

Applying the differential evolution algorithm to the optimisation of cross sections of steel columns

J. Kilkki¹, J. Lampinen² & H. Martikka³

^{1,3}*Department of Mechanical Engineering*

²*Department of Information Technology*

Lappeenranta University of Technology, Finland

Abstract

The suitability of the evolution algorithm for mixed number optimisation is presented. The cross section of a welded or edged steel column is optimised using the total mass as the object function. The optimisation is repeated using different effective lengths of the column. The thickness of the material is chosen from the commonly available plate thicknesses. The widths and heights of the plate fields are used as continuous optimisation parameters. The discrete optimisation parameters are the thickness of the steel plates, the number of plate fields, and material properties. The selection of the population size, the mutation probability and the value of the differential factor have a strong effect on both the ability of the algorithm to achieve the global optimum and on the solution time. An attempt is made to automate the selecting of these parameters of the algorithm. It is important to find a reliable automated method for setting the parameters of the algorithm in the practical task of optimising steel structures. Optimal cross sections of the columns are presented, together with the effect of the parameters of the optimisation algorithm. The suitability of the differential evolution algorithm for mixed number optimisation in the current case is evaluated. Automating the selection of the parameters of the algorithm is also evaluated and discussed critically.

Introduction

An application of a new optimisation program is presented in this paper. The model of the optimisation problem uses mathematical and table components. The optimisation solver is based on the Differential Evolution (DE) algorithm.

A double symmetric uniform loaded I-beam is used as an example.

Possible cross sections are classes 1, 2 and 3, in which the yield strength of the material is reached before any local buckling occurs. Cross section class 4 is not considered. The cross section class defines which dimensioning formulas are used. The strength and deflection of the beam are considered as restraints. The total mass and deflection of the beam are used as objective functions.

The dimensioning method used is limit state dimensioning. This method is suitable for the optimisation of steel structures.

Optimisation

The aim of structural optimisation is the minimisation or maximisation of an objective function e.g. cost of materials and labour, structural weight and storage capacity. The problems of structural optimisation may be classified as: 1) sizing or optimisation of the cross-sectional dimensions of one- or two-dimensional structures, for which the cross-sectional geometry is partially prescribed, so that the cross-section can be fully described by a finite number of variables; 2) shape optimisation (e.g. the shape of the centroidal axis of bars and the middle surface shells, or interfaces between different materials in composites); 3) layout optimisation, which consists of three simultaneous operations: a) topological optimisation (spatial sequence or configuration of members and joints), b) geometrical optimisation (location of joints and shape of member axes), and c) optimisation of the cross-sections.[1]

The optimisation procedure of an objective function fulfilling the inequality and equality constraints as follows is searched for:

$$\left. \begin{array}{ll} \min f(\mathbf{x}) & \mathbf{x} = (x_1, \dots, x_i, \dots, x_n)^T \\ g_j(\mathbf{x}) \geq 0 & j = 1, 2, \dots, p \\ h_j(\mathbf{x}) = 0 & j = p+1, \dots, q \end{array} \right\} \quad (1)$$

where n is the number of unknowns and p is the number of constraints. The functions may be continuous or the unknowns may be defined by a series of discrete values. Typically it is required that the variables are positive ($x_i \geq 0$), or their upper and lower limits may be prescribed with box limits $x_i^l \leq x_i \leq x_i^u$. The functions f, g, h may be linear or non-linear. In structural synthesis problems it is characteristic that the number of constraints is larger than the number of variables ($p \geq n$).

Optimisation program

A program called Optimaze has been developed in the Department of Mechanical Engineering at Lappeenranta University of Technology. The program is based on calculation components, and it speeds up the development of the optimisation model. The whole optimisation model is formed using these components. The components are linked together with special relations. The components can be mathematical or table components or other programs like finite element pre- and post processors and solvers. The math component contains objective functions and restraints. The table components contain discrete data like the thickness of the steel plate. The finite element component contains a special interface to the finite element program. The solver of the program is based on the Differential Evolution algorithm.

Differential Evolution algorithm

Storn and Price introduced first the Differential Evolution algorithm (DE) [2]. It can be categorised into a class of evolutionary optimisation algorithms (EA).

DE maintains a population of alternative solutions for the optimisation problem to be solved. Alternative solutions are individuals of the population. The next population will be created using a specific reproduction scheme.

Input:	$D, G_{\max}, NP \geq 4, F \in (0,1+), CR \in [0,1], \text{initials: } \mathbf{x}^l, \mathbf{x}^h.$
Initialise:	$\begin{cases} \forall i \leq NP \wedge \forall j \leq D: x_{j,i,G=0} = x_j^l + rand_j[0,1] \cdot (\mathbf{x}_j^h - \mathbf{x}_j^l) \\ i = \{1,2,\dots, NP\}, j = \{1,2,\dots, D\}, G = 0, rand_j[0,1] \in [0,1] \end{cases}$
While :	$G < G_{\max} \quad \forall i \leq NP$
	Cross over and recombination:
	$r_1, r_2, r_3 \in \{1,2,\dots, NP\}, \text{ randomly selected, except: } r_1 \neq r_2 \neq r_3 \neq i$ $j_{rand} \in \{1,2,\dots, D\}, \text{ randomly selected for each}$ $\forall j \leq D, u_{j,i,G+1} =$ $\begin{cases} x_{j,i,G} + F \cdot (x_{j,r_1,G} - x_{j,r_2,G}) & \text{if } (rand_j[0,1] < CR \vee j = j_{rand}) \\ x_{j,i,G} & \text{otherwise} \end{cases}$
	Selection:
	$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G+1} & \text{if } f(\mathbf{u}_{i,G+1}) \leq f(\mathbf{x}_{i,G}) \\ \mathbf{x}_{i,G} & \text{otherwise} \end{cases}$
	$G = G + 1$

Algorithm 1: Differential evolution algorithm [3].

Population contains n_p solution vectors called individuals of population. Each solution vector or individual represents a possible solution for the optimisation problem. Parameters are chromosomes of the individual. The genetic algorithm

90 Computer Aided Optimum Design of Structures VII

works with binary codes of the chromosomes but the DE algorithm works with parameters. An individual of the next generation is created by adding a weighted difference of the corresponding components of two other individuals or parameter vectors.

A selection scheme is elite because all individuals in the next generation $k+1$ are equal or better than counterparts in the current k population. [2],[11]

The boundary points satisfy the equation $g_j(\mathbf{x}) = 0$. The interior points satisfy the equation $g_j(\mathbf{x}) < 0$. The set of all feasible designs forms the feasible region. The exterior points satisfy the equation $g_j(\mathbf{x}) > 0$. The set of all points which satisfy the constraints constitutes the feasible domain of $f(\mathbf{x})$.

The most common approach in the GA community to handle constraints (particularly inequality constraints) is to use penalties. The basic approach is to define the fitness value of an individual i by extending the domain of the objective function f using

$$fitness = f_i(\mathbf{X}) \pm \mathbf{Q}_i \quad (2)$$

where $\mathbf{Q}_i$ represents either a penalty for an infeasible individual i or a cost for repairing such an individual (i.e., the cost of making it feasible). It is assumed that if i is feasible then $\mathbf{Q}_i = 0$. There are at least three main choices to define a relationship between an infeasible individual and the feasible region of the search space: 1) an individual might be penalised just for being infeasible (i.e., no information is used about how close it is to the feasible region); 2) the amount of its infeasibility can be measured and used to determine its corresponding penalty; or 3) the effort of repairing the individual (i.e., the cost of making it feasible) might be taken into account.

The following guidelines were derived from the design of the penalty functions: 1) Penalties which are functions of the distance from feasibility are better performers than those which are merely functions of the number of violated constraints. 2) For a problem having few constraints and few full solutions, penalties which are solely functions of the number of violated constraints are not likely to find solutions. 3) Good penalty functions can be constructed from two quantities: the maximum completion cost and the expected completion cost. The completion cost is the cost of making an infeasible solution feasible. 4) Penalties should be close to the expected completion cost, but should not frequently fall below it. The more accurate the penalty, the better will be the solution found. When a penalty often underestimates the completion cost, then the search may not find a solution.

Penalties change over time in the dynamic penalties technique. Individuals are evaluated using:

$$fitness_i(\mathbf{X}) = f_i(\mathbf{X}) + (\mathbf{C} \times t)^\alpha \sum_{j=1}^m f_j^\beta(\mathbf{X}) \quad (3)$$

where $\mathbf{C}$, α and β are constants defined by the user and m is the number of constraints. This dynamic function increases the penalty with progress through generations. The quality of the solution found is very sensitive to changes in the values of the parameters [3].

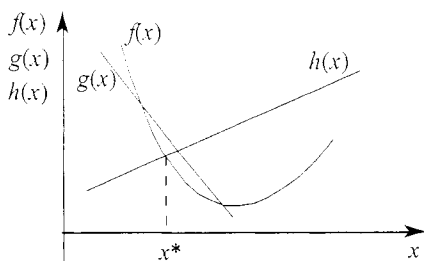


Figure 1: Restraints and an objective function.

Restraints are included to the objective function with simple penalty function.

$$f(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \text{if } \mathbf{x} \text{ is on feasible area} \\ f(\mathbf{x}) + \rho \cdot g(\mathbf{x}) & \text{otherwise} \end{cases} \quad (4)$$

Coello has presented how penalty functions can be treated as objective functions [4]. The difference $f - f^* \leq \epsilon$ is not a usable optimality criteria in probabilistic optimisation methods. There is never a guarantee that ϵ -optimality is achieved with any given ϵ . The stopping condition is based on the assumption that the probability to raise ϵ -optimality cannot be raised without an undue amount of additional effort. [5]

Structure

The structure used as an example is the cross section of a symmetric I-beam, presented in figure 2. The loading is uniform loading. Factor ψ represents the stress distribution on a cross section. In the case of bending of a symmetric cross section the value of $\psi = -1$. The nominal yield strength of the material is 355. The thicknesses are from the catalogue of the manufacturer (Rautaruukki steel products, Designer's guide, 1996) [6]. Dimensioning formulas are written according to Finnish code of building regulations. [7]. The analytical optimisation of the selected structure has been presented by Farkas and Niemi. [8],[9].

Constants: beam length $l = 30\,000$ mm, stress distribution factor $\psi = -1.0$, elasticity factor $E = 210\,000$ MPa, loading density $q = 38$ N/mm, density of steel $\rho = 7850 \cdot 10^{-9}$ kg/mm³, deflection limit $k_{\text{vali}} = 1/300$, normal force $N = 0.0$ N, acceleration $g = 9.81$ m/s².

Optimisation parameters: web height $h_w = [100\,10000]$ mm, flange width $b = [100\,10000]$ mm, web thickness $t_w = \{2, 3, 4, 5, 6, 8, 10, 12, 14, 16, 18, 20, 22, 25, 30, 35, 40, 45, 50\}$ mm, flange thickness $t_f = \{2, 3, 4, 5, 6, 8, 10, 12, 14, 16, 18, 20, 22, 25, 30, 35, 40, 45, 50\}$ mm.

The material strength depends on the thickness as follows: $f_y = 355$ MPa, when $t = [2\,16]$ mm, $f_y = 345$ MPa, when $t = [18\,35]$ mm ja $f_y = 335$ MPa, when $t = [40\,50]$ mm.

92 Computer Aided Optimum Design of Structures VII

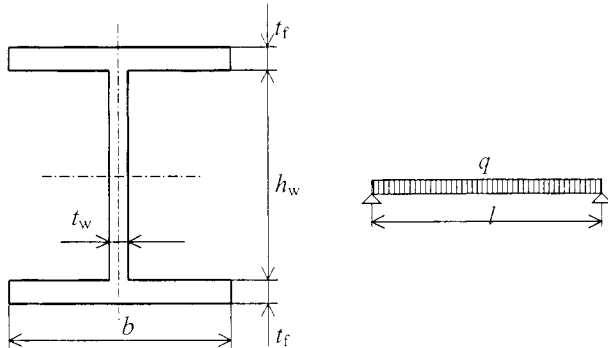


Figure 2: Dimensions of the I-beam.

Cross section characteristics:

$$\begin{aligned}
 h &= 2 \cdot t_f + h_w \\
 b_f &= b - t_w \\
 A &= b \cdot h - b_f \cdot h_w \\
 e_z &= \frac{1}{2} h \\
 e_f &= \frac{1}{2} (h_w + t_f) \\
 I_z &= \frac{1}{12} b \cdot h^3 - b_f \cdot h_w^3 \\
 W &= \frac{b \cdot h^3 - b_f \cdot h_w^3}{6 \cdot h} \\
 W_p &= 2 \cdot b \cdot t_f \cdot e_f + \frac{1}{4} h_w^2 \cdot t_w \\
 q_m &= A \cdot \rho \cdot g
 \end{aligned} \tag{5 a-i}$$

Cross section class of the flange:

$$PL_f = \begin{cases} 4, & \text{if } \frac{b_f}{t_f} > 0.44 \sqrt{\frac{E}{f_{yf}}} \\ 3, & \text{if } \frac{b_f}{t_f} \leq 0.44 \sqrt{\frac{E}{f_{yf}}} \\ 2, & \text{if } \frac{b_f}{t_f} \leq 0.36 \sqrt{\frac{E}{f_{yf}}} \\ 1, & \text{if } \frac{b_f}{t_f} \leq 0.30 \sqrt{\frac{E}{f_{yf}}} \end{cases} \tag{6}$$

Design strength:

$$f_d = \min\{f_{dw}, f_{df}\} \quad (7)$$

$$N_p = \frac{f_y}{A} \quad (8)$$

Cross section class of the web:

$$PL_w = \begin{cases} 1, & \text{when } \begin{cases} \frac{h_w}{t_w} \leq 3.00 \cdot \left(1 - 1.60 \frac{N}{N_p}\right) \cdot \sqrt{\frac{E}{f_{yw}}} & \text{ja } \frac{N}{N_p} < 0.125 \\ \frac{h_w}{t_w} \leq 2.57 \cdot \left(1 - 1.53 \frac{N}{N_p}\right) \cdot \sqrt{\frac{E}{f_{yw}}} & \text{ja } \frac{N}{N_p} \geq 0.125 \end{cases} \\ 2, & \text{when } \begin{cases} \frac{h_w}{t_w} \leq 2.40 \cdot \left(1 - 1.40 \frac{N}{N_p}\right) \cdot \sqrt{\frac{E}{f_{yw}}} & \text{ja } \frac{N}{N_p} < 0.39 \\ \frac{h_w}{t_w} \leq 1.10 \cdot \sqrt{\frac{E}{f_{yw}}} & \text{ja } \frac{N}{N_p} \geq 0.125 \end{cases} \\ 4, & \text{otherwise} \end{cases} \quad (9)$$

Cross section class of the beam:

$$PL = \min\{PL_f, PL_w\} \quad (10)$$

Check of the bending stress:

$$\eta = \begin{cases} \frac{W_c}{W}, & \text{when } PL = 4 \\ 1, & \text{when } PL = 3 \\ \frac{W_p}{W}, & \text{when } PL = \{1, 2\} \end{cases}$$

have to be $\eta \leq 1.2$

$$M_R = \eta \cdot f_d \cdot W \quad (11a-c)$$

$$R_2 = 1 - \frac{M}{M_R}$$

94 Computer Aided Optimum Design of Structures VII

Total mass of the beam:

$$m = A \cdot l \cdot \rho \quad (12)$$

Limit state check for the deflection of the beam:

$$\begin{aligned} v &= \frac{5}{384} \frac{q_l \cdot l^4}{E \cdot I_z} \\ R_1 &= 1 - \frac{v}{v_{\text{all}}} = \frac{v}{l \cdot k_{v\text{all}}} \end{aligned} \quad (13)$$

When possible, the above formulas are given in the form used by the standard [7][10].

The restraints are formulated to satisfy the demands of the limit state dimensioning method.

Safety factor of the loading $\gamma_F = 1.0$. [10]

Restraints are calculated from the formula

$$R = 1 - x / x_R \quad (14)$$

Restraint R is satisfied if $x \leq x_R$.

Analysis

The solution of the optimisation problem may differ from one run to another if the population is not large enough. The population size is set so that about 95 % of the solution results are the same. The selected population size is 100. The maximum number of generations is 200.

The differential factor is constant $F = 0.6$ and the mutation probability $CR = 0.95$. The optimisation algorithm has been run both with and without the limitation of $l/300$ on deflection.

Results

When deflection was not limited the optimised cross section was: *Flanges*: thickness $t_f = 30$ mm, yield limit $f_{yf} = 345$ MPa, width $b = 205.7$ mm. *Web*: thickness $t_w = 16$ mm, yield limit $f_{yw} = 355$ MPa, height $h_w = 1167$ mm. *Beam*: total mass $m = 7305$ kg, deflection $v = 310$ mm, cross section class $PL = 2$.

Restraints: deflection limit $R_1 = 1.0$, strength limit $R_2 = 1.0$.

When the deflection was limited to $l/300$ the optimised cross section was: *Flanges*: thickness $t_f = 50$ mm, yield limit $f_{yf} = 335$ MPa, width $b = 144.3$ mm. *Web*: thickness $t_w = 20$ mm, yield limit $f_{yw} = 345$ MPa, height $h_w = 1580$ mm. *Beam*: total mass $m = 10\,370$ kg, deflection $v = 150$ mm, cross section class $PL = 2$. *Restraints*: deflection limit $R = 0.0$, strength limit $R = 0.36$.

Discussion and conclusion

The optimisation example is the optimisation of the cross section of an I-beam. The beam was uniformly loaded. The results were compared to the results of analytical optimisation. [9]. The optimisation problem was easy to reformulate. The values of the restraints were monitored and critical restraints were easy to detect.

The optimisation program and the implemented optimisation algorithm help the designer to solve dimensioning problems. The optimisation model is easy to build. The formulas used in the standards do not need significant reformulation. The selection of structural components can be done during the optimisation.

The advantages of the Differential Evolution are mixed number optimisation and the fact that the optimisation can be distributed.

The disadvantages of the Differential Evolution are that the user has to select a suitable population size and that the solution is time consuming if the population is large.

The advantages of the optimisation program are: 1) the solution satisfies restraints; 2) the optimisation model is easy to build using component libraries, and also an earlier optimisation model can be used; 3) the optimisation model is easy to modify; 4) it is easy to use discrete data.

References

- [1] Rozvany, G.I.N. and Zhou M., Layout and Generalized Shape Optimization by Iterative COC Methods, *Optimisation of Large Structural Systems*. Dordrecht, Netherlands: Kluwer Academic Publishers. Vol. 1, pp. 103-120., 1993.
- [2] Price K. V., An Introduction to Differential Evolution. *New Ideas in Optimization*, Edited by Corne D., Dorigo M. and Glover F. The McGraw-Hill Companies. pp. 79-108., 1999.
- [3] Lampinen J. and Zelinka I., Mechanical Engineering Design Optimization by Differential Evolution. *Advanced Topics in Computer Science Series*. Edited by Corne, D, Dorigo M. and Glover Fred. pp. 127-146, 1999.
- [4] Coello C., Treating Constraints as Objectives for Single-objective Evolutionary Optimization. *Engineering Optimization Vol. 32* pp. 275-308, 2000.
- [5] Viitanen S., Some New Global Optimization Algorithms. Diss. Department of Computer Science. Åbo Akademi University, 1997.
- [6] Rautaruukki Steel Products, Designer's Guide. Otava, Keuruu, 1996.
- [7] Ympäristöministeriö, Asunto- ja rakennusosasto. Finnish code of building regulations B7. 51 p., 1996. (in Finnish)
- [8] Farkas J. & Jarmai K., Analysis and Optimum Design of Metal Structures. A.A. Balkema. Rotterdam, 1997.
- [9] Niemi E., Karkealevyrakenteiden suunnittelu. Tekninen tiedotus 3/81. Suomen metalliteollisuuden keskusliitto., 1981. (in Finnish)
- [10] MacGinley T.J. & Ang T. C., Structural Steelwork: Design to Limit State



96 *Computer Aided Optimum Design of Structures VII*

Theory. Butterworth&Co Ltd. 359 p., 1987.

- [11] Lampinen J., Differential Evolution - New Naturally Parallel Approach for Engineering Design Optimization. *Developments in Computational Mechanics with High Performance Computing*. Edited by Topping B. H. V. Civil-comp press. Edinburgh, UK. pp. 217-228, 1999.