인공지능 기반 설계 이론 및 사례 연구

13차) Stage 1: Generative Design

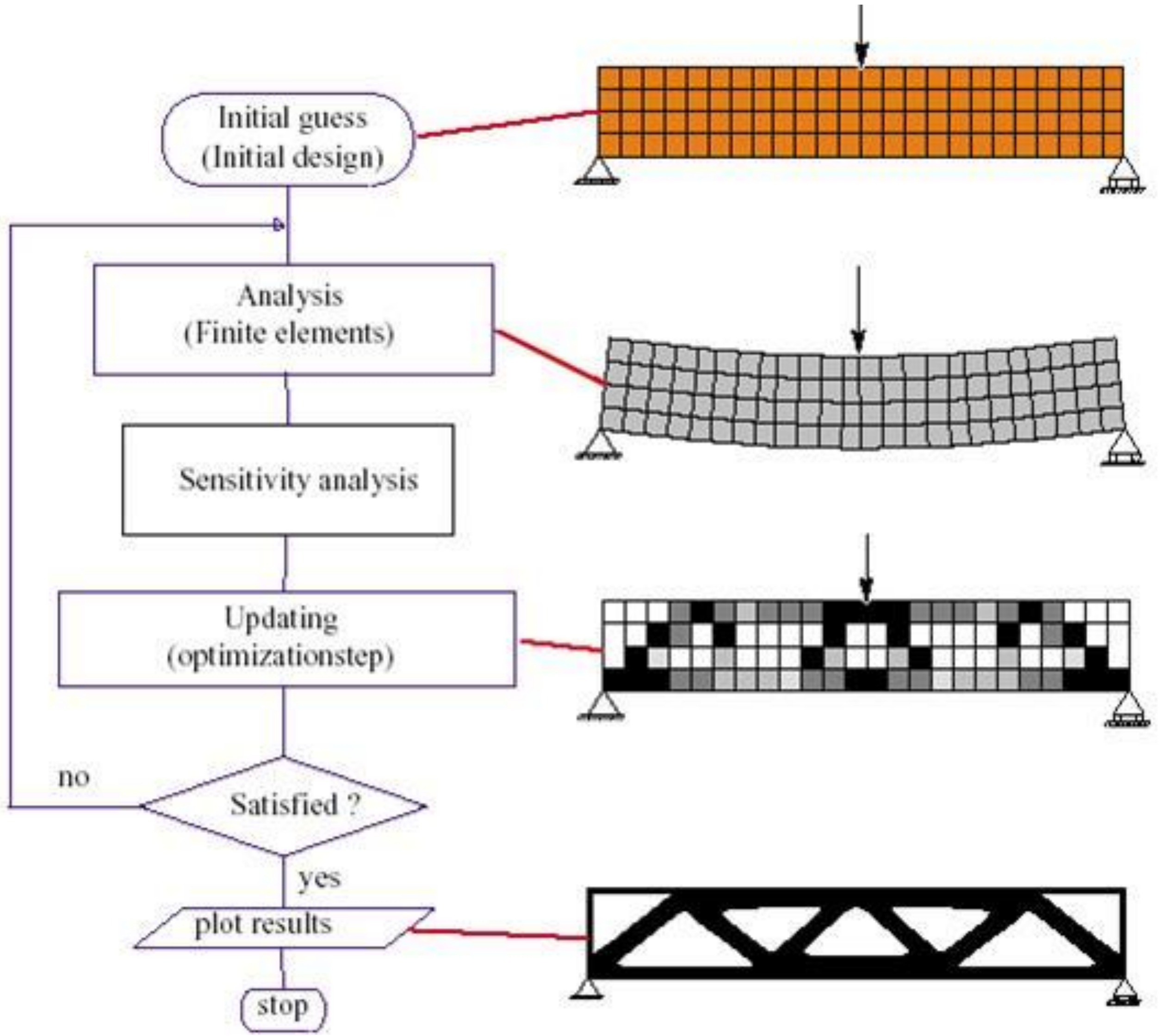
2020년 11월

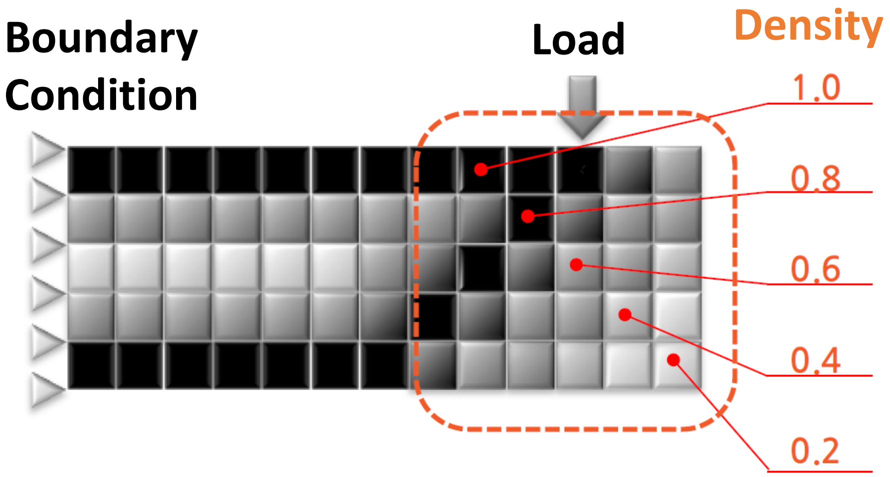
강남우

기계시스템학부
숙명여자대학교

Contents

위상최적화 (Topology Optimization)





- **Objective:** Minimize Compliance
  (=Maximize Stiffness)
- **Design Variables:** Density
- **Constraint:** Volume Fraction

<Bendsøe, M. P., and Sigmund, O., 2003, Topology Optimization: Theory, Methods and Applications, Springer, Berlin>
<http://www.midasuser.com/>

❖ Beam 예제 (88line code)

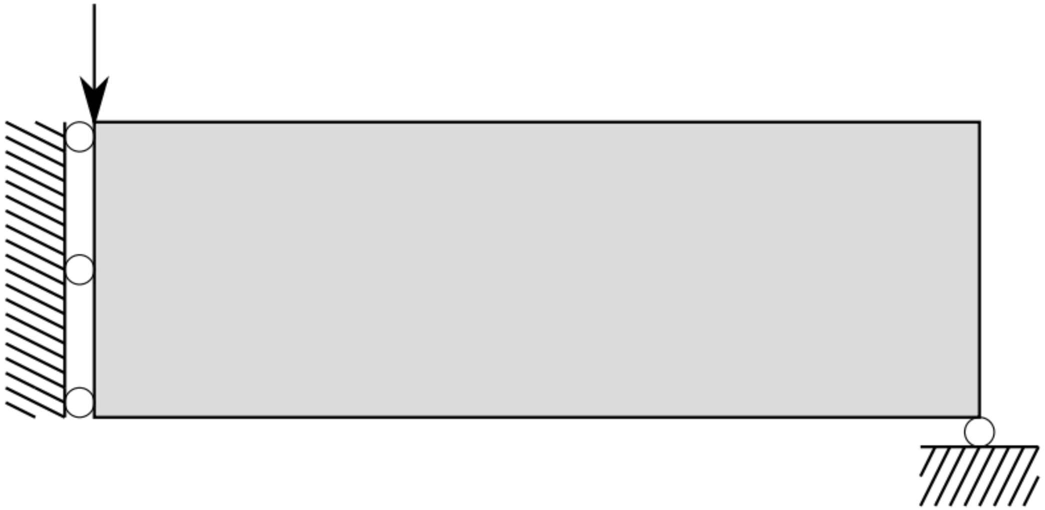


Fig. 1 The design domain, boundary conditions, and external load for the optimization of a symmetric MBB beam.

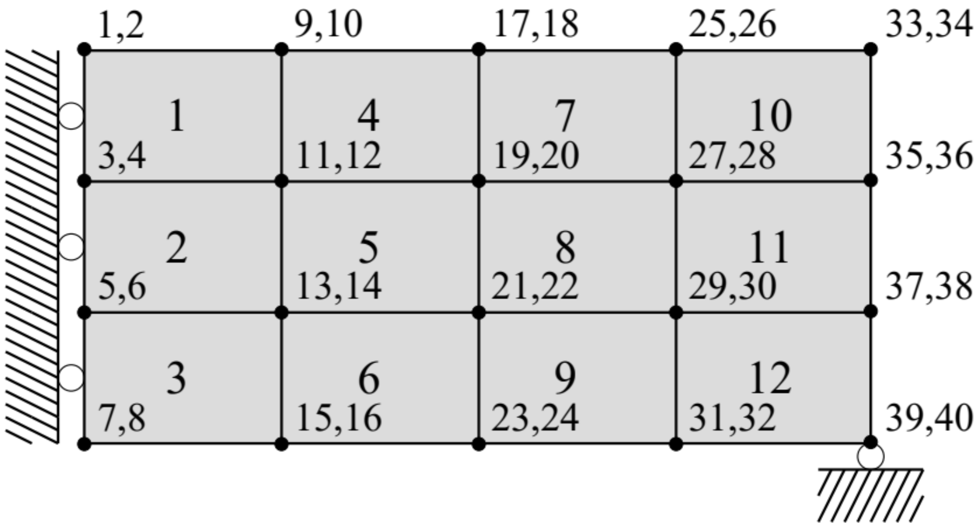


Fig. 2 The design domain with 12 elements.

❖ Optimization Problem

the total compliance (strain energy) is the summation of element-wise compliance

$$\begin{aligned} \min_{\mathbf{x}}: \quad & c(\mathbf{x}) = \mathbf{U}^{\mathrm{T}}\mathbf{K}\mathbf{U} = \sum_{e=1}^{N} E_e(x_e)\mathbf{u}_e^{\mathrm{T}}\mathbf{k}_0\mathbf{u}_e \\ \text{subject to:} \quad & V(\mathbf{x})/V_0 = f \\ & \mathbf{K}\mathbf{U} = \mathbf{F} \\ & \mathbf{0} \leq \mathbf{x} \leq \mathbf{1} \end{aligned}$$

- c :compliance (=strain energy)
- $\mathbf{U}$: global displacement
- $\mathbf{F}$: force vectors
- $\mathbf{K}$: global stiffness matrix
- $\mathbf{u}_e$: element displacement vector,
- $\mathbf{k}_0$: element stiffness matrix for an element with unit Young's modulus
- $\mathbf{x}$: vector of design variables (i.e. the element densities)
- N: number of elements used to discretize the design domain
- $V(\mathbf{x})$: material volume
- V_0: design domain volume
- f: volume fraction

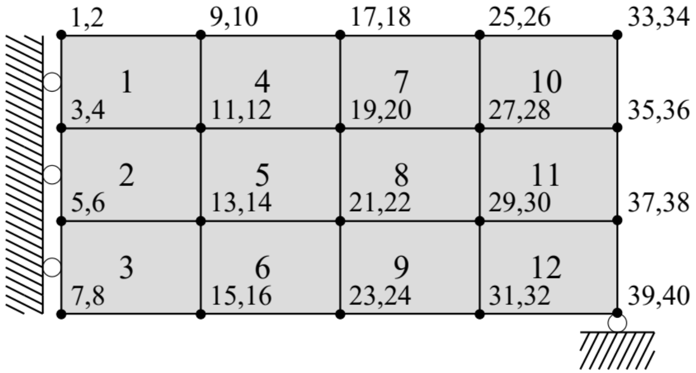


Fig. 2 The design domain with 12 elements.

❖ Modified SIMP approach

$$E_e(x_e) = E_{\min} + x_e^p(E_0 - E_{\min}), \qquad x_e \in [0,1]$$

- x_e: density
- E_e: Young's modulus
- E_0: stiffness of the material
- E_{min}: very small stiffness assigned to void regions
- p: penalization factor (typically p = 3) introduced to ensure black-and-white solutions.

❖ Optimality criteria method (OC)

$$x_e^{\text{new}} = \begin{cases} \max(0, x_e - m) & \text{if } x_e B_e^\eta \le \max(0, x_e - m) \\ \min(1, x_e + m) & \text{if } x_e B_e^\eta \ge \min(1, x_e + m) \\ x_e B_e^\eta & \text{otherwise} \end{cases}$$

- m: positive move limit
- η (= 1/2): numerical damping coefficient

Optimality condition

$$B_e = \frac{-\dfrac{\partial c}{\partial x_e}}{\lambda \dfrac{\partial V}{\partial x_e}}$$

Lagrangian multiplier λ must be chosen so that the volume constraint is satisfied

❖ Sensitivities of the objective function

$$\begin{aligned} \min_{\mathbf{x}}: \quad & c(\mathbf{x}) = \mathbf{U}^{\mathrm{T}}\mathbf{K}\mathbf{U} = \sum_{e=1}^{N} E_e(x_e)\mathbf{u}_e^{\mathrm{T}}\mathbf{k}_0\mathbf{u}_e \qquad E_e(x_e) = E_{\min} + x_e^p(E_0 - E_{\min}) \\ \text{subject to:} \quad & V(\mathbf{x})/V_0 = f \\ & \mathbf{K}\mathbf{U} = \mathbf{F} \\ & \mathbf{0} \le \mathbf{x} \le \mathbf{1} \end{aligned}$$

$$\frac{\partial c}{\partial x_e} = -p x_e^{p-1}(E_0 - E_{\min})\mathbf{u}_e^{\mathrm{T}}\mathbf{k}_0\mathbf{u}$$

$$\frac{\partial V}{\partial x_e} = 1$$

Reduced Gradient Method

$$\frac{d f}{d\mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}} - \frac{\partial f}{\partial \mathbf{u}}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\right)^{-1}\frac{\partial \mathbf{h}}{\partial \mathbf{x}},$$

$$\Downarrow$$

$$\frac{d f}{d\mathbf{x}} = -\mathbf{u}^T\frac{\partial \mathbf{K}}{\partial \mathbf{x}}\mathbf{u}.$$

Assumption: each element has unit volume

❖ Filtering

In order to ensure existence of solutions to the topology optimization problem and to avoid the formation of checkerboard patterns

- Sensitivity filtering

$$\widehat{\frac{\partial c}{\partial x_e}} = \frac{1}{\max(\gamma, x_e)\sum_{i \in N_e} H_{ei}} \sum_{i \in N_e} H_{ei} x_i \frac{\partial c}{\partial x_i}$$

$$H_{ei} = \max\left(0, r_{\min} - \Delta(e,i)\right)$$

- Density filtering

$$\frac{\partial \psi}{\partial x_j} = \sum_{e \in N_j} \frac{\partial \psi}{\partial \tilde{x}_e} \frac{\partial \tilde{x}_e}{\partial x_j} = \sum_{e \in N_j} \frac{1}{\sum_{i \in N_e} H_{ei}} H_{je} \frac{\partial \psi}{\partial \tilde{x}_e}$$

$$\tilde{x}_e = \frac{1}{\sum_{i \in N_e} H_{ei}} \sum_{i \in N_e} H_{ei} x_i$$

SMART DESIGN LAB

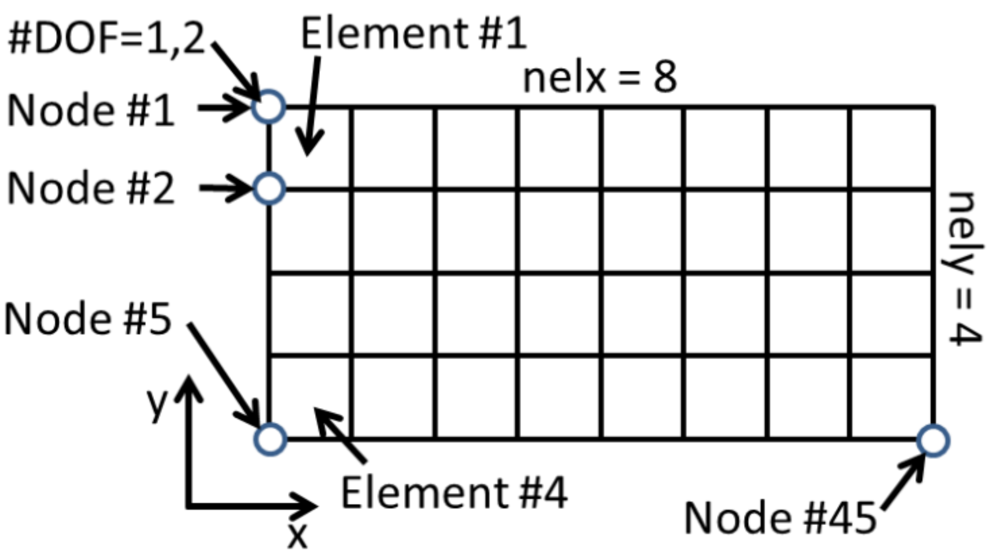
MATLAB Code

$$\min_{\mathbf{x}}: \quad c(\mathbf{x}) = \mathbf{U}^{\mathrm{T}}\mathbf{K}\mathbf{U} = \sum_{e=1}^{N} E_e(x_e)\mathbf{u}_e^{\mathrm{T}}\mathbf{k}_0\mathbf{u}_e$$

$$\text{subject to:} \quad V(\mathbf{x})/V_0 = f$$

$$\mathbf{K}\mathbf{U} = \mathbf{F}$$

$$\mathbf{0} \le \mathbf{x} \le \mathbf{1}$$




```
top88(nelx,nely,volfrac,penal,rmin,ft)
```

- nelx and nely: the number of elements in the horizontal and vertical direction, respectively
- volfrac: the prescribed volume fraction f,
- penal: the penalization power p
- rmin: the filter radius rmin (divided by the element size)
- ft: specifies whether sensitivity filtering (ft = 1) or density filtering (ft = 2) should be used

$$E_e(x_e) = E_{\min} + x_e^p(E_0 - E_{\min})$$

$$H_{ei} = \max\left(0, r_{\min} - \Delta(e,i)\right)$$

```
top88(60,20,0.5,3,1.5,1)
```

MATLAB Code

```matlab
%%%% AN 88 LINE TOPOLOGY OPTIMIZATION CODE Nov, 2010 %%%%
function top88(nelx,nely,volfrac,penal,rmin,ft)
%% MATERIAL PROPERTIES
E0 = 1;
Emin = 1e-9;
nu = 0.3;
%% PREPARE FINITE ELEMENT ANALYSIS
A11 = [12  3 -6 -3;  3 12  3  0; -6  3 12 -3; -3  0 -3 12];
A12 = [-6 -3  0  3; -3 -6 -3 -6;  0 -3 -6  3;  3 -6  3 -6];
B11 = [-4  3 -2  9;  3 -4 -9  4; -2 -9 -4 -3;  9  4 -3 -4];
B12 = [ 2 -3  4 -9; -3  2  9 -2;  4  9  2  3; -9 -2  3  2];
KE = 1/(1-nu^2)/24*([A11 A12;A12' A11]+nu*[B11 B12;B12' B11]);
nodenrs = reshape(1:(1+nelx)*(1+nely),1+nely,1+nelx);
edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,nelx*nely,1);
edofMat = repmat(edofVec,1,8)+repmat([0 1 2*nely+[2 3 0 1] -2 -1],nelx*nely,1);
iK = reshape(kron(edofMat,ones(8,1))',64*nelx*nely,1);
jK = reshape(kron(edofMat,ones(1,8))',64*nelx*nely,1);
% DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
F = sparse(2,1,-1,2*(nely+1)*(nelx+1),1);
U = zeros(2*(nely+1)*(nelx+1),1);
fixeddofs = union([1:2:2*(nely+1)],[2*(nelx+1)*(nely+1)]);
alldofs = [1:2*(nely+1)*(nelx+1)];
freedofs = setdiff(alldofs,fixeddofs);
```

Boundary Condition & Load Vectors

MATLAB Code

```
%% PREPARE FILTER
iH = ones(nelx*nely*(2*(ceil(rmin)-1)+1)^2,1);
jH = ones(size(iH));
sH = zeros(size(iH));
k = 0;
for i1 = 1:nelx
  for j1 = 1:nely
    e1 = (i1-1)*nely+j1;
    for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
      for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
        e2 = (i2-1)*nely+j2;
        k = k+1;
        iH(k) = e1;
        jH(k) = e2;
        sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
      end
    end
  end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);
```


SMART DESIGN LAB

MATLAB

```matlab
%% INITIALIZE ITERATION
x = repmat(volfrac,nely,nelx);
xPhys = x;
loop = 0;
change = 1;
%% START ITERATION
while change > 0.01
  loop = loop + 1;
  %% FE-ANALYSIS
  sK = reshape(KE(:)*(Emin+xPhys(:)'.^penal*(E0-Emin)),64*nelx*nely,1);
  K = sparse(iK,jK,sK); K = (K+K')/2;
  U(freedofs) = K(freedofs,freedofs)\F(freedofs);
  %% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
  ce = reshape(sum((U(edofMat)*KE).*U(edofMat),2),nely,nelx);
  c = sum(sum((Emin+xPhys.^penal*(E0-Emin)).*ce));
  dc = -penal*(E0-Emin)*xPhys.^(penal-1).*ce;
  dv = ones(nely,nelx);
  %% FILTERING/MODIFICATION OF SENSITIVITIES
  if ft == 1
    dc(:) = H*(x(:).*dc(:))./Hs./max(1e-3,x(:));
  elseif ft == 2
    dc(:) = H*(dc(:)./Hs);
    dv(:) = H*(dv(:)./Hs);
  end
  %% OPTIMALITY CRITERIA UPDATE OF DESIGN VARIABLES AND PHYSICAL DENSITIES
  l1 = 0; l2 = 1e9; move = 0.2;
  while (l2-l1)/(l1+l2) > 1e-3
    lmid = 0.5*(l2+l1);
    xnew = max(0,max(x-move,min(1,min(x+move,x.*sqrt(-dc./dv/lmid)))));
    if ft == 1
      xPhys = xnew;
    elseif ft == 2
      xPhys(:) = (H*xnew(:))./Hs;
    end
    if sum(xPhys(:)) > volfrac*nelx*nely, l1 = lmid; else l2 = lmid; end
  end
  change = max(abs(xnew(:)-x(:)));
  x = xnew;
```

$$\frac{\partial c}{\partial x_e} = -p x_e^{p-1}(E_0 - E_{\min})\mathbf{u}_e^{\mathrm{T}}\mathbf{k}_0\mathbf{u}$$

$$\frac{\partial V}{\partial x_e} = 1$$

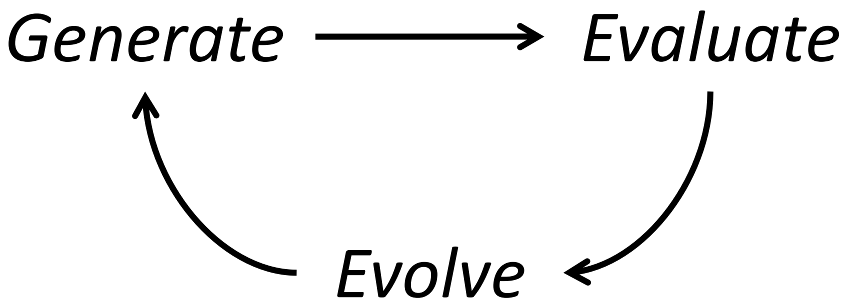
$$x_e^{\text{new}} = \begin{cases} \max(0, x_e - m) & \text{if } x_e B_e^{\eta} \leq \max(0, x_e - m) \\ \min(1, x_e + m) & \text{if } x_e B_e^{\eta} \geq \min(1, x_e - m) \\ x_e B_e^{\eta} & \text{otherwise} \end{cases}$$

MATLAB Code

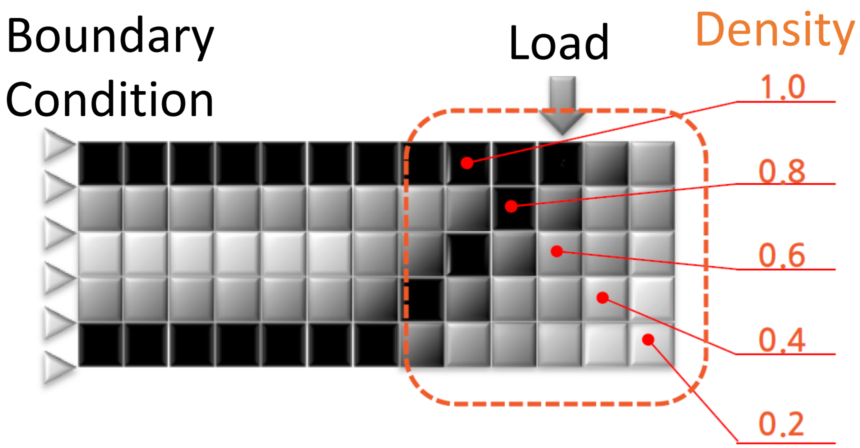
```
%% PRINT RESULTS
fprintf(' It.:%5i Obj.:%11.4f Vol.:%7.3f ch.:%7.3f\n',loop,c, ...
  mean(xPhys(:)),change);
%% PLOT DENSITIES
colormap(gray); imagesc(1-xPhys); caxis([0 1]); axis equal; axis off; drawnow;
```

What is Generative Design?

Generative Design



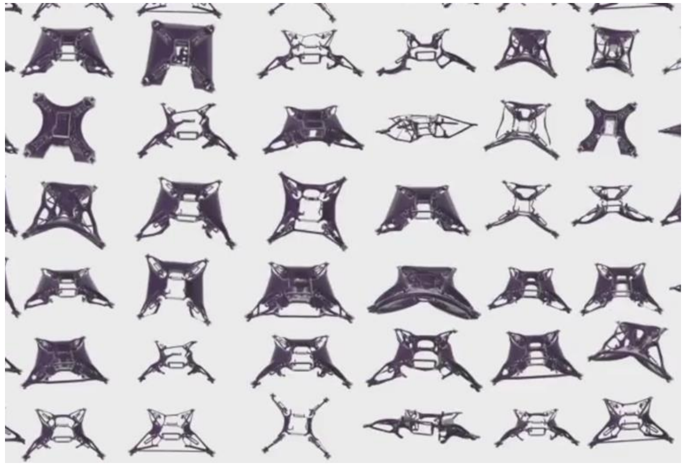
Topology Optimization



- Objective: Minimize Compliance
- Design Variables: Density
- Constraint: Volume Fraction

What if we vary
Parameters* of *Problem Definition
in Topology Optimization?

What is Generative Design?



	Generative Design	Topology Optimization	Parametric Design
Objective	Explore feasible *design sets* (*thousands of designs*)	Find the *optimal design*	Explore *design sets*
Method	Vary parameters of *problem definition* in Topology Optimization	Optimize material layout within given design space	Vary parameters of *geometry* directly

Kang (2019)

Data-driven Generative Design

❖ Wheel Generation based on Reference Designs

Input

Reference Design

ID: 110

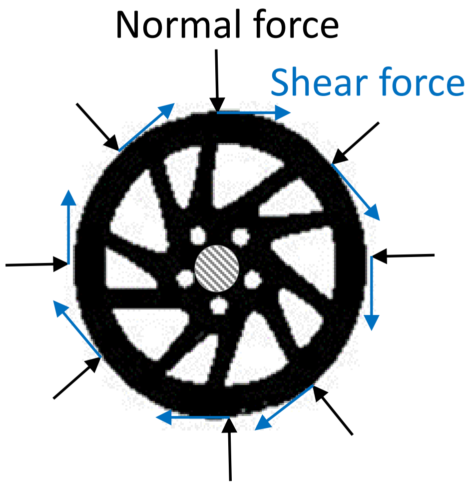
Design Parameters

Similarity weight λ

Force ratio

= Normal / Shear

Topology Optimization

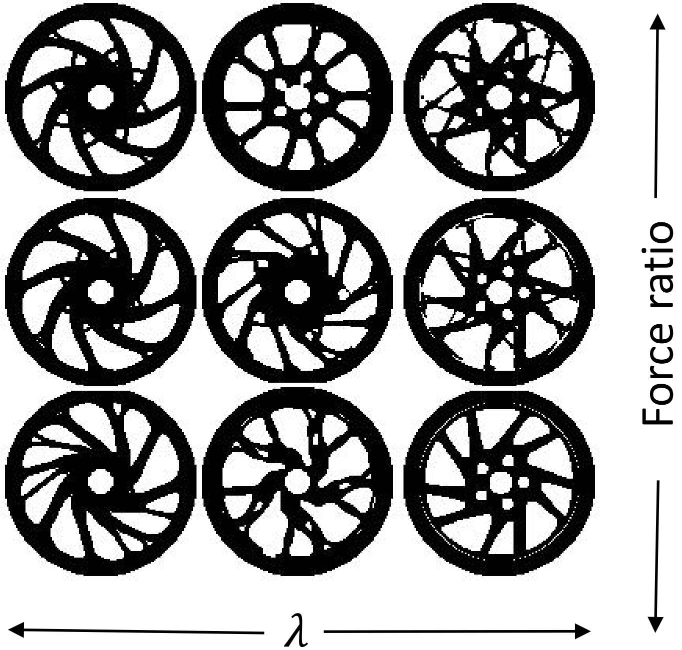


$$\min_{\mathbf{x}} f(\mathbf{x}) = \underbrace{\mathbf{U}^T\mathbf{K}(\mathbf{x})\mathbf{U}}_{\text{Compliance}} + \underbrace{\lambda\|\mathbf{x} - \mathbf{x}_r\|_1}_{\text{L1 distance}}$$

λ: Weight for reference design

$\mathbf{x}_r$: Reference design

Output



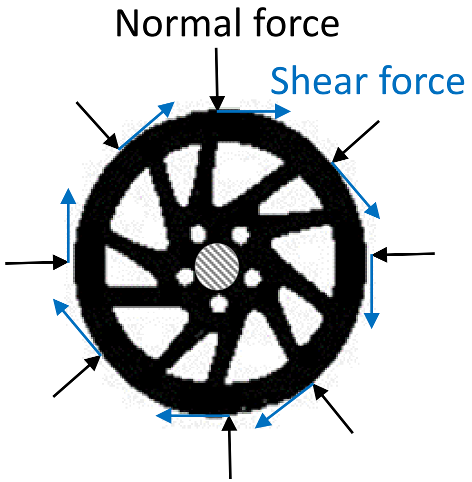
Oh et al (2019)

SMART DESIGN LAB

Data-driven Generative Design

❖ Wheel Generation based on Reference Designs

Topology Optimization



$$\min_{\mathbf{x}} f(\mathbf{x}) = \mathbf{U}^T\mathbf{K}(\mathbf{x})\mathbf{U} + \lambda\|\mathbf{x} - \mathbf{x}_r\|_1$$

Compliance **L1 distance**

λ: Weight for reference design

$\mathbf{x}_r$: Reference design

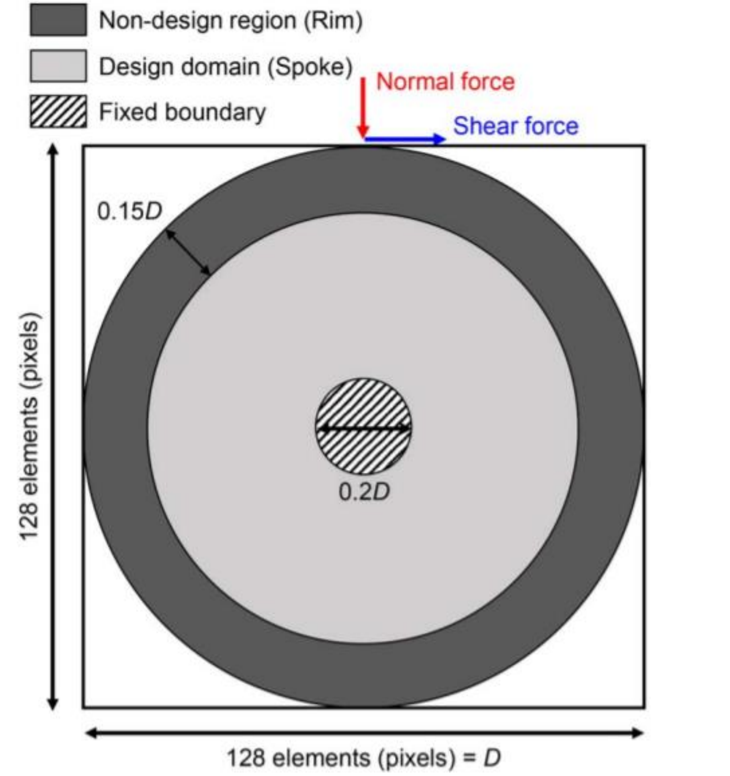


Fig. 2 Design domain and boundary conditions of a 2D wheel design

$$\frac{\partial}{\partial \mathbf{x}}(\lambda\|\mathbf{x}^* - \mathbf{x}\|_1) \cong -\lambda\mathbf{x}^*$$

$$x^* = 1 \rightarrow -\lambda$$
$$x^* = 0 \rightarrow 0$$

 Oh et al (2019)

What Questions Do You Have?

nwkang@sm.ac.kr

www.smartdesignlab.org