Generating Dataset Using Random Positions

Clearing Workspace

```
clc;
clear;
close all;
```

Initialization of global variables

```
global_variables
```

Initialization Of variables vounds and Veloscicity factor (Young Model)

```
VarMin = 0.000001;        % Lower Bound of Decision Variables
VarMax =  1;           % Upper Bound of Decision Variables
E = 2e5;
```

Initialization of parameters

```
nodes = load('nodes')
```

```
nodes = 6×2
         0         0
    0.5000         0
    1.0000         0
    1.0000    0.5000
    0.5000    0.5000
         0    0.5000
```

```
connectivity_matrix = load('connectivity_matrix')
```

```
connectivity_matrix = 6×6
     0     1     0     1     1     1
     1     0     1     1     1     1
     0     1     0     1     1     1
     1     1     1     0     1     0
     1     1     1     1     0     1
     1     1     1     0     1     0
```

```
nodal_force = load('nodal_force')
nodal_force = 6×2
           0           0
       -2000           0
           0       -2000
        2000           0
           0        2000
           0           0
```

```
fixed_nodes = load('fixed_nodes')
```

```
fixed_nodes = 1×4
     1     2    11    12
```

```
number_of_nodes = size(nodes, 1)
```

```
number_of_nodes = 6
```

```
number_of_bars = sum(sum(connectivity_matrix))/2;
bars = zeros(2, number_of_bars)
```

```
bars = 2×13
     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0
```

```
lengths_of_bars = zeros(1, number_of_bars)
```

```
lengths_of_bars = 1×13
     0     0     0     0     0     0     0     0     0     0     0     0     0
```

```
k = 0;
for i=1:number_of_nodes
    for j=i+1:number_of_nodes
        if connectivity_matrix(i,j)== 1
            k = k + 1;
            lengths_of_bars(k) = sqrt((nodes(i,1)-nodes(j,1))^2+(nodes(i,2)-nodes(j,2))^2);
            bars(1,k)= i;
            bars(2,k)= j;
        end
    end
end
```

```
population_size = 100000;
```

Creating Individuals data structure

every individual will have the structure (position, energy), where position is a number_of_bars-dimensional vector and energy if the deformation energy with respect to the lengths of bars.

```
agent.position=[];
agent.energy=[];
```

now we will create the population with population_size agents (i.e individuals)

```
population=repmat(agent,population_size,1);
```

```
dataset = zeros(population_size,number_of_bars);
energy = zeros(population_size, 1);
```

```
cost_function = @compliance;
for i=1:population_size
    population(i).position = unifrnd(VarMin, VarMax, [1 number_of_bars]);
    % Evaluation
    population(i).energy=cost_function(population(i).position);


end
```

```
for i=1:population_size
    for j=1:number_of_bars
    dataset(i,j)=population(i).position(j);
    end
end
```

```
for i=1:population_size
    energy(i)=population(i).energy;
end
dataset = [dataset energy(:)];
```

Saving created data points to a Dataset

```
csvwrite("Dataset.csv",dataset)
```