Surrogate Assisted Optimization of a 1D Heterogeneous Bar

Ryan (Yunsheng) Wang
Mech 293 Final Project Report

Abstract

This project looks at compliance minimization for a 1D bar whose Young's modulus varies along the length. I use a standard linear finite element model as the ground truth solver. Then I train a fully connected neural network to learn the mapping from element moduli to two outputs: compliance and mass. Since the problem is only 1D, it can easily become trivial that the best answer becomes "make everything as stiff as possible". To avoid that, I use a nonlinear relationship between Young's modulus and density, so very stiff elements become much heavier. With this setup, the optimizer must trade off stiffness and weight instead of pushing all elements to the maximum modulus. I compare a direct FEA based random search against a two stage surrogate assisted strategy. In the baseline run, the surrogate reaches a compliance RMSE of about 1.27×10^{-2} and a mass RMSE of about 9.77×10^{-3} on test data. Using the surrogate for prescreening, the method finds a better design than direct random search while cutting the number of FEA evaluations by about a factor of 40.

1 Introduction

Compliance minimization with stiffness or mass constraints is a classic topic in structural optimization. A common workflow is to discretize a structure into finite elements and use design variables to control the material distribution. These problems are often nonconvex and high dimensional, and evaluating many candidate designs with FEA can be costly.

Surrogate modeling is one way to reduce this cost. Instead of running FEA for every candidate design, we first fit a regression model that approximates the input and output mapping. Once trained, the surrogate is cheap to evaluate and can be used inside an optimization loop to reduce the number of expensive FEA calls.

In this project I use a simple test case: a 1D, linearly elastic bar with heterogeneous material properties. The design variables are Young's moduli, constrained between a minimum and a maximum value. The bar is loaded in axial tension, and the goal is to minimize compliance under a global mass limit. Because the system is 1D, it is easy to end up with a trivial solution. To make the problem more meaningful, I strengthen the coupling between modulus and density so that high moduli become disproportionately heavy. This forces a real tradeoff between stiffness and mass and makes the 1D setup a better test for surrogate assisted optimization.

2 Problem Setup

2.1 Physical model

We consider a straight bar of length L and constant cross-sectional area A, fixed at the left end and loaded by an axial force F at the right end. The bar is modeled as a 1D linear elastic continuum with spatially varying Young's modulus $E(x)$.

The axial displacement $u(x)$ satisfies

$$-\frac{d}{dx}\big(AE(x)\tfrac{du}{dx}\big) = 0, \quad 0 < x < L, \tag{1}$$

with boundary conditions

$$u(0) = 0, \qquad AE(L)\frac{du}{dx}\bigg|_{x=L} = F. \tag{2}$$

2.2 Discretization and design variables

The bar is divided into n equal elements with length $\Delta L = L/n$. In each element $i = 1, \ldots, n$, the modulus E_i is constant. The design vector is

$$\mathbf{E} = (E_1, E_2, \ldots, E_n)^\top, \qquad E_{\min} \le E_i \le E_{\max}. \tag{3}$$

In all experiments, I use $L = 1$, $A = 1$, $F = 1$, $E_{\min} = 1$, $E_{\max} = 10$, and $n = 10$.

2.3 Elasticity and density tradeoff

To connect stiffness and mass, I use a simple nonlinear mapping from modulus to density. Let $\rho_{\min}$ and $\rho_{\max}$ be the densities corresponding to $E_{\min}$ and $E_{\max}$. For an element with modulus E_i, define

$$t_i = \frac{E_i - E_{\min}}{E_{\max} - E_{\min}} \in [0, 1], \qquad \rho_i = \rho_{\min} + t_i^p \, (\rho_{\max} - \rho_{\min}), \tag{4}$$

with $\rho_{\min} = 1$, $\rho_{\max} = 3$, and exponent $p = 2$ in the baseline case.

When $p = 1$, density grows linearly with modulus and the optimizer tends to prefer large E_i everywhere. For $p > 1$, high modulus values become much heavier, which makes intermediate E_i more attractive and helps avoid the trivial "all-max" design.

The total mass is

$$M(\mathbf{E}) = \sum_{i=1}^{n} \rho_i A \, \Delta L = \sum_{i=1}^{n} \rho_i \frac{L}{n}. \tag{5}$$

For the fully stiff bar ($E_i = E_{\max}$), we have $\rho_i = \rho_{\max} = 3$, so $M_{\text{full}} = 3$. In the baseline setting I impose

$$M(\mathbf{E}) \le M_{\max} = 0.7 \, M_{\text{full}} = 2.1, \tag{6}$$

so the design must be 30% lighter than the fully stiff case.

2.4 Objective and constraint

Let $\mathbf{u}$ be the nodal displacement vector from FEA, and $\mathbf{f}$ be the nodal load vector. Compliance is

$$C(\mathbf{E}) = \mathbf{f}^\top \mathbf{u} = F \, u_N, \tag{7}$$

where u_N is the displacement at the loaded right-end node.

The optimization problem is

$$\min_{\mathbf{E}} \; C(\mathbf{E}) \tag{8}$$

$$\text{s.t.} \;\; M(\mathbf{E}) \le M_{\max}, \tag{9}$$

$$E_{\min} \le E_i \le E_{\max}, \quad i = 1, \ldots, n. \tag{10}$$

3 Finite Element Model and Verification

3.1 Element stiffness and global system

Each element i connects nodes i and $i+1$ and is modeled with a linear 1D bar element. The local stiffness matrix is

$$\mathbf{k}^{(i)} = \frac{AE_i}{\Delta L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \tag{11}$$

The global stiffness matrix $\mathbf{K} \in \mathbb{R}^{(n+1)\times(n+1)}$ is assembled by adding each local matrix into the correct rows and columns. The load vector has one nonzero entry at the right end: $f_N = F$.

The left node is fixed, $u_0 = 0$, by removing the corresponding row and column from $\mathbf{K}$. Let $\tilde{\mathbf{K}}$ and $\tilde{\mathbf{f}}$ denote the reduced system. Solving

$$\tilde{\mathbf{K}}\,\tilde{\mathbf{u}} = \tilde{\mathbf{f}} \tag{12}$$

gives the unknown displacements. The full displacement vector is obtained by inserting $u_0 = 0$. Then compliance is computed as $C = Fu_N$.

3.2 Analytical reference and numerical check

For a homogeneous bar with $E_i \equiv E$, the closed-form solution is

$$u(L) = \frac{FL}{AE}, \qquad C_{\text{ana}}(E) = Fu(L) = \frac{F^2 L}{AE}. \tag{13}$$

I use this as a check for the FEA code. For several values of E in $[E_{\min}, E_{\max}]$, I compute $C_{\text{FE}}(E)$ using a uniform $\mathbf{E}$ and compare it to $C_{\text{ana}}(E)$.

Figure 1 shows the comparison: the two curves overlap, and the maximum relative error is about 2.6×10^{-15}, which is essentially double precision. This suggests the assembly and boundary conditions are implemented correctly.

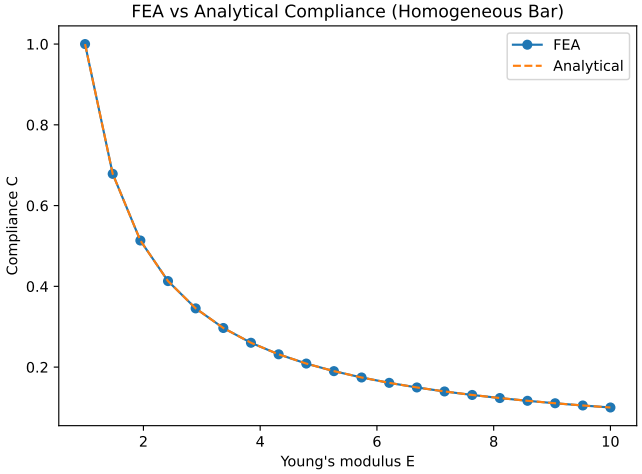

Figure 1: Verification of the finite element model.

4 Neural Network Surrogate

4.1 Dataset generation

To train a surrogate, I generate data by sampling random stiffness distributions. Each E_i is drawn independently and uniformly from $[E_{\min}, E_{\max}]$. For each sample $\mathbf{E}^{(k)}$, I compute:

- Compliance $C^{(k)} = C(\mathbf{E}^{(k)})$ using FEA.
- Mass $M^{(k)} = M(\mathbf{E}^{(k)})$ using the density model.

This creates pairs $(\mathbf{E}^{(k)}, \mathbf{y}^{(k)})$ where $\mathbf{y}^{(k)} = (C^{(k)}, M^{(k)})^\top$. In the baseline run, I use $N_{\text{train}} = 4000$ training samples and $N_{\text{test}} = 1000$ test samples.

Before training, I standardize each input dimension and each output component using the training set mean and standard deviation. This makes optimization more stable.

4.2 Network architecture and training

The surrogate is a fully connected neural network:

- Input dimension: $n = 10$ (element-wise moduli).
- Hidden layers: three layers, each with 64 neurons and ReLU.
- Output dimension: 2 (compliance and mass).

I train it using Adam to minimize mean-squared error on standardized outputs. The learning rate is 10^{-3}, batch size is 64, and training runs for 200 epochs.

Figure 2 shows the training and test MSE curves. The test loss decreases and levels off around 2.5×10^{-2}, which indicates reasonable generalization on unseen random designs.

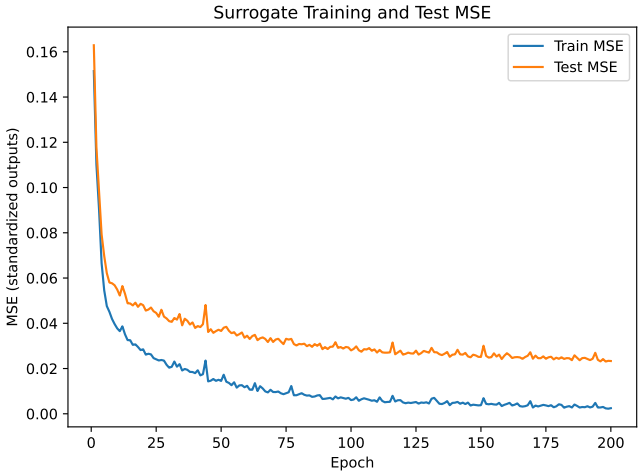

Figure 2: Training and test mean-squared error (MSE) versus epoch (standardized outputs).

4.3 Error distribution in physical units

To interpret the errors in physical units, I de-standardize the predictions and compute

$$e_C^{(k)} = \hat{C}^{(k)} - C^{(k)}, \qquad e_M^{(k)} = \hat{M}^{(k)} - M^{(k)} \tag{14}$$

on the test set. Figure 3 shows the error histograms.

In the baseline run, the RMSE values are

$$\text{RMSE}_C \approx 1.27 \times 10^{-2}, \qquad \text{RMSE}_M \approx 9.77 \times 10^{-3}.$$

These are small compared to typical compliance and mass values in the dataset (compliance is around 10^{-1} and mass is around 2). The errors are roughly centered at zero with relatively light tails. That said, for optimization we care about picking the best few designs, so even small errors can still affect ranking.

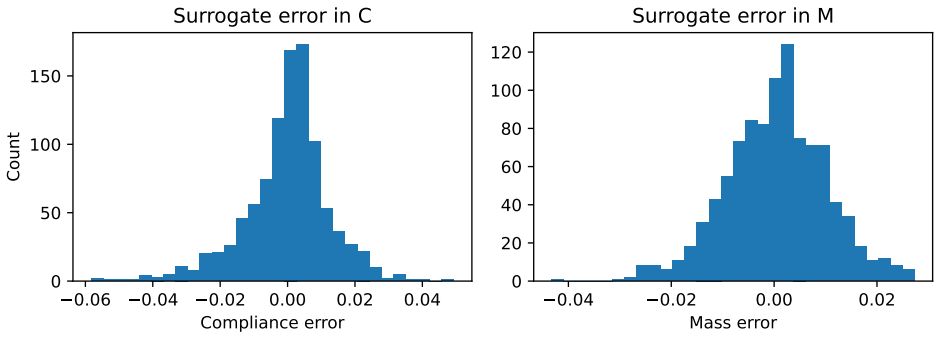

Figure 3: Prediction error histograms on the test set: compliance (left) and mass (right).

5 Implementation Overview

- **FEA solver:** `fea_bar` takes $(E_1, \ldots, E_{10})$, assembles the standard 1D bar stiffness matrix, applies the fixed boundary at the left end, solves the reduced linear system, and returns compliance $C = Fu(L)$.
- **Density and mass:** `density_from_E` and `mass_from_E` implement the nonlinear modulus–density relation ($p = 2$) and sum element masses to get $M(\mathbf{E})$.
- **Surrogate training:** `train_surrogate` samples random designs, runs FEA to get (C, M), standardizes the data, trains the neural network, and saves loss curves and diagnostics.
- **Optimization loops:** `direct_random_search` performs baseline random search using FEA. `surrogate_prescreen_search` uses the surrogate to filter many candidates and then re-evaluates a small set with FEA.

6 Optimization Strategies and Baseline Results

6.1 Direct FEA based random search

1. Sample N_{direct} candidate designs by drawing each E_i uniformly from $[E_{\min}, E_{\max}]$.
2. For each candidate, compute C and M using FEA and the density model.
3. Discard candidates with $M > M_{\max}$.
4. Report the feasible design with the smallest compliance.

In the baseline experiment, $N_{\text{direct}} = 2000$ FEA evaluations are used.

6.2 Surrogate prescreening with FEA reevaluation

The surrogate-assisted method is a simple two-stage pipeline:

1. Sample N_{surr} candidate designs (here $N_{\text{surr}} = 20000$).
2. Use the surrogate to predict $(\hat{C}, \hat{M})$ for each design.
3. Discard designs with predicted mass $\hat{M} > M_{\max}$.
4. Pick the K designs with smallest predicted compliance $\hat{C}$ (here $K = 50$).
5. Reevaluate these K designs using the true FEA solver and mass model.
6. Report the best design based on *true* compliance among these K.

6.3 Numerical comparison

Table 1 summarizes the baseline results (quadratic density law $p = 2$ and mass limit $\alpha = 0.7$). Both methods return feasible designs with mass close to $M_{\max} = 2.1$.

Table 1: Baseline optimization results for direct random search and surrogate-assisted search.

Method	C_{best}	M_{best}	FEA calls	Surrogate calls
Direct random search	0.140773	2.086059	2000	0
Surrogate pre-screening	0.135981	2.054015	50	20000

In this run, surrogate prescreening finds a slightly lower compliance (0.13598 vs. 0.14077) with a similar mass, while reducing FEA calls from 2000 to 50. The surrogate is not perfect, so the true best design among all 20000 candidates may not land in the top 50 by predicted compliance. Still, the ranking is good enough here to make the two stage approach effective.

Conclusion

This project builds a 1D FEA solver for compliance minimization in a heterogeneous bar, then trains a small fully connected network to predict compliance and mass from moduli. To avoid choosing the stiffest material everywhere, stiffness is coupled to density through a quadratic law ($p = 2$) and a global mass constraint, which creates a meaningful stiffness and weight tradeoff.

The surrogate achieves low prediction errors on random designs and is effective for screening: instead of evaluating all candidates with FEA, many designs are scored cheaply by the surrogate and only a small top set is reevaluated with FEA to ensure feasibility and accurate ranking. In the baseline experiment, the online optimization stage reduces FEA calls from 2000 to 50 (about $40\times$) while returning a design with improved compliance and similar mass. This gain should be interpreted as an online saving; an offline cost remains because the surrogate is trained using 4000 FEA labeled samples, which can be amortized if multiple optimizations are performed.

Overall, the study demonstrates a complete surrogate assisted optimization pipeline, and it can be extended to higher dimensional problems where FEA is more expensive.

References

[1] M. P. Bendsøe and O. Sigmund. *Topology Optimization: Theory, Methods, and Applications*. Springer, 2003.

[2] A. Forrester, A. Sóbester, and A. Keane. *Engineering Design via Surrogate Modelling: A Practical Guide*. Wiley, 2009.