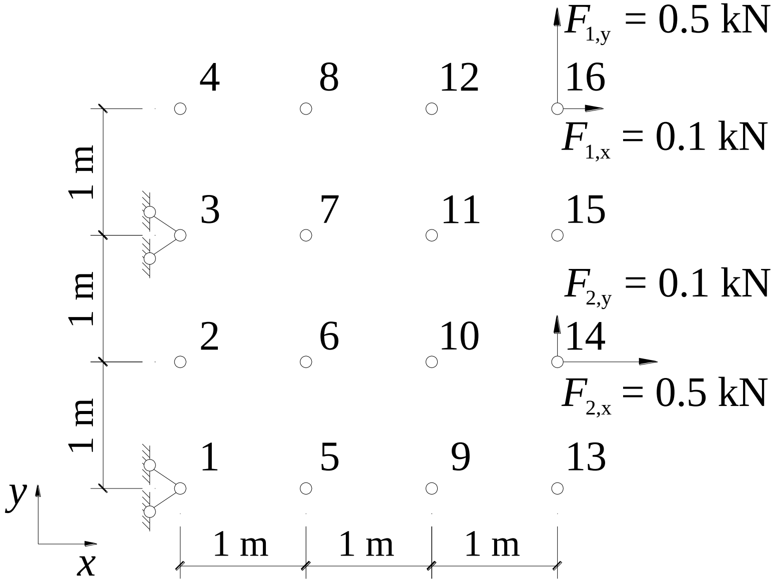
(node-set for training)





(different-sized node-set without re-training)

MGSs for topology optimization



Typical topology optimization problem with singular optimal solutions:

$$\begin{cases} \text{find} & \mathbf{A} \in \text{Section library} \\ \text{min.} & V(\mathbf{A}) \\ \text{s.t.} & \text{stress constraint} \rightarrow \max\limits_{i\in\Omega_m, j\in\{1,2,\cdots,n_L\}} \left(\dfrac{|\sigma_{i,j}(\mathbf{A})|}{\bar{\sigma}} \right) \le 1 \\ & \text{displacement constraint} \rightarrow \max\limits_{i\in\Omega_d, j\in\{1,2,\cdots,n_L\}} \left(\dfrac{|u_{i,j}(\mathbf{A})|}{\bar{u}} \right) \le 1 \\ & A_i \in \{\bar{A}\times 10^{-6}, \bar{A}\} \end{cases}$$

avoid unstable optimal solutions

Step 1: Generate a stable truss and randomly add to *m* members

Step 2: Remove the member with the lowest stress ratio until unstable

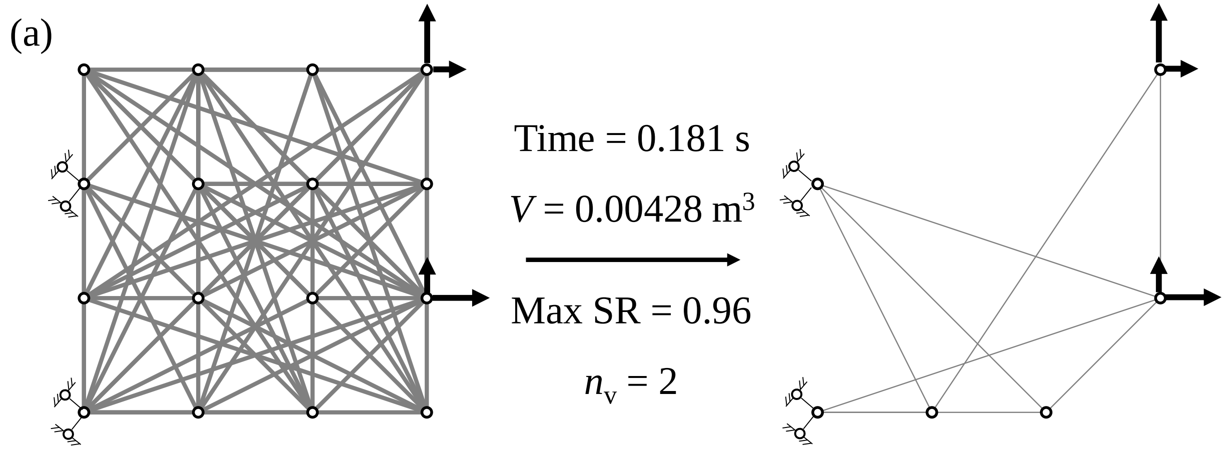
Step 3: Choose cross-sections according to fully-stressed design (FSD)

Numerical example

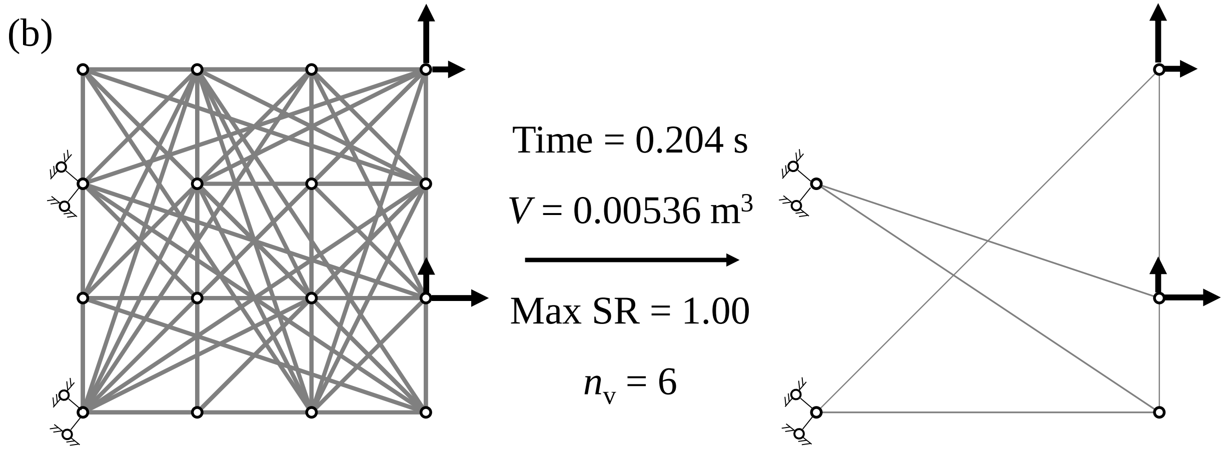
$n = 16 \quad m = 65$

$m_{min} = 2\times16-4 = 28$

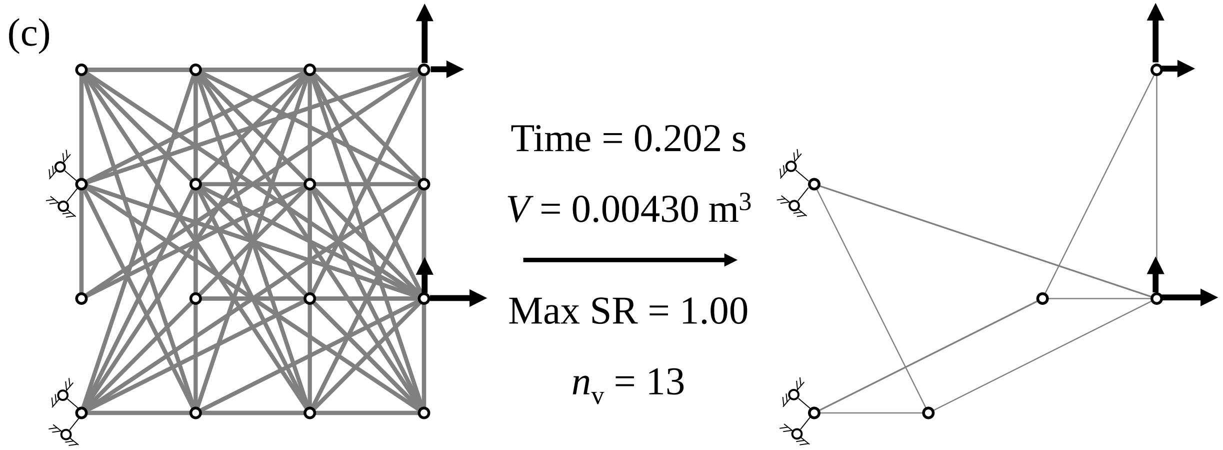
$m_{max} = 16\times(16-1)/2 = 90$



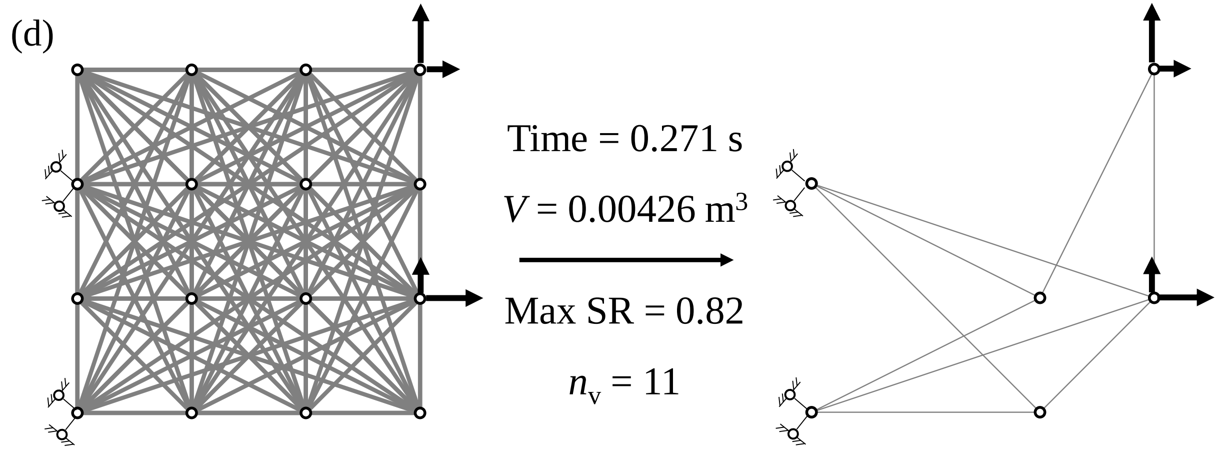
MGS1



MGS2



MGS3



Fully-connected GS

Contributions of this study

- An RL framework is proposed for MGSs
- A trained agent can generate a variety of MGSs
- Application of GE
  → different-sized problems without re-training
- MGSs lead to different optimal solutions
  → more possible to obtain the global optimum

contact: zhushaojuntj@gmail.com

website: www.zhushaojun.com