Combining evolutionary optimization with finite element method for optimizing transmission-line tower

Sz Nagy\(^1\), K Jármai\(^2\), A Baksa\(^3\)

\(^1\) Emerson Automation FCP Kft., Bánki Dónáé út 3, Eger 3300, H
\(^2\) Institute of Energy Engineering and Chemical Machinery, University of Miskolc, Egyetemváros, Miskolc 3515, H
\(^3\) Institute of Applied Mechanics, University of Miskolc, Egyetemváros, Miskolc 3515, H

E-mail: szilard.nagy@emerson.com

Abstract. Metaheuristic evolutionary methods are powerful tools for solving non-linear, many dimensional optimisation problems. In this article, one of these methods is combined with finite element method for optimising truss like structure. The advantages of the combined method are presented through a real engineering task, optimisation of transmission line tower. The design of the latter can take a variety of shapes, so in the present case, only the lower part can be optimised. For optimisation, we use the adaptive differential evolution (SaDE) technique. The objective function of the optimisation is the mass of the structure. Finite element analysis is used to determine the stresses. The overall and local buckling constraints have also been considered.

1. Introduction

Meta heuristic and evolutionary algorithms are powerful tools for solving non-linear, high dimensional optimisation or searching problems. They are population based stochastic techniques. Self-adaptive differential was proposed in [1] as an improvement of original differential evolution [2]. It was used efficiently in many real cases, such as pattern recognition [3], communication [4] and engineering [5].

The usage of closed circular hollow section (CHS) in truss like structures is more preferable then other opened sections [6; 7], like angel section. The buckling stiffness of opened sections is very small [5].

In this paper we combine evolutionary optimisation with finite element method for solving optimization of truss like structures from CHS. Proposed general method is used for optimizing real life structure, bottom part of transmission line tower.

2. Constrained self-adaptive differential evolution

Differential evolution (DE) was introduced in 1997 by Storn [2]. It is a simple but powerful evolutionary optimization algorithm. The basic idea behind the algorithm is to produce new individuals by combining positions of randomly selected individuals from the current population. During the iteration steps, it repeats three operations mutation, crossover, and selection until it finally reaches a stopping condition.
Mutation operation produces \((G)v\) mutant vector for each \((G)x\) individual in \(G^{th}\) generation population:

- **DE/rand/1:**
  \[
  (G)v_i = (G)x_{r_1} + F \left( (G)x_{r_2} - (G)x_{r_3} \right) \tag{1}
  \]

- **DE/best/1:**
  \[
  (G)v_i = (G)x_{\text{best}} + F \left( (G)x_{r_1} - (G)x_{r_2} \right) \tag{2}
  \]

- **DE/current to best/2:**
  \[
  (G)v_i = (G)x_i + F \left( (G)x_{\text{best}} - (G)x_i \right) + F \left( (G)x_{r_1} - (G)x_{r_2} \right) \tag{3}
  \]

- **DE/rand/2:**
  \[
  (G)v_i = (G)x_{r_1} + F \left( (G)x_{r_2} - (G)x_{r_3} \right) + F \left( (G)x_{r_4} - (G)x_{r_5} \right) \tag{4}
  \]

- **DE/best/2:**
  \[
  (G)v_i = (G)x_{\text{best}} + F \left( (G)x_{r_1} - (G)x_{r_2} \right) + F \left( (G)x_{r_3} - (G)x_{r_4} \right) \tag{5}
  \]

where \(i\) is index of \(i^{th}\) individual in population, \(r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5\) are random indices of individuals in population, and finally \(F\) is a scaling factor in \([0, 2)\) range.

After mutation, the crossover operation is applied for every \((G)v_i\) and \((G)x_i\) vector pairs to produce \((G)u_i\) trial vectors

\[
(G)u_{i,j} = \begin{cases} 
(G)v_{i,j} & \text{if } \text{rand} \leq C_R \lor j_{\text{rand}} = j \\
(G)x_{i,j} & \text{otherwise} 
\end{cases}
\tag{6}
\]

where \(j\) is \(j^{th}\) coordinate of \(i^{th}\) individual, \(j_{\text{rand}}\) is random coordinate index, and \(C_R\) is crossover rate.

If the fitness value of newly generated trial vectors better than \((G)x_i\) individual’s own, trial vector will replace the original individual during the selection operation

\[
(G+1)x_i = \begin{cases} 
  u_i & \text{if } F(u_i) \leq F(x_i) \\
  x_i & \text{otherwise} 
\end{cases}
\tag{7}
\]

where \(F()\) is the fitness function to be optimized.

The mutation strategies presented above show different efficiencies in solving different optimization problems. It is not possible to say in advance which method will be the most effective tool for solving a given problem. To achieve a good performance, all five method need to be tried, and \(F, C_R\) critical parameters need to be fine tuned. Self-adaptive differential evolution (SaDE) proposes a learning base adaptive method to solve this time-consuming problem in [1].

SaDE algorithm uses two of the five mutation strategies presented above according to equation (1) and (5). In every iteration step, each strategy is applied randomly. The probability of applying of strategy is depend from \(p\) variable

\[
(G)v_i = \begin{cases} 
  \text{acc. to eq. (1)} & \text{if } \text{rand} \leq p \\
  \text{acc. to eq. (5)} & \text{otherwise} 
\end{cases}
\tag{8}
\]
where \( p \) probability variable is defined by following
\[
p = \frac{n_{s1} (n_{s2} + n_{f2})}{n_{s2} (n_{s1} + n_{f1}) + n_{s1} (n_{s2} + n_{f2})}
\]
where \( n_{s1} \) and \( n_{s2} \) are number of trial vectors successfully entering the next generation produced according to equations (2) or (5), \( n_{f1} \) and \( n_{f2} \) are the number of discarded vectors produced each mutation strategy. The \( p \) probability variable is updated every 50th iteration step and at the same time \( n_{s1}, n_{f1}, n_{s2}, n_{f2} \) will be reset. In every iteration step, \( F \) scaling factor is get a random value in \([0, 2]\) range with normal distribution of mean 0.5 and standard deviation 0.3 and it is different for individuals. After every 5th iteration step, \( C_R \) crossover rate randomly change in \([0, 1]\) range with gaussian distribution of mean \( C_{Rm} \) and standard deviation 0.1. \( C_{Rm} \) is the average of successfully applied (individual is entering the next generation) \( C_R \) values during the learning period, what consists of 25 iteration steps.

Self-adaptive differential evolution is developed for solving continuous optimization problems as a most of evolutionary algorithms. In real life, most of engineering optimization problems are constrained. It must meet mechanical and economical conditions, for example static stress, local buckling, deflection and fatigue. Constrained optimization problem could be written in following general form
\[
\begin{align*}
\min & \quad f(\mathbf{x}) \\
\text{subject to} & \quad g_i(\mathbf{x}) \leq 1 \\
& \quad 1 \leq i \leq r \\
& \quad h_j(\mathbf{x}) = 0 \\
& \quad 1 \leq j \leq q
\end{align*}
\]
where \( \mathbf{x} \) is vector of design variables in \( n \) dimensional \( S \) searching space, \( g_i(\mathbf{x}) \) are \( r \) count inequality constraints, \( h_j(\mathbf{x}) \) are \( q \) count equality constraints.

Usage of penalty function is an effective and simple constraint handling method [8]. These methods convert constrained problems to unconstrained problems.
\[
\mathcal{F}(\mathbf{x}) = f(\mathbf{x}) + \sum_{i=1}^{r} G_i(\mathbf{x}) + \sum_{j=1}^{q} H_j(\mathbf{x})
\]
\[
G_i(\mathbf{x}) = \begin{cases} 
0 & \text{if } g_i(\mathbf{x}) \leq 1 \\
(R_i g_i(\mathbf{x}))^\beta & \text{otherwise} 
\end{cases}
H_j(\mathbf{x}) = \begin{cases} 
0 & \text{if } |h_j(\mathbf{x})| \leq \epsilon \\
(Q_j |h_j(\mathbf{x})|)^\gamma & \text{otherwise}
\end{cases}
\]
where \( G_i(\mathbf{x}) \) and \( H_j(\mathbf{x}) \) are static penalty functions [9; 10], \( R_i \) and \( Q_j \) are quite large constants, \( \beta \) and \( \gamma \) are typically 1 or 2 [8].

3. Produce constraints with finite element method
Connections of circular hollow section (CHS) truss structures are frequently modelled as pin connection for elastic analysis [11]. From this property, CHS structures could be modelled with pushed–pulled elements for finite element analysis [12–14], see in figure 1.

Node displacement is only possible along the \( \xi \) axis (local coordinate system) through the nodes \( i \) and \( j \). In the \( x-y \) global coordinate system of whole structure, the projections \( x, y \) of this displacement are interpreted.
\[
\mathbf{e} \mathbf{u}' = \left[ \mathbf{e} u'_i \quad \mathbf{e} u'_j \right]^T \\
\mathbf{e} \mathbf{u} = \left[ \mathbf{e} u_{ix} \quad \mathbf{e} u_{iy} \quad \mathbf{e} u_{jx} \quad \mathbf{e} u_{jj} \right]^T
\]
The connection between the two coordinate systems is the transformation matrix
\[
\mathbf{e} \mathbf{T} = \begin{bmatrix} 
\mathbf{e} T_{11} & \mathbf{e} T_{12} & 0 & 0 \\
0 & \mathbf{e} T_{23} & \mathbf{e} T_{24}
\end{bmatrix}
\]
\begin{align}
\epsilon T_{11} &= \epsilon T_{23} = \frac{\epsilon u_{jx} - \epsilon u_{ix}}{\epsilon L} \\
\epsilon T_{12} &= \epsilon T_{24} = \frac{\epsilon u_{jy} - \epsilon u_{iy}}{\epsilon L} \\
\epsilon u' &= \epsilon T \cdot \epsilon u
\end{align}

where \( \epsilon L \) is length of \( \epsilon \)th truss element.

In equilibrium, the \( \Pi \) potential energy of the structure is minimal. This means that the \( \delta \Pi \) first variation of the potential energy is zero. The total potential energy of the elements in local and global coordinate system

\begin{align}
\epsilon \Pi' &= \frac{1}{2} \epsilon u^T \left( \epsilon K' \cdot \epsilon u' - \epsilon f' \right) \\
\epsilon \Pi &= \frac{1}{2} \epsilon u^T \left( \epsilon K \cdot \epsilon u - \epsilon f \right)
\end{align}

where \( \epsilon f' \) and \( \epsilon f' \) are generalized load vectors reduced to nodes in corresponding coordinate system, \( \epsilon K' \) and \( \epsilon K \) are stiffness matrices

\begin{align}
\epsilon K' &= \frac{\epsilon E \cdot \epsilon A}{\epsilon L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \\
\epsilon K &= \epsilon T^T \cdot \epsilon K \cdot \epsilon T
\end{align}

Introducing \( u \) vector of nodal displacements and \( f \) vector of loads

\begin{align}
u = \begin{bmatrix} \epsilon u_1 \\ 2 \epsilon u_2 \\ 3 \epsilon u_3 \\ \vdots \\ \epsilon u_n \end{bmatrix}^T \\
f = \begin{bmatrix} \epsilon f_1 \\ 2 \epsilon f_2 \\ 3 \epsilon f_3 \\ \vdots \\ \epsilon f_m \end{bmatrix}^T
\end{align}

the first variation of potential energy of total structure can be written in the form below

\begin{align}
\delta \Pi = 0 = \frac{1}{2} u^T (K u - f)
\end{align}

where \( K \) is stiffness matrix of whole structure according to element connection rules [12]. After applying boundary conditions [12; 13] (fixations, required nodal displacements, etc.) the non-trivial solution of (22) is

\begin{align}
K u = f
\end{align}
For further calculations, it is not enough to know nodal displacements, stress of elements also should be known

\[
e_\sigma = e_\frac{E}{L} \cdot \begin{bmatrix}
e T_{11} & -e T_{12} & e T_{11} & e T_{12}
\end{bmatrix} e u
\]  

(24)

First proposed inequality constraint describes resistance of elements to static load

\[
g_{le} = \begin{cases} \frac{\gamma_{M0} e_\sigma}{e \chi f_y} & \text{if } e_\sigma < 0 \\ \frac{\gamma_{M0} e_\sigma}{f_y} & \text{otherwise} \end{cases}
\]

(25)

where \( e \) means the element under inspection, \( e \chi \) is buckling factor according to [15] by elements, \( \gamma_{M0} \) is generalized safety factor according to [15] and \( f_y \) is yield stress.

Equation (26) is true if only if when unit of \( f_y \) is MPa, and unit of \( D_i \) and \( t_i \) are millimetres.

CHS cross section is a thin walled section (see figure 2), which need to be examined for local buckling. The truss like structure may consist of one or more CHS cross sections. Each cross-section of different diameters and wall thicknesses should be inspected for local buckling.

\[
g_{IIi} = \frac{D_i f_y}{21150 t_i}
\]

(26)

4. Example truss like structure

Transmission-line towers could be divided into two parts, conical bottom part (see figure 3), and various shaped upper part. In this paper, we have optimized a lower part of an intermediate pylon.

The applied loads were according to [5; 16]. Without detailing of calculations, the size giving load is half wire pulling according to [16]. Forces transmitted from upper part are \( F_V = 209.03 \text{kN} \) vertical force, \( F_H = 312.14 \text{kN} \) horizontal force and \( M_h = 2850.5 \text{kNm} \) bending moment. Forces reduced to nodes with notation of figure 3

\[
F_{y1} = \frac{M_h}{2 a_2} - \frac{F_V}{4} = \frac{2850.5 \text{kNm}}{2 \cdot 3.7 \text{m}} - \frac{209.03 \text{kN}}{4} = 332.94 \text{kN}
\]

(27)

\[
F_{y1} = \frac{M_h}{2 a_2} + \frac{F_V}{4} = \frac{2850.5 \text{kNm}}{2 \cdot 3.7 \text{m}} + \frac{209.03 \text{kN}}{4} = 437.46 \text{kN}
\]

(28)

where \( a_2 = 3.7 \text{m} \) is the width of upper pylon part. It is sufficient to perform the calculations on only one inclined plane relevant to the loads as shown in figure 4. In the case of a pyramid with a side skew of \( \beta = 80^\circ \), the forces acting on the plane

\[
F_1 = \frac{F_{y1}}{\sin \beta} = \frac{332.94 \text{kN}}{\sin 80^\circ} = 338.08 \text{kN}
\]

(29)
Figure 3. Simplified illustration of lower part of transmission line tower.

\[ F_2 = \frac{F_{y2}}{\sin \beta} = \frac{437,46 \text{kN}}{\sin 80^\circ} = 444,21 \text{kN} \]  
\[ F_3 = \frac{F_H}{2} = \frac{312,14 \text{kN}}{2} = 156,07 \text{kN} \]  

Trusses of structure were divided into three cross-section group. The first group is elements of side column (1–10 elements). Trusses in second group are horizontal bars (17–22 elements).

Figure 4. Planar truss like structure.
The final group consists of grid elements forming a deltoid and bottom triangle (23–26). In each cross-section group, CHS is used. The characteristic dimensions according to figure 2 were outer diameters \(D_1, D_2, D_3\), and wall thicknesses \(t_1, t_2, t_3\).

Weight minimum was the target function of optimisation

\[
f(x) = \sum_{e=1}^{n_e} \frac{\pi \rho A_e^2}{4}
\]

where \(\rho\) is density of steel, \(A_e\) is cross-sectional area of \(e^{th}\) elements, \(n_e\) is number of elements, \(x = [D_1, D_2, D_3, t_1, t_2, t_3]^T\) is vector of unknowns. According to (11), (25), (26) and (32) the final fitness function for SaDE algorithm

\[
\mathcal{F}(x) = f(x) + \sum_{e=1}^{n_e} G_e(g_{Ie}(x)) + \sum_{i=1}^{3} G_i(g_{IIi}(x))
\]

\(R_e = R_i = 10^6\) penalty constants was used for \(G_e(x)\) and \(G_i(x)\) penalty functions.

5. Result of optimisation

Several different topologies with different number of grids were examined. Grid division was changed between 1–6. Also yield stress of raw material steel was changed between \(f_y = 235\) MPa and \(f_y = 690\) MPa. You can see results in figure 5 and figure 6.

![Figure 5. Optimized weight with different grid division and raw material](image)

![Figure 6. Percentage change in optimised weight](image)

Minimal weight could be found when number of grid division was 5 and \(f_y\) yield stress was 460 MPa. As you can see the optimized weight decreases if count of grid division increase. Usage of higher strength steel does not result lighter structure. This is because a thicker bar is required for greater yield strength due to buckling.

6. Conclusion

We have combined an evolutionary optimisation algorithm – such as self-adaptive differential evolution – with finite element method for optimizing truss like structures from circular hollow
sections. The proposed method could be used for fully automatic optimization, when finite element model of structure is known.

We have optimised a bottom part of an intermediate transmission-line tower with this method. The optimized weight not necessary decrease if higher strength steel is used. On one hand higher yield stress increases allowable static stress, but on the other hand it is harmful for $\chi$ buckling factor.

References
[1] Qin A and Suganthan P 2005 Self-adaptive differential evolution algorithm for numerical optimization IEEE Congress on Evolutionary Computation vol 2 pp 1785–1791
[2] Storn R and Price K 1997 Journal of Global Optimization 11 341–359
[3] Ilonen J, Kamarainen J K and Lampinen J 2003 Neural Processing Letters 17 93–105
[4] Nurhan K 2005 Journal on Applied Signal Processing 8 1269–1276
[5] Károly J and József F Innovative design of steel structures (in Hungarian) (Miskolc: Gazdássz Elasztik Kiadó és Nyomda) ISBN 978-963-358-064-6
[6] Rao G 1995 Computers & Structures 57 81–92
[7] da Silva J, Vellasco P d S, de Andrade S and de Oliveira M 2005 Journal of Constructional Steel Research 61 1108–1134
[8] Yeniay O 2005 Mathematical and Computational Applications 10 45–56 ISSN 2297-8747
[9] Homaifar A, Qi C X and Lai S H 1994 Simulation 62 242–254
[10] Morales A K and Quezada C V 1998 A universal eclectic genetic algorithm for constrained optimization Proceedings 6th European Congress on Intelligent Techniques and Soft Computing (EUFIT) vol 2 (Verlag) pp 518–522
[11] Wardenier J, Kurobane Y, Packer J A, van der Vegte A and Zhao X L 2008 Design guide for circular hollow section (CHS) joints under predominantly static loading (Zürich: CIDECT)
[12] István P 1999 Finite element methods in engineering practice (in Hungarian) (Miskolc-Egyetemváros: Miskolci Egyetemi Kiadó) ISBN 963-661-312-5
[13] António J M F MATLAB Codes for Finite Element Analysis (Berlin: Springer-Verlag) ISBN 978-94-007-8955-5
[14] Ian M S and Lee M Programming the Finite Element Method, 4th edition (West Sussex: John Wiley and Sons Ltd.) ISBN 0-470-84969-3
[15] EN 1993-1-1:2009 Eurocode 3: Design of steel structures - Part 1-1: General rules and rules for buildings
[16] MSZ 151-1:2000 Power overhead lines. Installation requirements for overhead lines with nominal voltage greater than 1 kV

Acknowledgments
The research was supported by the Hungarian National Research, Development and Innovation Office under the project number K 134358, and by the NTP-SZKOLL-20-0022 identifier ”Focus’21-Focus on community by developing digital competencies” project, supported by the Ministry of Human Resources and Human Resources Support Manager.