Truss Structures Optimization Using Genetic Algorithms

A. Bahaj R. Haffadi

June 20, 2019

Contents

0.1 Structural Optimization

0.1.1 Introduction

Optimization is a concern of the various actors in many areas including consultancies in mechanical disciplines. An optimization problem seeks to find the better solution to a problem respecting a number of constraints. For shape optimization, the objective is to seek the best structure that ensures efficient performance at minimum cost: minimum weight , minimum volume, minimum deformation energy or other. According to G. Allaire [1], we distinguish between three categories of shape optimization problems. One that seeks the best dimensions of a structure. Another that change only the coordinates of the structure borders without changing its topology. And last, gives the possibility to modify the initial topology of the structure without restrictions to find the best possible shape: it is the topology optimization.

0.1.2 Calculating Deformation Energy

A reticulated structure is designed to be rigid and balanced. Moreover it can be subjected to mechanical or thermal forces of deformation influencing its stiffness. These forces are applied on the nodes of the structure what produces a displacement of these nodes. Optimizing a reticulated structure means minimizing its deformation energy. This energy is defined using the nodal displacement matrix:

$$E_{def} = \frac{1}{2} U^t F$$

U: nodal displacement vector, F: vector of nodal forces So, after specifying the reticulated structure we need to calculate the deformation energy. We start by calculating the nodal displacements. For this, we use an equation defining a relation between stiffness matrix, vector of nodal displacements and vector of nodal forces:

$$KU = F \Longrightarrow U = K^{-1} F$$

s.t K is a square matrix of size $2 * n$ and n is the number of structure nodes. U and F are vectors of size $2 * n$ s.t U is the vector of nodal displacements and F is vector of nodal forces.

0.1.3 Formal problem definition 1: truss structure optimization using analytic fitness function

In this section we obtain the optimization problem we are interested in. We assume elastic linear behavior for the truss bars. In this section, for the sake of simplicity, we will assume that our truss is made of bars of the same material with constant Young modulus E. If we collect together the cross section areas of all the bars to be optimized in the truss in a vector $= (x_1, x_2, \ldots, x_n) \in R^n$, (we will use boldface characters for vector and matrices) being n the number of such bars, then our optimization problem has the form.

$$
\begin{array}{ll}
\underset{x \in U_{ad}}{\text{Minimize}} & C(x) = \dfrac{1}{2} F^t U \\
\text{s.t} & K(x)U = F, (C1). \\
& \displaystyle\sum_{j=1}^{n} l_j x_j \leq V_{max}, (C2)
\end{array}
$$

The feasible set for the optimization problem, U_{ad}, is given by

$$U_{ad} = \{x \in R^n : x_j^{min} \leq x_j \leq x_j^{max}, j = 1, \dots, n\}$$

Where x_j^{min} , x_j^{max} are respectively the maximal and minimal cross section areas allowed for the j-th bar. We call m the number of free nodes of the truss, that is to say, nodes whose displacements is not constrained by the boundary conditions. Then $U \in R^{2m}$ stands for the displacements vector and $F \in R^{2m}$ stands for the vector of loads applied on the nodes of the structure. Recall that our model is two-dimensional, thus if the number of free nodes is m then the components of the displacements and load vector have to be 2m. The cost functional $C(x) = F^t U$, is the compliance, or the work made by the load to deform the structure. Minimizing the compliance is equivalent to maximizing the stiffness of the truss, since the smallest the compliance is, the stiffest the truss is. The equilibrium equation of the system is given by the linear system of equations (C1), where K(x) stands for the stiffness matrix of the structure. Finally, in the volume constraint (C2), l_j stands for the length of the j-th bar, so that its volume is $l_j x_j$, and the total truss volume is $\sum_{j=1}^{n} l_j x_j$, V_{max} is the maximal volume allowed for the structure.

In practice x_j^{min} is a very small but strictly positive value, so that we assume that if this value is reached in the optimized design, then j-th bar is removed from the truss. This makes perfect mechanical sense since the stiffness contribution of such a bar to the truss will be non significant. We do not consider $x_j^{min} = 0$ because the stiffness matrix K(x) might become singular during the optimization process. Practically, to solve the problem with the minimum computation power, we assume that the global stiffness matrix is positive definitive for any admissible design x. Then, for any $x \in U_{ad}$ there exists a unique solution of the system:

$$K(x)U = F \Longrightarrow U(x) = K(x)^{-1} F$$

Which we denote U(x). Then, the nested formulation of the problem is:

$$
\begin{array}{ll}
\underset{x \in U_{ad}}{\text{Minimize}} & C(x) = \dfrac{1}{2} F^t U \\
\text{s.t} & \displaystyle\sum_{j=1}^{n} l_j x_j \leq V_{max}, (C2)
\end{array}
$$

This is a non-linear programming problem. The constraints now are just the box constraints (those defining the admissible set) and a linear constraint (the volume constraint).

0.1.4 Formal problem definition 2: truss structure optimization using approximation of the fitness function

In the second problem instead of minimizing the analytic formula of compliance we will try to approximate the compliance function, using a certain amount of data points evaluated using the compliance function, with a deep neural network using the mean squared error as the loss function. So the second problem is a learning problem:

The learner's input: the learner has access to the following:

Domain set (X): a set of vectors containing the cross sections of bars.

Label set (Y): a set of numbers given by evaluating C using the cross sections of the bar.

Training data (S) : S = $((x_1, y_1)...(x_m, y_m))$ is a finite sequence of pairs in $X * Y$: that is, a sequence of labeled domain points.

Loss function: The measure of success in our situation, since we have a regression problem:

$$y = C(x) = ANN(x) + \epsilon, (P2.1)$$

Is the mean squared error $E = \dfrac{1}{m}\sum_{i=1}^{m}(ANN(x_i) - y_i)^2$. So we need to minimize E to get a good approximation on a training data set and also on a validation dataset in order to avoid overfitting. After training the neural network we will use as the compliance function that we need to minimize. ANN(x) , (P2.2)

0.2 Genetic Algorithm

In general, a genetic algorithm is composed out of: In general, a genetic algorithm is composed out of:

- Representation (Definition of Individuals): this part deals with the
- Evaluation Function (Fitness Function)
- Population:
- Parent Selection Mechanism:
- Variation Operators (Mutation and Recombination):
- Survivor Selection Mechanism (Replacement):
- Initialization:
- Termination Condition:

Each one of these components can be implemented in many ways. In the context of real-valued genetic algorithms these components are presented as follows:

Each one of these components can be implemented in many ways. In the context of real-valued genetic algorithms these components are presented as follows:

Representation: The individuals in the context of Real-valued Genetic Algorithms need to be presented by a vector $(x_1, \ldots, x_n), x_k \in R$.

Evaluation function: The evaluation function in our (P1) problem is going to be the deformation energy. While in the (P2.2) problem it's going to be presented by the approximating ANN.

Population: The population is presented by a set of vectors of n dimensions that represent the n transversal sections of the bars.

Parent selection mechanism: The parents are selected randomly according to the uniform distribution.

Crossover: After generating a number m of parents randomly, crossover will be applied randomly to a nc/2 individuals by selecting two random parents x_1 and x_2 and generate new offspring as follow:

$$y_1 = alpha. * x_1 + (1 - \alpha). * x_2$$

$$y_2 = alpha. * x_2 + (1 - \alpha). * x_1$$

And α is a random vector. This type of crossover is called "simulated binary crossover".

Mutation: Mutation is performed by taking a random individual and changing a number (related to mutation rate) of randomly selected elements in it as follow.

Survivor Selection Mechanism: After generating a crossed population and mutated population, we combine them with the original population in one big population, sort them using the evaluation function from the individual with the smallest evaluation to the one with the biggest, and take the first m individuals from the population, to process them in the same way in the next iteration.

Termination Condition: The algorithm is terminated after a number of iterations.

0.3 Case Study: Deformation Energy Minimization

0.3.1 Annex 1: Dataset

0.3.2 Annex 2: Approximation Of Fitness function Using ANN

0.3.3 Annex 3: Genetic Algorithm

0.3.4 Annex 4: Numerical Results (GA with function approximation)