Tutorial: How to Simulate a Truss Optimization Problem in Improved Fully Stressed Design Evolution Strategy (FSD-ES II)

By: Ali Ahrari (Updated 3/3/2016)

aliahrari1983@gmail.com

Abstract This tutorial explains step by step simulation of an arbitrary truss optimization problem in the improved fully stressed design evolution strategy (FSD-ES II). The limitations of the method are also mentioned.

1. Introduction

This tutorial explains how to use the improved fully stressed design evolution strategy (FSD-ES II) [1] for an arbitrary test problem. The code is provided in the MATLAB© environment. Minor revisions were performed on the code for improving readability, which should not result in a considerable change in the results. This code can be used only for educational and scientific research purposes provided that the use of this code is acknowledged and the corresponding publication ([1]) is cited. Researchers are also recommended to review previous publications on FSD-ES [2, 3]. For commercial use, please contact Ali Ahrari at aliahrari1983@gmail.com. Your comments, feedback and recommendations are highly welcomed.

2. Extent of the Code

2.1. Constraints

FSD-ES can handle two types for design specifications and therefore two templates are provided which should be used depending on the constraints in the problem:

- **Simplified specifications**: Constraints are maximum axials stress, Euler buckling and maximum displacement. Sections can be discrete or continuous.
- **AISC-ASD specifications:** The stress and slenderness constraint are based on AISC-ASD (9^{th} edition) design specifications. Displacement constraints are also accepted.

As a reference, data for two test problems (110-bar transmission tower [1] and 224-bar pyramid [4]) are provided. To optimize these problems, simply run *bar110SIMP.m* and *bar224ASD.m*, respectively. If the simplified specifications govern your problem, use *bar110SIMP.m* and replace the 110-bar data by your problem data. If the AISC-ASD specifications govern your problem, use *bar224ASD.m* and replace the 224-bar data by your problem data. The steps to enter the simulation data of your problem are explained.

2.2. Objective Function

The objective function is simply the weight of the structure.

3. Control Parameters of the Algorithm

FSD-ES II does not need ad-hoc parameter tuning. Modification of the default values of the control parameters is not recommended. These parameters are provided in the cell named 'opt'. The population size and the maximum number of iterations can be controlled by *opt.lambda_coeff* and *opt.maxiter_coeff*, respectively.

4. Truss Simulation Data

For each problem, the following data should be provided.

4.1. Specify seed number, truss type, number of members and nodes

Parameter *RandSeedNo* specifies the seed number for the random number generator. Identical values result in identical optimization process.

Parameter *D* specifics whether truss is planar or spatial. *D*=2 and *D*=3 should be used for planar and spatial structures. *N_node* and *N_member* are the number of nodes (N_n) and members (N_m) in the truss, respectively.

4.2. Specify the connectivity plot in the ground structure

GSCP determines the connectivity plot in the ground structure, which is an $N_m \times 2$ matrix. The *i*-th row of this matrix specifies the end nodes of the *i*-th member.

4.3. Specify topologically grouped members

In many truss problems, some members are coupled to some other members. Matrix **sym_M** specifies topologically dependent members. Each column of this matrix refers to a group of members that must be active or passive altogether. If the groups differ in the number of members, repeat the last member of the group. For example, assume that there are two groups of member which must have identical values for topology variables. The first group is $\{1, 2, 4\}$ and the second group is $\{3, 5\}$. To provide this information, set:

$$\textbf{sym\_M} = \begin{bmatrix} 1 & 3 \\ 2 & 5 \\ 4 & 5 \end{bmatrix}$$

This means only members 1 and 3 have independent topology variables.

- Do not include members in **sym_M** which do not belong to any group.
- If there is no topology grouping in the problem, then set: $\textbf{sym\_M} = \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}$

4.4. Specify members that must have similar cross sections

Matrix **sym_A** specifies the size variables that are coupled. In most benchmark problems, topologically grouped members must have identical cross sections. If it is the case, simply let: **sym_A**=**sym_M**, otherwise define **sym_A** in a similar style. Each column of **sym_A** refers to a group of members that must have similar cross sections. If the groups differ in the number of members, repeat the last member of the group. If there is no cross section grouping in the problem, then set:

$$\textbf{sym\_A} = \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}$$

Limitations:

- All members whose cross sections are coupled together must be topologically coupled as well (So far we haven’t encountered any test problems in which this condition is violated)
- If AISC-ASD specifications are used, coupled members must have identical lengths.

4.5. Specify basic nodes and members

Basic nodes and basic members are determined by ***basic_node*** and ***basic_member***. For example, if nodes 1, 3 ,8 are basic and members 2, 3, 6 are basic, then let:

basic_node=[1,3,8]

basic_member=[2, 3, 6]

4.6. Specify dependent/independent coordinates

In FSD-ES II, coordinates are stored as follows:

- For a planar truss: $\boldsymbol{X}$=[x_1, y_1, x_2, y_2, …, x_{Nn}, y_{Nn}]
- For a spatial truss: $\boldsymbol{X}$=[x_1, y_1, z_1, x_2, y_2, z_2 …, x_{Nn}, y_{Nn}, z_{Nn}]

X_indep_ind is a vector which specifies the independent coordinate variables. For example, if x_2 and y_3 of a planar truss are independent, then: ***X_indep_ind***=[3, 6].

X_const_ind is vector which specifies the fixed coordinates. The value of these parameters are entered in ***X_const_val***. For example, assume a spatial truss design problem which requires that x_2=100, y_3=0 y_4=0. For this problem, let:

X_const_ind=[4, 8, 11]

X_const_val=[100, 0 ,0]

4.7. Specify anchored degrees of freedom (DOF) of the truss

Each structure is anchored at some nodes. ***DOF_constrained*** is a vector which specifies the degree of freedom that are anchored by the support. For example, if the truss is planar and nodes 1 is fully anchored and nodes 2 and 4 are constrained such that they may not have a displacement along the *y* axis, then: ***DOF_constrained***=[1 2 4 8]

4.8. Specify loading

Fext specifies the applied load(s) on the structure. It is a $DN_n \times N_l$ matrix, where N_l is the number of load cases applied to the structure. Each column refers to one load case.

4.9. Specify mechanical properties.

Enter mechanical properties:

> *M_elasticity*: Modulus of elasticity of the material
>
> *Density*: Density of the truss material
>
> *dis_all*: Allowable displacement along each direction. If no displacement constraint, then set *dis_all*=1e12;

If you are using *template_ASD*, also specify:

> *Fy*: Yield strength according to AISC-ASD 9th edition

If you are using *template_SIMP*, also specify:

> *SigT_all*: tensile strength
>
> *SigC_all*: compressive strength

kappa: The parameter in the elastic buckling equation. $\sigma_c \leq \kappa \frac{EA}{l^2}$. Usually $\kappa \cong 4$.

4.10. Specify the given section list

Specify the available section list number for this problem. Use *section_no*=0 if the sections are continuous and specify *Min_A* and *Max_A* (the lower and upper limit of section areas), otherwise:

- If simplified specifications govern the problem, use a section list number between 1 and 99.
- If AISC-ASD specifications are used, use a number equal to or greater than 101.

Data of the available sections must be provided in the file *sections.m*. function *sections* returns:

- A vertical vector specifying available discrete sections, if simplified specifications are used.
- A matrix with two columns if AISC-ASD specifications govern the problem. In this case, the first column contains the section areas while the second column contains the corresponding radii of gyration. If two section have identical areas, remove the one with the lower radius of gyration from the list.

4.11. Apply shape symmetry rule

If there are coupled coordinates, their relation should be specified in the file *enforce_shape_sym.m* function *enforce_shape_sym* gets the vector of coordinate variables and update the dependent coordinate variables. Data for different problems can be provided in this file using the switch-case option. Provide the number of members in the problem in front of 'case' and then calculate *newX* (coordinate variables after applying the coupling rules) from *X* (input coordinates before applying the coupling rules). The data for the 110-bar transmission tower and the 224-bar pyramid are already provided.

5. Run the Code

The simulation process is complete now. You can run the file. The code writes the optimization history and the best feasible solution in two csv files, at intervals specified by *opt.WriteInterval*. The best solution will be plotted at the end of the optimization process.

Please contact Ali Ahrari (aliahrari1983@gmail.com) if you encounter an error, or obtained odd results.

References

[1] A. Ahrari and K. Deb, "An improved fully stressed design evolution strategy for layout optimization of truss structures," *Computers & Structures*, vol. 164, pp. 127-144, 2016.

[2] A. Ahrari and A. A. Atai, "Fully stressed design evolution strategy for shape and size optimization of truss structures," *Computers & Structures*, vol. 123, pp. 58-67, 2013.

[3] A. Ahrari, A. A. Atai and K. Deb, "Simultaneous topology, shape and size optimization of truss structures by fully stressed design based on evolution strategy," *Engineering Optimization*, vol. 47, no. 8, pp. 1063-1084, 2015.

[4] O. Hasançebi and F. Erbatur, "Layout optimisation of trusses using simulated annealing," *Advances in Engineering Software*, vol. 33, no. 7, pp. 681-696, 2002.