

## OPTIMAL CONSTRAINED DESIGN OF STEEL STRUCTURES BY DIFFERENTIAL EVOLUTIONARY ALGORITHMS

R. Greco<sup>\*,†</sup> and G.C. Marano

*Department of Civil Engineering and Architecture, Technical University of Bari, Via  
Orabona, 4, 70100, Bari, Italy*

### ABSTRACT

Structural optimization, when approached by conventional (gradient based) minimization algorithms presents several difficulties, mainly related to computational aspects for the huge number of nonlinear analyses required, that regard both Objective Functions (OFs) and Constraints. Moreover, from the early '80s to today's, Evolutionary Algorithms have been successfully developed and applied as a computational alternative to many optimization problems, such as structural ones. In this study the effectiveness of a relatively new Evolutionary Algorithm, namely Differential Evolutionary, is investigated for constrained optimization. This presents many interesting advantages and so that it is a candidate to be widely used in many real structural optimization problems. The algorithm version here used has been developed by hybridizing some recent versions of Differential Evolutionary algorithms proposed in literature, and uses a specific way for dealing with constraints which, always, concern real structural optimization problems. The effectiveness of proposed approach has been demonstrated by developing two cases of study, which regard simple but very significant structural problems for steel structures, one of which is a standard benchmark in structural optimization. The analyses show the simplicity and effectiveness of the proposed approach, so that it can be suitably ready for practical uses out of academic contest.

Received: 18 October 2011; Accepted 27 December 2011

**KEY WORDS:** Evolution algorithms (EAs); differential evolution algorithms (DEAs);  
constraint handling problems; structural optimization; steel elements  
optimization

---

\*Corresponding author: R. Greco, Department of Civil Engineering and Architecture, Technical University of Bari, Via Orabona, 4, 70100, Bari, Italy

†E-mail address: [g.c.marano@gmail.com](mailto:g.c.marano@gmail.com)

## 1. INTRODUCTION

Optimization problems are ubiquitous in real-world applications more than in academic researches. In principle, every human activity should be subject to an “optimization” in order to better use available resources and to maximize their efficiency. However, in front of this philosophical definition, in an engineering sense, optimization is a mathematical problem which needs of an appropriate representation of resources to be preserved and constraints to be satisfied. With no doubt, researchers and practitioners need an efficient and robust optimization approach in order to solve problems of different characteristics that are fundamental to their daily work. It is expected that, solving a complex optimization problem itself should not be very difficult; e.g. an engineer with expert knowledge of channel coding does not have to be an expert in optimization theory just to improve his/her code design. In addition, an optimization algorithm should be able to reliably converge to the true optimum, for a variety of different problems. Furthermore, the computing resources spent on searching for a solution should not be excessive. Thus, a useful optimization method should be easy to use, reliable and efficient to achieve satisfactory solutions. Instead of classical methods, that rarely (or never) present above mentioned characteristics, it is interesting to observe that some Evolution Algorithms (EAs) have these abilities.

Nowadays, the use of EAs to solve optimization problems is a common practice due to their competitive performance on complex search spaces [1,2]. On the other hand, optimization problems usually include constraints in their models, but EAs, in their original version, do not consider a mechanism to incorporate feasibility information in the search process. Therefore, several constraint-handling techniques have been proposed in the specialized literature [3,4].

Specifically, Differential Evolution (DE) is a heuristic method that has yielded promising results for solving complex optimization problems. The potentialities of DE are its simple structure, easy use, convergence property, quality of solution, and robustness. Since its first formulation by Storn and Price [5], DE has been shown to be a simple yet efficient optimization approach for solving a variety of benchmark problems as well as many real-world applications. Differential Evolution, together with Evolution Strategies (ES) [6,7], Genetic Algorithms (GA) [8] and Evolutionary Programming (EP) [9], can be categorized into a class of population-based, derivative-free methods, known as Evolutionary Algorithms (EAs). All these approaches mimic Darwinian evolution and evolve a population of individuals from one generation to another, by analogous evolutionary operations such as mutation, crossover and selection.

All attractive features of EAs are in opposition to some criticisms. For instance, EAs suffer the lack of well posed theories about their convergence and, typically, a larger computational time is required. Moreover, in their original formulation they were limited to unconstrained problems and do not include a method to incorporate feasibility information into the fitness function. In effect, a primary question in structural engineering is to produce, as optimal solution, a feasible condition, according to all constraints involved. In more effective words, it simply means that engineers firstly need of solutions that satisfy all constraints. Calling as unfeasible all solutions that unsatisfied at least one of the problem constraints (feasible are solutions which satisfy all constraints), there are no cases for any engineering to use

"unfeasible" solutions in problem solving. Actually, there are different approaches for dealing with constrained optimization. The most popular one is the use of (mainly exterior) penalty functions [10], whose the aim is to decrease the fitness of infeasible solutions in order to encourage the selection of feasible solutions. Despite its simplicity, a penalty function requires the definition of penalty factors to determine the severity of the penalization, and these values depend on the problem being solved [11]. Based on this important disadvantage, several alternative constraint-handling techniques have been proposed. In [12] it is developed a comprehensive study of DEs applied to constraints problems from a numerical point of view, where the efficiency in different combinations of mutations and constraint handling techniques is analyzed. Moreover, this important piece of paper is mainly related to dealing with the numerical approach in mathematical problems. On the contrary, from the engineering point of view, more in structural one, it is fundamental that the final solution is a feasible one, that simply means it satisfy all constraints. Moreover, it is possible to state that any structural engineering has to satisfy primary constraints, and only subsequently the level of optimization. In few words, one can say that the main require in structural design is "safety first" and only then engineers are interested in cost reduction.

Among alternative approaches, multi-objective optimization techniques can be used to solve constrained problems by treating the constraint as one or several objectives. In this way, the constraints and the objective function are optimized simultaneously. Surry and Radcliffe [13] treated the constrained optimization problem as a constrained satisfaction problem, by ignoring the objective function and treated it as an unconstrained optimization problem by neglecting the constraints. In [14] the authors proposed a method based on the Pareto dominance concept and produced very competitive results. Besides, in [15] all constraints are treated as one objective and the constrained optimization it is solved as a bi-objective optimization problem.

Some other popular methods treat the objective function and constraints separately. Deb [16] proposed a constraint handling technique based on feasibility rules for pair-wise comparison. In particular, feasible solutions are always better than infeasible ones; comparison between feasible solutions depends on the objective function values; and comparison between infeasible solutions depends on constraint violations. To balance dominance between the objective and penalty functions, Runarsson and Yao [17] proposed a Stochastic Ranking (SR) method. A parameter  $p_f$  was used to determine the choice between the objective and the constraint violation when comparing two solutions in the sorting process. Inspired by fuzzy control theory, Takahama and Sakai [18] introduced a satisfaction level for the constraints, in order to indicate how well a solution satisfied the constraints, and the selection of solutions was conducted based on both the satisfaction level and the objective function. In addition, a homomorphous mapping [19] was proposed to transform the feasible region to a high-dimensional cube with simple topology. For such kind of methods, the design of special operators is problem-dependent and it is difficult to generalize.

In this study a specific constraint handling approach for DEs is developed, and advantages in using this modified version are presented. The algorithm here proposed is developed by a robust formulation and few parameters, with the aim of increase its appeal for practical applications. Two practical cases of study, regarding simple but significant structural problems for steel structures, one of which is a standard benchmark in structural optimization,

are analyzed. Results show the simplicity and effectiveness of the proposed approach, so that it can be suitably ready for practical uses out of academic contest.

The next part of the paper is organized as follows: in section 2 the basic formulation of proposed DE is summarized. In section 3 the proposed algorithm is applied to optimal design of two simple but very effective problems of steel structures. In section 4 some conclusions are extrapolated.

## 2. DIFFERENTIAL EVOLUTION ALGORITHM

Like nearly all EAs, differential evolution algorithm is a population-based optimizer that starts the optimization process by sampling the search space at multiple, randomly chosen initial points (i.e., a population of individual vectors) [20]. Similar to ES, DE algorithm is in nature a derivative-free continuous function optimizer, as it encodes parameters as floating-point numbers and manipulates them with simple arithmetic operations such as addition, subtraction and multiplication. Like other evolutionary algorithms, DE generates new points that are the perturbations/mutations of existing points; the perturbations, however, come neither from samples of a default probability distribution like those in ES, not from centroid-based difference vectors as in some nonlinear optimization methods, such as the Nelder-Mead method [21]. Instead, DE mutates a (parent) vector in the population with a scaled difference of other randomly selected individual vectors. The resultant mutation vector is crossed over with the corresponding parent vector to generate a trial or offspring vectors. Then, in a one-to-one selection process of each pair of offspring and parent vectors, the one with a better fitness value survives and enters the next generation. This procedure repeats for each parent vector and the survivors of all parent-offspring pairs become the parents of a new generation in the evolutionary search cycle. The evolutionary search stops when the algorithm converges to the true optimum or a certain termination criterion such as the number of generations is reached.

A general constrained optimization problem can be formulated as a typical minimization problem in the form:

$$\begin{aligned} & \min_{\mathbf{x}} \{f(\mathbf{x})\} \\ & \text{subject to} \\ & g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, n_p \\ & h_i(\mathbf{x}) = 0, \quad i = 1, \dots, n_q \\ & \mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u \end{aligned} \quad (1)$$

in which  $\mathbf{x} = \{x_1, \dots, x_j, \dots, x_D\}$  is the design vector (for example the collection of  $D$  system parameters to be identified),  $\mathbf{x}^l = \{x_1^l, \dots, x_j^l, \dots, x_D^l\}$  and  $\mathbf{x}^u = \{x_1^u, \dots, x_j^u, \dots, x_D^u\}$  are its lower and upper bounds, respectively, and  $f(\mathbf{x})$  is the objective function. In addition,  $g_i(\mathbf{x})$  and  $h_i(\mathbf{x})$  are inequality and equality constraints, respectively. The shape of the objective function may have many local optima and high complex topology and, therefore, when preliminary information are not available, it may not always be convex. In these circumstances special optimizers have to be used. In the following it is illustrated the state of the art of DEa for

problems in form (1).

The initial population  $\{\mathbf{x}_{i,0} = x_{1,i,0}, x_{2,i,0}, \dots, x_{D,i,0} \mid i = 1, 2, \dots, NP\}$  is randomly generated according to a normal or uniform distribution, being  $x_j^l \leq \mathbf{x} \leq x_j^u$ , for  $j=1, 2, \dots, D$ , where NP is the population size, D is the dimension of the problem, and where  $x_j^l$  and  $x_j^u$  are the upper and lower limits of the j-th component of the vector  $\mathbf{x}$ . After initialization, DE enters a loop of evolutionary operations: mutation, crossover and selection.

### 2.1. Mutation

The main idea of DEa is to construct at each generation and for each element of the population a mutation vector. At each generation  $g$ , this operation creates mutation vectors  $v_{i,g}$  based on the current parent population  $\{x_{i,g} = x_{1,i,g}, x_{1,i,g}, \dots, x_{D,i,g} \mid i = 1, 2, \dots, NP\}$ . The mutant vector is constructed through a specific mutation operation based on adding differences between randomly selected elements of the population to another element.

Different mutation strategies frequently used in the literature are (Price at all., 2005):

$$\text{DE/rand/1} \quad v_{i,g} = x_{r_0,g} + F_i(x_{r_1,g} - x_{r_2,g}) \quad (2)$$

$$\text{DE/current-to-best/1} \quad v_{i,g} = x_{i,g} + F_i(x_{best,g} - x_{i,g}) + F_i(x_{r_1,g} - x_{r_2,g}) \quad (3)$$

$$\text{DE/best/1} \quad v_{i,g} = x_{i,g} + x_{best,g} + F_i(x_{r_1,g} - x_{r_2,g}) \quad (4)$$

where the indices  $r_0$ ,  $r_1$  and  $r_2$  are distinct integers uniformly chosen from the set  $\{1, 2, \dots, NP\} \setminus \{i\}$ ,  $x_{r_1,g} - x_{r_2,g}$  is a difference vector in order to mutate the parent,  $x_{best,g}$  is the best vector at the current generation  $g$ , and  $F_i$  is the mutation factor which usually ranges on the interval (0, 1+). In classic DE algorithms,  $F_i = F$  is a single parameter used for the generation of all mutation vectors, while in many adaptive DE algorithms each individual "i" is associated with its own mutation  $F_i$ .

The above mutation strategies can be generalized by implementing multiple difference vectors other than  $x_{r_1,g} - x_{r_2,g}$ . The resulting strategy is named as 'DE/-/k' depending on the number k of difference vectors adopted.

The mutation operation may generate trial vectors whose components violate the default boundary constraints. Possible solutions to tackle this problem include resetting schemes, penalty schemes, etc (Prince K., 2005). A simple method is to set the violating component to be the middle between the violated bound and the corresponding components of the parent individual, i.e.:

$$\begin{aligned}
 v_{j,i,g} &= \frac{x_j^l + x_{j,i,g}}{2} \quad \text{if } v_{j,i,g} < x_j^l \\
 v_{j,i,g} &= \frac{x_j^u + x_{j,i,g}}{2} \quad \text{if } v_{j,i,g} > x_j^u
 \end{aligned} \tag{5}$$

where  $v_{j,i,g}$  and  $x_{j,i,g}$  are the  $j$ -th components of the mutation vector  $v_{i,g}$  and the parent vector  $x_{i,g}$  at generation  $g$ , respectively. This method performs well, especially when the optimal solution is located near or on the boundary.

## 2.2. Crossover

After mutation, a ‘binomial’ crossover operation forms the final trial vector  $u_{i,g} = (u_{1,i,g}, u_{1,i,g}, \dots, u_{D,i,g})$

$$u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{if } \text{rand}_j(0,1) \leq CR_i \text{ or } j = j_{rand} \\ x_{j,i,g} & \text{otherwise} \end{cases} \tag{6}$$

where  $\text{rand}_j(a,b)$  is a uniform random number on the interval  $(a, b]$  and newly generated for each  $j$ ,  $j_{rand} = \text{rand int}_j(1, D)$  is an integer randomly chosen from 1 to  $D$  and newly generated for each  $i$ , and the crossover probability  $CR_i \in [0, 1]$ , roughly corresponds to the average fraction of vector components that are inherited from the mutation vector. In classic DE,  $CR_i = CR$  is a single parameter that is used to generate all trial vectors, while in many adaptive DE algorithms, each individual “ $i$ ” is associated with its own crossover probability  $CR_i$ .

## 2.3. Selection

The selection operation selects the better one from the parent vector  $x_{i,g}$  and the trial vector  $u_{i,g}$  according to their fitness values  $f(x_{i,g})$ . For example, in a minimization problem, the selected vector is given by:

$$x_{i,g+1} = \begin{cases} u_{i,g} & \text{if } f(u_{i,g}) < f(x_{i,g}) \\ x_{i,g} & \text{otherwise} \end{cases} \tag{7}$$

and used as a parent vector in the next generation. The above one-to-one selection procedure is generally kept fixed in different DE algorithms, while the crossover may have variants other than the binomial operation in (2.6). Thus, a DE algorithm is historically named, for example, DE/rand/1/bin connoting its DE/rand/1 mutation strategy and binomial crossover operation.

#### 2.4. Handling Constraints

The main aim of this work is to develop a simple and ready algorithm for structural optimization, able to properly treat constraints according engineering requires. In this way, a methodology that partially adopts the Deb feasibility rules is developed, that uses not only information about performance but also about feasibility of two individuals.

Let consider the following violation function for the  $i$ th individual:

$$\Phi(x_{i,g}) = \sum_{p=1}^{n_g+n_r} \max\{0, g_p(x_{i,g})\} \geq 0 \quad (8)$$

Its value is zero if and only if all constrains are satisfied and it is a positive scalar number otherwise. Otherwise, the individual lies outside the feasible region. The Deb's rule simply assumes that in any case a feasible individual is preferred to an unfeasible one. This is a sort of static domination-based selection scheme that can be formulated for as follows:

$$x_{i,g+1} = \begin{cases} u_{i,g} & \text{if } \{f(u_{i,g}) < f(x_{i,g})\} \text{ and } \{(\Phi(x_{i,g}) = 0) \text{ and } (\Phi(u_{i,g}) = 0)\} \\ x_{i,g} & \text{if } \{(\Phi(x_{i,g}) = 0) \text{ and } (\Phi(u_{i,g}) > 0)\} \\ u_{i,g} & \text{otherwise} \end{cases} \quad (9)$$

Actually, it works by starting from a random initial population, that in principle (especially dealing with limited populations) should be of unfeasible individuals only. In this view there is no assurance that the final optimal solution is a feasible one, that is unacceptable from the practical point of view. To overcome this limitation and by using the specific aspect of DEa, an initial population that presents at least one feasible individual will assure a feasible final optimal solution. Due to specific one to one selection of DEa by using Deb feasibility rules, and starting with some feasible individuals in the initial population, it is assured that all feasible individuals will survive also in the worst case, that other feasible ones produced by mutation and crossover application. This simply because the Deb selection rule preserves a feasible solution when compared with unfeasible one. So that, the number of feasible individuals present in the initial population can only increase, and this gives the assurance that final optimal solution will be a feasible one. By using this approach an initial population check, that simply verifies if and how many feasible individuals are present in the initial random generated population, is developed. If this number isn't reached, the random generation continues till a minimum number of feasible individuals are really produced. At this point the DE starts normally by using the Deb feasible selection rule. This gives the main advantage (that is imperative in structural design) to produce in the worst case a low optimized, but feasible (that means safe), final solution.

### 3. APPLICATION OF MODIFIED DIFFERENTIAL EVOLUTION ALGORITHMS AND RESULTS

In this section, two simple but effective cases of study are developed, regarding steel structures, in order to illustrate the efficiency of the proposed methodology in practical structural optimization problems.

- Example 1: *Optimal Design of a Simply Supported Beam with Uniformly Distributed Load;*
- Example 2: *A Welded Beam at the free end Optimal Design loaded.*

The analyzed studies have been addressed according to Eurocode 3, which regulates the design of steel structures and to rules governing the welding UNI (UNI EN 287-1 2004, UNI EN 719 of 1996, UNI EN ISO 14329, 15607, 15610 and 15611 of 2005).

#### 3.1. Optimal Design of a Simply Supported Beam with Uniformly Distributed Load

The optimization problem is aimed at determining the best section of a steel beam with a IPE profile, simply supported and subject to an uniformly distributed load. More in details, a steel double pinned beam (Fig.1a ) is considered whose modulus of elasticity  $E$  is 200 GPa and whose yield stress  $f_y$  is 235 MPa. The cross-section of this beam is shown in Figure 1b.

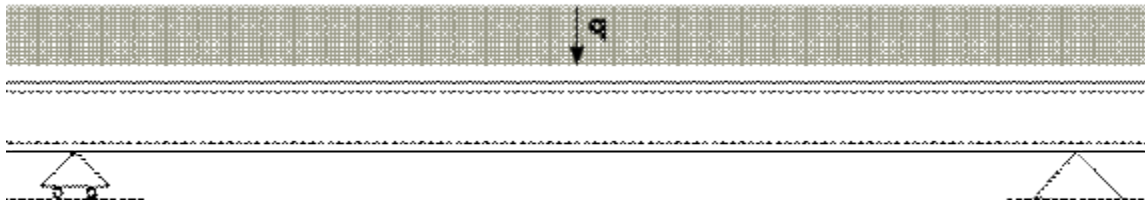


Figure 1a. Simply supported beam with an uniformly distributed load

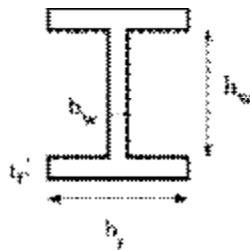


Figure 1b. Cross section of IPE

The optimization problem attempts to minimize the weight of this simple supported beam



in accordance with the constraints of EC3 code.

According to Chapter III of the EC3 the following load combination is considered:

$$q = g_g G_k + g_q Q_k \quad (10)$$

where  $Q_k = 200 \text{ daN} / \text{m}$  and  $G_k = \left[ 2(b_f t_f) + t_w h_w \right] 7850 \text{ daN} / \text{m}$ .

In addition,  $g_g$  and  $g_q$  are load safety coefficients and are assumed 1,3 and 1,5 respectively. The function to be minimized is the weight:

$$w = gL(A_w + 2A_f) \quad (11)$$

For this structure the following optimization problem is then posed:

*Minimize*

$$w(\mathbf{x}), \quad \mathbf{x} = (t_f, b_f, t_w, h_w) \in \mathbf{R}^4 \quad (12)$$

*Under the following constraints:*

$$\begin{aligned} g_1(\mathbf{x}) &\leq 1 \\ g_2(\mathbf{x}) &\leq 1 \\ g_2(\mathbf{x}) &\leq 1 \\ g_2(\mathbf{x}) &\leq 1 \end{aligned} \quad (13)$$

$$\begin{aligned} b_f^l &\leq b_f \leq b_f^u \\ h_w^l &\leq h_w \leq h_w^u \\ t_f^l &\leq t_f \leq t_f^u \\ h_w^l &\leq h_w \leq h_w^u \end{aligned} \quad (14)$$

where

$$\begin{aligned} g_1(\mathbf{x}) &= M_{ed}(\mathbf{x}) / M_{c,rd}(\mathbf{x}) \\ g_2(\mathbf{x}) &= V_{ed}(\mathbf{x}) / V_{i,pl,Rd}(\mathbf{x}) \\ g_3(\mathbf{x}) &= M_{ed}(\mathbf{x}) / M_{b,rd}(\mathbf{x}) \\ g_4(\mathbf{x}) &= d_{\max}(\mathbf{x}) / d_{\lim}(\mathbf{x}) \end{aligned} \quad (15)$$

being  $M_{c,rd}$ ,  $V_{i,pl,Rd}$ ,  $M_{b,rd}$  and  $d_{\lim}$  calculated according to Eurocode 3

### 1) Bending moment

This constraint is given by the following formula of Eurocode 3:

$$M_{ed} \leq M_{c,rd} \quad (16)$$

where  $M_{ed} = \frac{1}{8}qL^2$  is the calculus bending moment and  $M_{c,rd}$  is the allowable bending moment. This is given by relation:

$$M_{c,rd} = W_{pl} \frac{f_y}{g_{M0}} \quad (17)$$

where  $f_y$  the yield strength of the steel,  $W_{pl}$  is the plastic moment of the section (for section of class 1 and 2) given by the following equation, and  $g_{M0} = 1.05$ .

$$W_{pl} = \frac{t_w h_w^2}{4} + b_f t_f (h_w + t_f) \quad (18)$$

### 2) Shear

This constraint is given by the following formula of Eurocode 3:

$$V_{ed} \leq V_{i,pl,Rd} \quad (19)$$

where  $V_{ed} = q \frac{L}{2}$  is the calculus shear and  $V_{i,pl,Rd}$  is the allowable shear. This is given by:

$$V_{i,pl,Rd} = \frac{A_{v,i} f_y}{g_{M0} \sqrt{3}}, \quad A_{v,i} = A - 2b_f t_f + t_w t_f \quad (20)$$

being  $A = 2(b_f t_f) + t_w h_w$  the total section area.

### 3) Flexural buckling

A laterally unrestrained member subject to major axis bending should be verified against lateral-torsion buckling by means of the following inequality given by Eurocode 3:

$$M_{ed} \leq M_{b,rd} \quad (21)$$

where  $M_{ed}$  is the design value of the moment and  $M_{b,rd}$  is the design buckling resistance moment.

The design buckling resistance moment of a laterally unrestrained beam for section of class 1 and 2 is

$$M_{b,Rd} = C_{LT} W_{pl} \frac{f_{yk}}{g_{M1}} \quad (22)$$

$g_{M1}$  = is the safety factor for the instability, whereas  $C_{LT}$  is a reduction factor and can be

evaluated by means relation:

$$c_{LT} = \frac{1}{\Phi_{LT} + \sqrt{\Phi_{LT}^2 - b\bar{I}_{LT}^2}} \leq \begin{cases} 1 \\ \frac{1}{\bar{I}_{LT}^2} \end{cases} \quad (23)$$

$$\Phi_{LT} = 0.5 \left[ 1 + a_{LT} (\bar{I}_{LT} - \bar{I}_{LT,0}) + b\bar{I}_{LT}^2 \right]$$

In equation (23)  $a_{LT}$  is the imperfection factor which corresponds to the appropriate buckling curve. The recommended values are given in table 1:

Table 1. Imperfection factor  $a_{LT}$

Buckling curve	a	b	c	d
Imperfection factor $a_{LT}$	0.21	0.34	0.49	0.76

The recommendation for buckling curves given by EC3 for rolled section are (Table 2):

Table 2. Buckling curves

Cross section	Limit	Buckling curve
	$h / b_f \leq 2$	<b>b</b>
	$h / b_f > 2$	<b>c</b>

In addition,  $\bar{I}_{LT} = \sqrt{\frac{W_{Pl} f_y}{M_{cr}}}$ , where  $M_{cr}$  is the elastic critical moment for lateral-torsion

buckling and it is based on gross cross sectional properties and takes into account the loading conditions, the real moment distribution and the lateral restraints. It is furnished by EC3 in the Appendix B by the following formula:

$$M_{cr} = C_1 \frac{p^2 EI_z}{(kL)^2} \sqrt{\left( \frac{k}{k_w} \right)^2 \frac{I_w}{I_z} + \frac{(kL)^2 GI_t}{p^2 EI_z}} \quad (24)$$

In equation (24)  $C_1$  is a coefficient which takes into account of distribution of moment and it is equal to 1.132 for a simply supported beam, whereas terms  $k$  and  $k_w$  are the effective length coefficients which depend on restraint conditions and are assumed equal to 1.

Moreover:

- $I_t = \frac{1}{3} [2b_f t_f^3 + h_w t_w^3]$  is the torsion constant;
- $I_w = \frac{t_f b_f^3}{24} h_w^2$  is the warping constant;
- $I_z = 2 \frac{t_f b_f^3}{12} + \frac{h_w t_w^3}{12}$  is the inertia moment around the minor axis;
- $L$  is the distance between two lateral restraints.

For rolled section or equivalent welded section EC3 for  $I_{LT,0}$  recommends the values 0.4. Finally  $b = 0.75$ .

#### 4) Displacement constraint

The displacement constraint is expressed by the following relation:

$$d_{\max} \leq d_{\lim} \quad (25)$$

where  $d_{\max} = \frac{qL^4}{348EJ}$  is the maximum deflection of the beam and  $d_{\lim} = \frac{L}{200}$  is given by EC3.

The constant parameters used in the optimization problem are given in Table 3:

Table 4. Constant input parameters in optimal design of simply supported beam

Parameters	E	L	f <sub>y</sub>	γ
Unit	MPa	mm	MPa	kg/m <sup>3</sup>
Value	200000	5000	235	7850

Table 5. Lower bound and the upper bound of design variables

Parameters	t <sub>r</sub>	b <sub>r</sub>	t <sub>w</sub>	h <sub>w</sub>
Unit	mm	mm	mm	mm
Upper Bound	13	200	10	200
Lower Bound	5	120	5	120

The problem has been solved through the use of Differential Evolution Algorithms with 200 elements from the initial population analyzed over 500 generations. The parameters used in the DEA (Differential Evolution Algorithms) are  $F^1 = F^2 = 0.5$  and  $p^c = 0.5$ .

The lower bound and the upper bound of design variables are listed in Table 5

Table 6: Optimal values of geometrical parameters and the objective function

$t_f$	$b_f$	$t_w$	$h_w$	$f(\bar{X})$
mm	mm	mm	mm	
12.997	120.003	5.000	199.993	1616.894

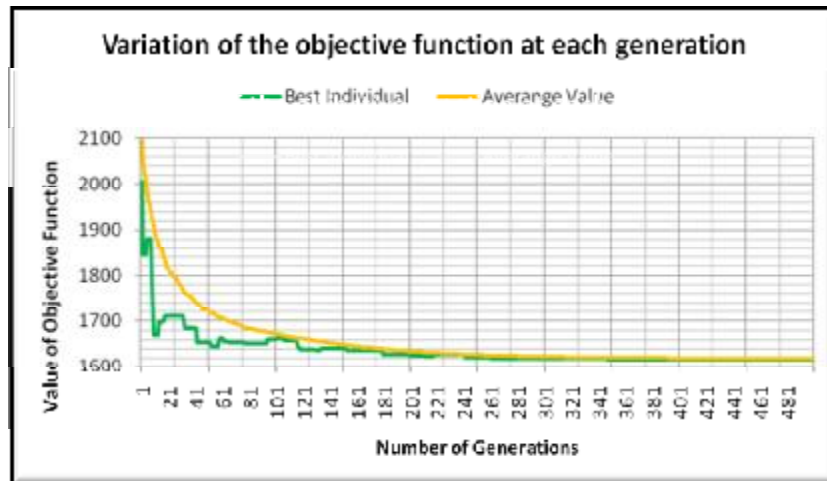
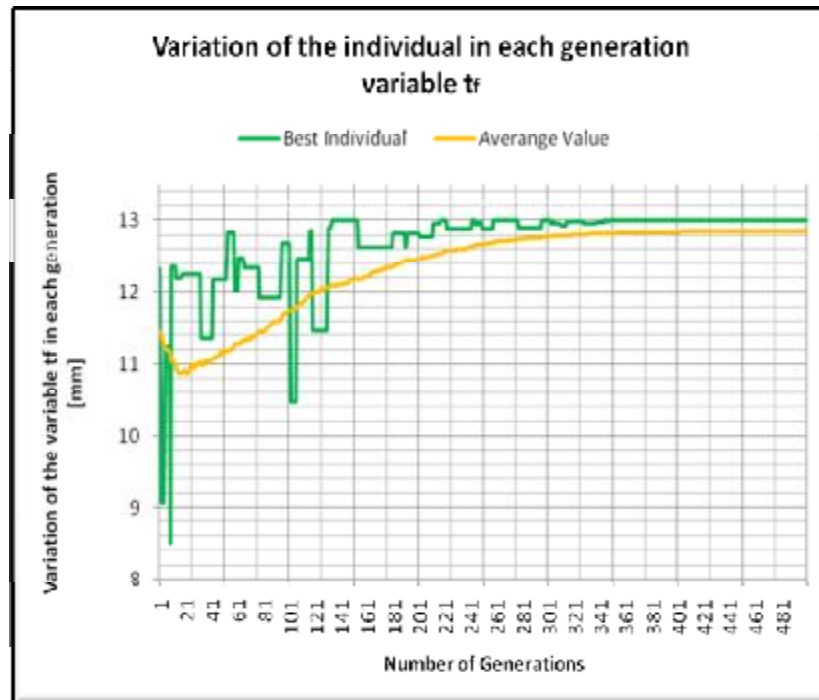


Figure 3. Variation of the objective function at each generation

Figure 4. Variation of the individual in each generation variable  $t_f$

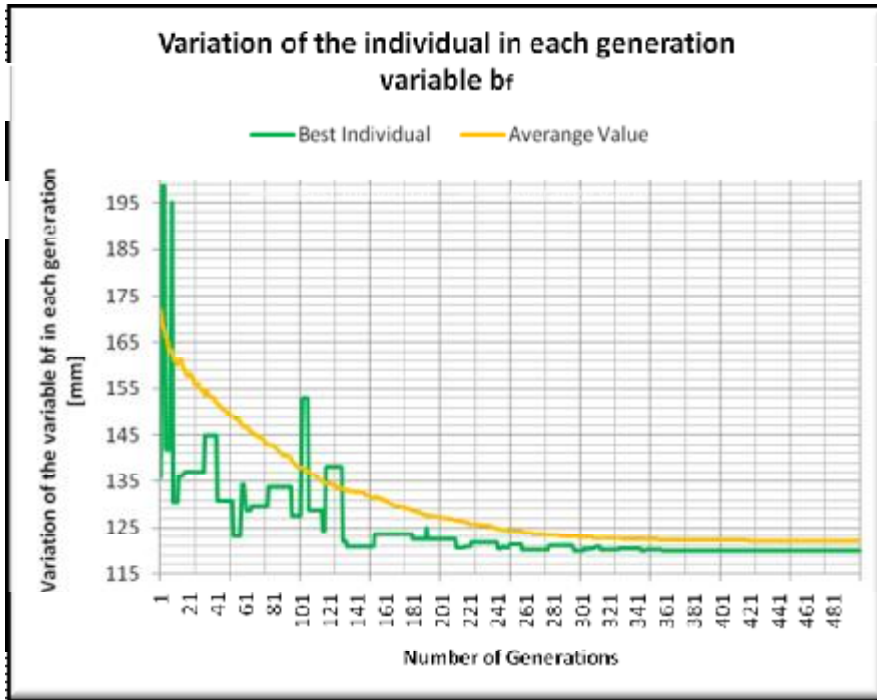


Figure 5. Variation of the individual in each generation variable  $b_f$

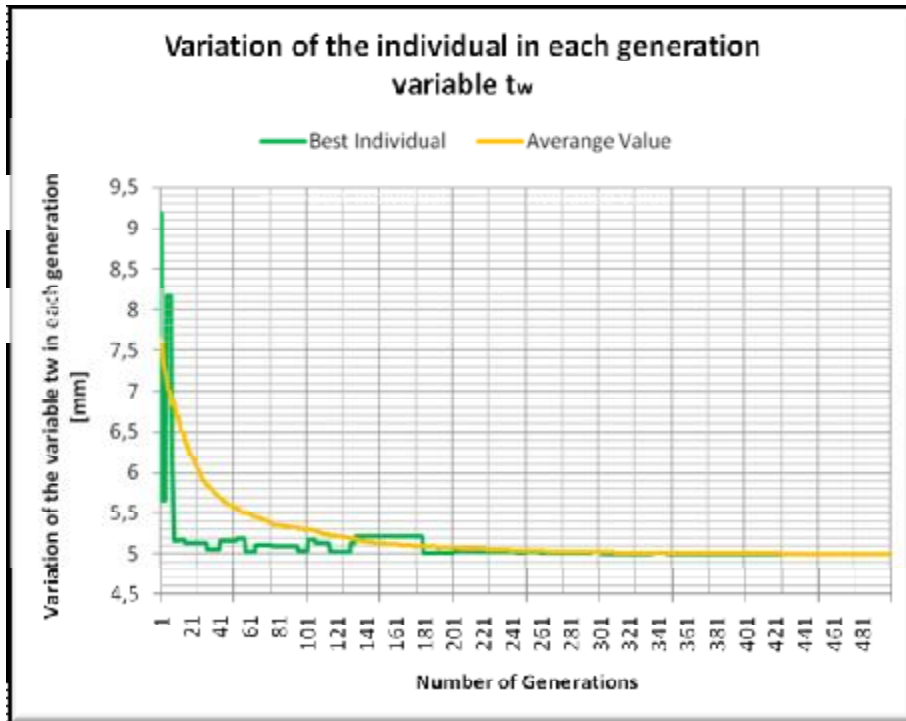


Figure 6. Variation of the individual in each generation variable  $t_w$

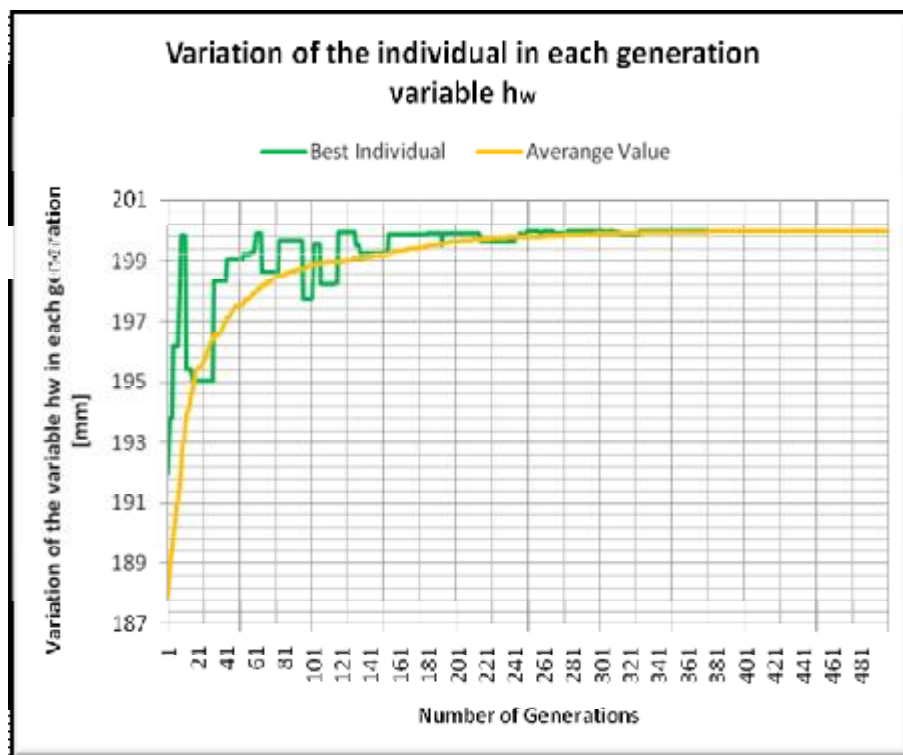


Figure 7. Variation of the individual in each generation variable  $h_w$

In this specific analyzed problem, that is strongly representative of many similar steel structure optimization problems, results are obtained within a relatively short time with an accurate precision. The optimal size of the IPE is shown in Table 6.

The optimal height ( $h_w$ ) is found to be equal to 200 mm; the comparison of this result with the size of the business's IPE shows that it is necessary to use a business profile of IPE400 in order to observe the remaining calculated size of the beam. The value of the height of the commercial division ( $h_w$ ) is 50% higher than the optimum needed value, the value of the width wing of the commercial division ( $b_f$ ) appears to be increased by 33.3%, the value of web's thickness of the commercial division ( $t_w$ ) appears to be increased by 41.86% and the value of the height of the wing of the commercial division ( $t_f$ ) appears to be increased by 3.7%.

The numerical results have been reported in Figures 3-7 that show how the development of the best individual, of the average value and the performance of the objective function at each generation is the better.

### 3.2. A Welded Beam Optimal Design

In this second case of study, the optimization is aimed at determining the lowest cost of construction of a shelf welded and loaded at the free end beam. For convenience, the parameters characterizing the project are expressed in the Anglo-Saxon metric system and then later they will be converted into the International System.

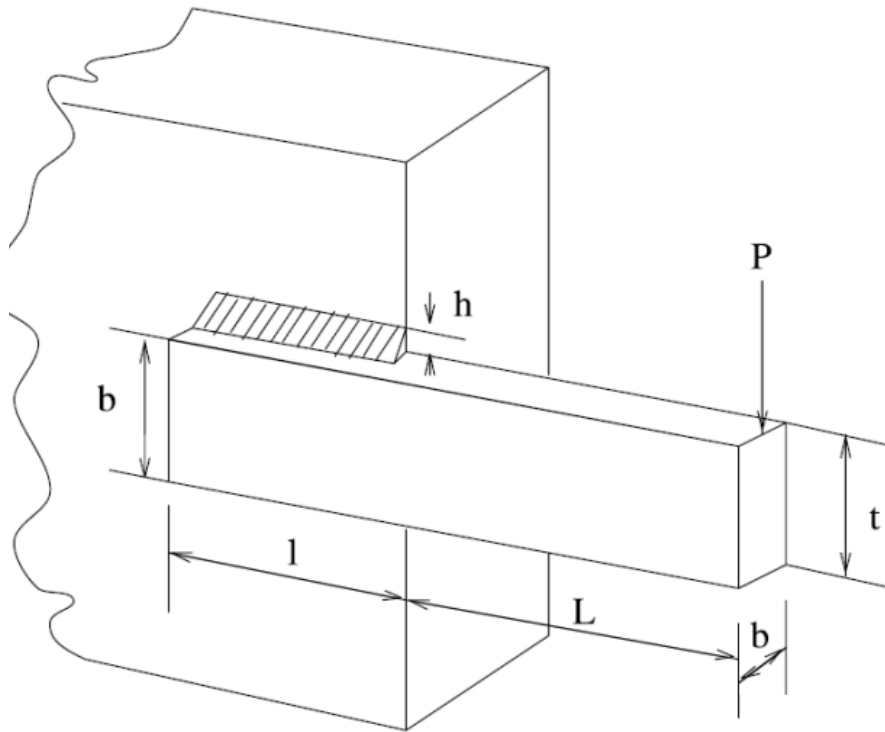


Figure 8. Illustrative pattern of the problem

The problem of optimal is bound by the shear stress ( $\tau$ ) calculation, by the tension due to bending moment ( $\sigma$ ), by the critical load acting on the rod ( $P_c$ ), by the beam bending deflection ( $\delta$ ), and by the geometric constraints. The parameters are calculated from the width of the beam ( $b$ ), height ( $t$ ), the height of the weld bead ( $h$ ) and the length of the cord itself ( $l$ ).

The optimum design attempts to minimize the cost function, including the cost of welding, the cost of labor and material, i.e.:

$$c = (1 + c_1)h^2l + c_2bt(L + l) \quad (26)$$

where  $c_1$  is the unit cost for volume of welding material ( $6.3898 \times 10^{-6} \text{ €/mm}^3$ ),  $c_2$  is the unit cost for volume of the bar ( $2.9359 \times 10^{-6} \text{ €/mm}^3$ ) and finally  $L$  is the distance from the load application, which in this case is equal to 356 mm.

For this structure the following optimization problem is then posed:

$$\begin{aligned} & \text{Minimize} \\ & c(\mathbf{x}), \quad \mathbf{x} = (h, l, t, b) \in \mathbf{R}^4 \end{aligned} \quad (27)$$

*Under the following constraints:*



$$\begin{aligned}
g_1(\mathbf{x}) &\leq 1 \\
g_2(\mathbf{x}) &\leq 1 \\
g_3(\mathbf{x}) &\leq 5 \\
g_4(\mathbf{x}) &\leq 1 \\
g_5(\mathbf{x}) &\leq 1 \\
g_6(\mathbf{x}) &\leq 1 \\
h^l &\leq h \leq h^u
\end{aligned}
\tag{28}$$

$$\begin{aligned}
l^l &\leq l \leq l^u \\
t^l &\leq t \leq t^u \\
b^l &\leq b \leq b^u
\end{aligned}
\tag{29}$$

where

$$\begin{aligned}
g_1(\mathbf{x}) &= t(\mathbf{x}) / t_{\max} \\
g_2(\mathbf{x}) &= s(\mathbf{x}) / s_{\max} \\
g_3(\mathbf{x}) &= (1 + c_1)h^2l + c_2bt(L + l) \\
g_4(\mathbf{x}) &= 0.125 / h \\
g_5(\mathbf{x}) &= d(\mathbf{x}) / d_{\max} \\
g_6(\mathbf{x}) &= P / P_c(\mathbf{x})
\end{aligned}
\tag{30}$$

being:

$$t(\mathbf{x}) = \sqrt{(t')^2 + 2t' \frac{t''l}{2R} + (t'')^2}
\tag{31}$$

$$t' = \frac{P}{\sqrt{2hl}} \quad t'' = \frac{MR}{J}
\tag{32}$$

$$M = P(L + l/2) \quad R = \sqrt{\frac{l^2}{4} + \left(\frac{h+t}{2}\right)^2}
\tag{33}$$

$$s(\mathbf{x}) = \frac{6PL}{bt^2} \quad d(\mathbf{x}) = \frac{4PL^3}{Ebt^3}
\tag{34}$$

$$P_c(\mathbf{x}) = \frac{4.013E}{L^2} \sqrt{t^2 \frac{b^6}{36}} \left( 1 - \frac{t}{2L} \sqrt{\frac{E}{4G}} \right)
\tag{35}$$

$$J = \begin{cases} \sqrt{2}hl \left[ \frac{l^2}{12} + \left( \frac{h+t}{2} \right)^2 \right] & \text{if there are two welded sides} \\ \sqrt{2}h \left( \frac{h+t+l}{12} \right)^3 & \text{if there are four wels} \end{cases} \quad (36)$$

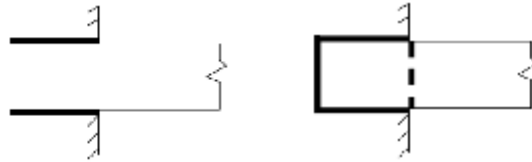


Figure 9. Element welded respectively on two or four sides

The constant parameters used in its optimization problem are listen in Table 7:

Table 7. Constant parameters of the problem of beam welded

P		L		E		G	
Lb	kgf	in	mm	psi	MPa	psi	MPa
6000	2721	14	355.60	3.00E+07	2.07E+05	1.20E+07	8.27E+04

$\sigma_{max}$		$\sigma_{tension}$		$\tau_{max}$	
in	mm	psi	MPa	psi	MPa
0.25	6.35	3000	2.07E+02	13600	9.38E+01

This application allows to search for the optimum constrained by geometric factors, stress and deformation in the civil and structural field. The problem was solved through the use of Differential Evolution Algorithms with 200 elements from the initial population analyzed over 500 generations. Moreover, the parameters used in the DEA (Differential Evolution Algorithms) are  $F^1 = F^2 = 0.5$  and  $p^c = 0.5$ .

The input variables parameters in the problem are given in Table 8:

Table 8. Input parameters variability of the welded beam problem

Parameters	h		L		t		b	
Unit	in	mm	in	mm	in	mm	in	mm
Upper bound	0.500	12.700	7.500	190.50	7.200	182.88	1	25.40
Lower bound	0.125	3.175	2.500	63.50	3	76.2	0.100	2.54

The results of the optimal design of the bracket welded and loaded at the free end through the use of DEa are carried out both in the case of welding on two and four sides and are shown in the Tables 9 and 10:

Table 9. Design values and objective function (case of 2-sided Welded)

<b>h</b>		<b>l</b>		<b>t</b>		<b>b</b>		<b>cost</b>
in	mm	in	mm	in	mm	in	mm	
0.423	10.744	2.658	67.513	7.084	179.934	0.335	8.509	2.427

Table 10. Design values and objective function (case of 4-sided welded)

<b>h</b>		<b>l</b>		<b>t</b>		<b>b</b>		<b>cost</b>
In	mm	in	mm	in	mm	in	mm	
0.234	5.944	2.500	63.500	7.084	179.934	0.335	8.509	2.034

Figures 10, 11, 12, 13, 14 show how the best individual of each variable varies at each generation until it reaches its optimum value. These figures also take into account the different ways in which it was made the welding of the beam, the average value of each variable at every generation in order to evaluate the difference between the two cases.

Figure 10 shows the variation of OF during the generation and for both cases of a beam welded on two and on four sides. This last figure also shows the variation of the OF average value.

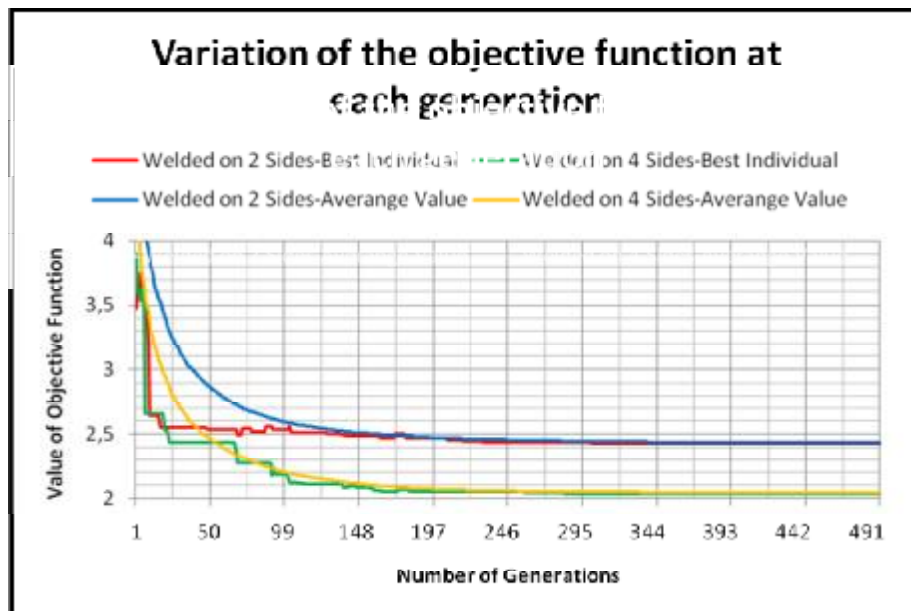


Figure 10. Variation of the objective function at each generation

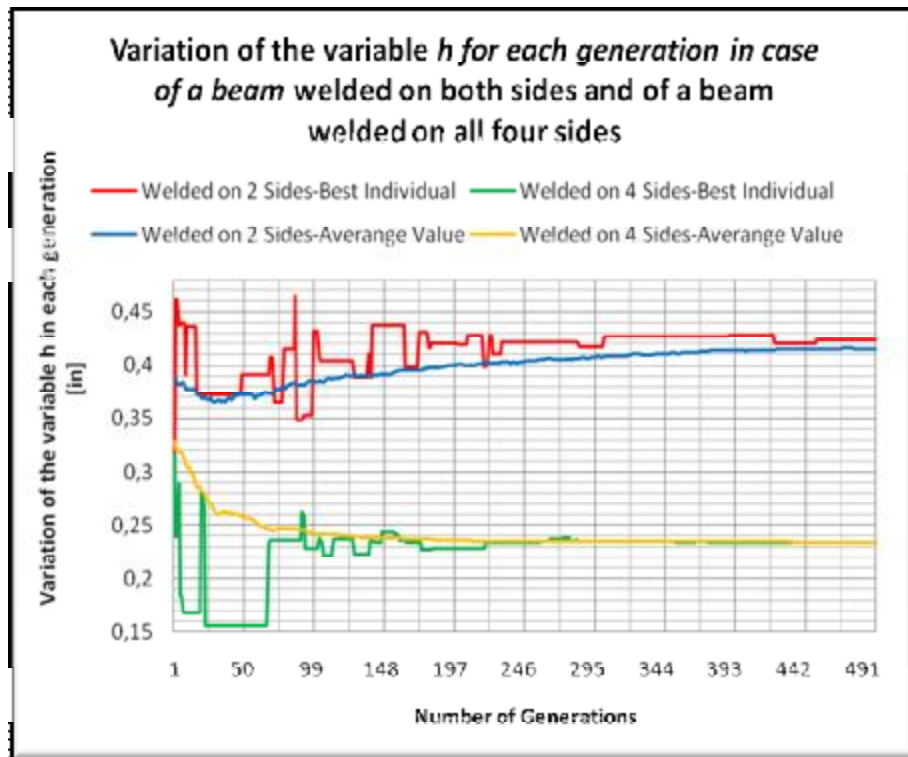


Figure 11. Variation of the variable  $h$  for each generation in case of a beam welded on both sides and of a beam welded on all four sides

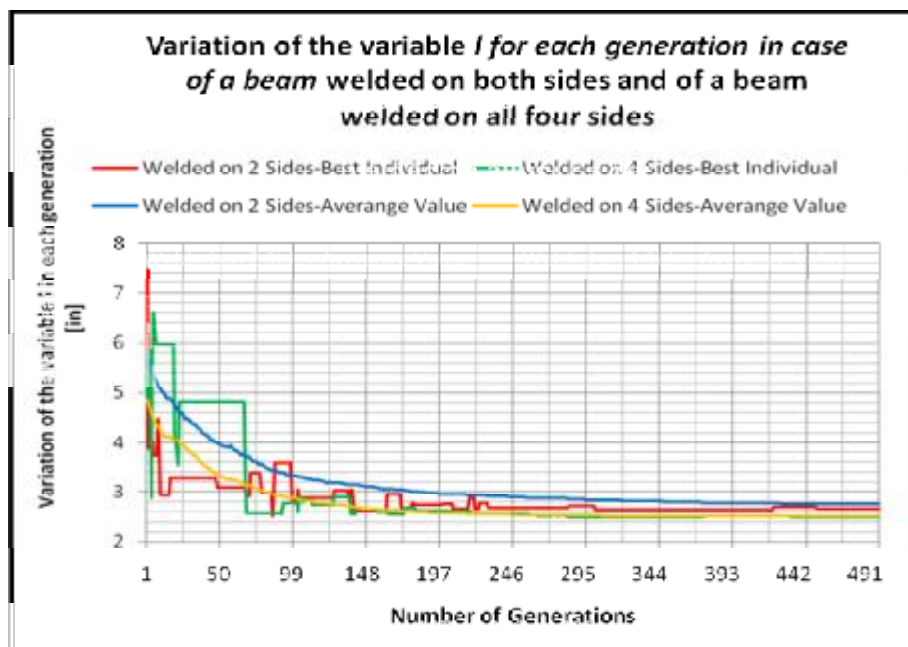


Figure 12. Variation of the variable  $l$  for each generation in case of a beam welded on both sides and of a beam welded on all four sides

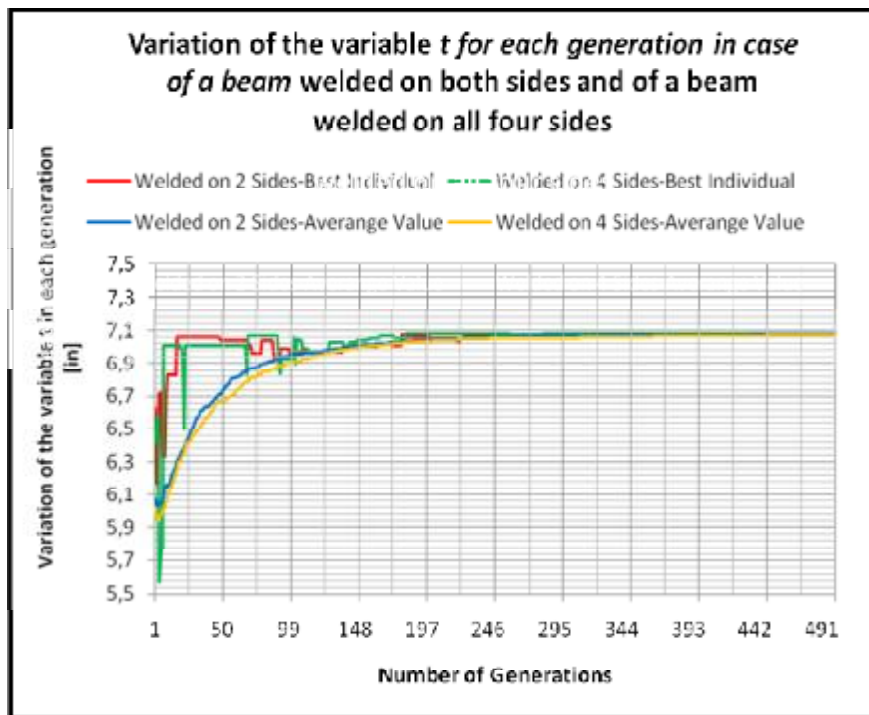


Figure 13. Variation of the variable  $t$  for each generation in case of a beam welded on both sides and of a beam welded on all four sides

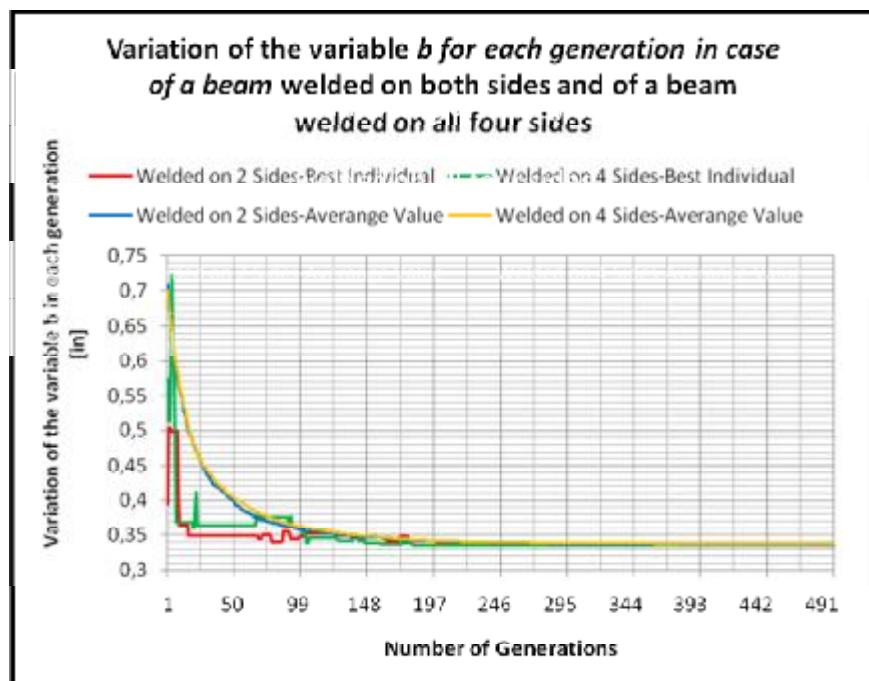


Figure 14. Variation of the variable  $b$  for each generation in case of a beam welded on both sides and of a beam welded on all four sides

Also in this second analyzed problem, the DEA has achieved the best result in short time and with an accurate precision. Tables 9 and 10 show the results obtained for the case of a beam welded on two sides and for the case of a beam welded on all four sides.

By comparing the values obtained, it is clear that for the beam welded on all four sides, the dimension of the height of welding ( $h$ ) is halved to about 44.7%, the size of the weld length ( $l$ ) decreases modestly by approximately 5.9%, while the size of the beam section welded remains unchanged. Figures 10-14 show the evolution of the best individual and of the average value at each generation and the objective function.

Therefore, through this study it is possible to deduce that for a structural design of a steel section in a short time it is possible to apply Differential Evolution Algorithms. This method demonstrates how the effective solution of structural optimization problems by DEa allows the design of steel elements at the lowest cost, since the solution is characterized by the minimum required size and then by a small weight, while always respecting the constraints and boundary conditions.

#### 4. CONCLUSIONS

This paper has been aimed to understand the efficiency of Differential Evolution Algorithm for solving problems of structural optimization by means of two simple but very significant cases of study referred to steel elements. This approach is used since it appears more user-friendly than that other Evolutionary Algorithms such as Genetic Algorithm (GA). Two main positive aspects must be considered: first, the functionality of the DE Algorithm is based on operators simpler than those advanced of GA. In addition, it requires only a small set of control parameters (usually the total number of control parameters is significantly lower than those adopted by the GA) so that the phase of preparation requires only a low time and it is more practical for non-experts in the field of Soft Computing techniques.

The DE Algorithm proposed is here developed for solving steel structures optimization problems in a practical way. It is organized by an approach which limits the number of parameters to be tuned by final operators; moreover, it uses a specific constraints handling technique, that assures the final solution is feasible.

The proposed algorithm is applied to some practical cases to verify its appeal and simplicity, more than its effectiveness.

#### REFERENCES

1. Eiben A, Smith JE, *Introduction to Evolutionary Computing*, Natural Computing Series, Springer-Verlag, 2003.
2. Mallipeddi R, Mallipeddi S, Suganthan P. Ensemble strategies with adaptive evolutionary programming, *Information Sciences*, 2010;**180**(9):1571-81.
3. Michalewicz Z, Schoenauer M. Evolutionary algorithms for constrained parameter optimization problems, *Evol Comput* 1996;**4**(1):1-32.
4. Coello Coello CA. Theoretical and numerical constraint handling techniques used with

evolutionary algorithms: a survey of the state of the art, *Comput Methods Appl Mech Eng* 2002;**191**(11–12):1245-87.

5. Storn RM, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim*, 1997;**11**(4):341-59.
6. Rechenberg I. Cybernetic Solution Path of an Experimental Problem. Royal Aircraft Establishment, Library Translation 1122, Farnborough, Hants, U.K, 1965.
7. Schwefel H P. Kybernetische Evolution als Strategie der experimentellen Forschung in der Str omungstechnik Diploma thesis Technische Universit at Berlin, 1965.
8. Holland JH. *Adaptation in Natural Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.
9. Fogel LJ, Owens AJ, Walsh MJ. *Artificial Intelligence through Simulated Evolution*. John Wiley & Sons, New York, 1966.
10. Smith AE, Coit DW. Constraint handling techniques—penalty functions, in: T. Bäck, D.B. Fogel, Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*, Oxford University Press, Institute of Physics Publishing, 1997. pp. C 5.2:1-C 5.2:6.
11. Runarsson TP, Xin Yao. Stochastic ranking for constrained evolutionary optimization, *IEEE Trans Evol Comput* 2000;**4**(3): 284-94.
12. Mezura-Montes E, Miranda-Varela ME, Gómez-Ramón R. del C. Differential evolution in constrained numerical optimization: An empirical study, *Information Sciences*, 2010; **180**: 4223-62.
13. Surry PD, Radcliffe NJ. The COMOGA method: constrained optimisation by multi-objective genetic algorithms. *Control Cybern* 1997;**26**(3): 391-412.
14. Cai ZX, Wang Y. A multiobjective optimization-based evolutionary algorithm for constrained optimization. *IEEE Trans Evol Comput* 2006;**10**(6): 658-75.
15. Venter G, Haftka R. Constrained particle swarm optimization using a bi-objective formulation. *Struct Multidisc Optim* 2009; **40**(1-6): 65-76.
16. Deb K. An efficient constraint handling method for genetic algorithms. *Comput Methods Appl Mech Eng*, 2000;**186**(2–4): 311–38.
17. Runarsson TP, Xin Yao. Stochastic ranking for constrained evolutionary optimization, *Evolutionary Computation*, *IEEE Transactions* 2000;**4**(3): 284-94.
18. Takahama T, Sakai S. Constrained optimization by applying the alpha constrained method to the nonlinear simplex method with mutations. *IEEE Trans Evol Comput* 2005;**9**(5):437-51.
19. Koziel S, Michalewicz Z. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. *Evol Comput* 1999;**7**(1):19-44
20. Price K, Storn R, Lampinen J. *Differential Evolution: A Practical Approach to Global Optimization*, Natural Computing Series, Springer-Verlag, 2005.
21. Nelder JA, Mear R. A simplex method for function minimization, *Comput J* 1965;**7**(4):308-13.

## LIST OF SYMBOLS

$\mathbf{x} = \{x_1, \dots, x_j, \dots, x_D\}$  -- design vector  
 $f(\mathbf{x})$  -- objective function

$g_i(\mathbf{x})$  --inequality constraint

$h_i(\mathbf{x})$  --equality constraint

$\mathbf{x}^l$  -- lower bound of design vector

$\mathbf{x}^u$  --upper bound of design vector

$\{x_{i,0} = x_{1,i,0}, x_{1,i,0}, \dots, x_{D,i,0} \mid i = 1, 2, \dots, NP\}$  --initial population

$x_j^l$  --lower bound of  $j^{\text{th}}$  element of design vector

$x_j^u$  -- upper bound of  $j^{\text{th}}$  element of design vector

$\{x_{i,g} = x_{1,i,g}, x_{1,i,g}, \dots, x_{D,i,g} \mid i = 1, 2, \dots, NP\}$  -- current parent population

$v_{i,g}$  - mutation vector

$r_0, r_1$  and  $r_2$ --integers

$x_{r_0,g}$  -vector of current parent population

$F_i$  -- mutation factor

$x_{r_1,g} - x_{r_2,g}$  --difference vector

$x_{i,g}$  --parent vector

$x_{\text{best},g}$  --best vector at the current generation  $g$

$v_{j,i,g}$  -- $j^{\text{th}}$  components of the mutation vector  $v_{i,g}$  at generation  $g$

$x_{j,i,g}$  -- $j^{\text{th}}$  components of the parent vector  $x_{i,g}$  at generation  $g$

$u_{i,g} = (u_{1,i,g}, u_{1,i,g}, \dots, u_{D,i,g})$  -- final trial vector

$\text{rand}_j(a, b)$  -- uniform random number on the interval  $(a, b]$  and newly generated for each  $j$

$j_{\text{rand}} = \text{rand int}_j(1, D)$  --integer randomly chosen from 1 to  $D$  and newly generated for each  $i$

$CR_i \in [0, 1]$  -- crossover probability

$u_{i,g}$  -- trial vector

$\Phi(x_{i,g})$  --violation function

$f_y$  -- steel yield stress

$q$  --uniformly distributed load

$g_g$  --permanent load safety coefficient

$G_k$  --characteristic permanent load

$g_q$  -- accidental load safety coefficient

$Q_k$  --accidental permanent load

$w$  -- weight of the beam

$g$  --specific load of the steel



$L$  --length of the beam

$A_f$  -- area of the flange of the beam

$A_w$  -- area of the web

$t_f$  --thickness of the wing

$b_f$  --width of the wing

$t_w$  -- thickness of the web

$h_w$  -- height of the web

$b_f^l$  --lower bound of width of the flange

$b_f^u$  --upper bound of width of the flange

$h_w^l$  --lower bound of the height of the web

$h_w^u$  -- upper bound of the height of the web

$t_f^l$  -- lower bound of the thickness of the flange

$t_f^u$  --upper bound of the thickness of the flange

$h_w^l$  --lower bound of the height of the web

$h_w^u$  --upper bound of the height of the web

$M_{ed}$  --calculus bending moment

$M_{c,rd}$  --allowable bending moment

$V_{ed}$  --calculus shear

$V_{i,pl,Rd}$  --allowable shear

$M_{b,rd}$  --design buckling resistance moment

$d_{lim}$  --limit displacement

$W_{pl}$  --plastic moment

$g_{M0}$  --safety coefficient

$A$  --total section area

$A_{v,i}$  --shear area

$c_{LT}$  --reduction factor

$M_{cr}$  --elastic critical moment

$g_{M1}$  --safety factor for the instability

$b = 0.75$ .

$\bar{I}_{LT}$  - lateral -torsion slenderness coefficient

$I_{LT,0} = 0.4$

$a_{LT}$  --imperfection factor

$C_1$  --coefficient which takes into account of distribution of moment

- $E$  --elastic modulus  
 $k, k_w$  --effective length coefficient.  
 $I_w$  --warping constant  
 $I_z$  --inertia moment around the minor axis  
 $G$  --shear modulus  
 $I_t$  --torsion constant  
 $\tau$  --shear tension  
 $\sigma$  --tension due to bending moment  
 $d$  --beam bending deflection  
 $P_c$  --critical load  
 $b$  --width of the beam  
 $t$  --height of the beam  
 $h$  --height of the weld bead  
 $l$  --length of the cord itself  
 $c$  --cost function  
 $c_1$  -- unit cost for volume of welding material  
 $c_2$  --is the unit cost for volume of the bar  
 $L$  --distance from the load application  
 $h^l$  --lower bound of  $h$   
 $h^u$  --upper bound of  $h$   
 $l^l$  --lower bound of  $l$   
 $l^u$  --upper bound of  $l$   
 $t^l$  --lower bound of  $t$   
 $t^u$  --upper bound of  $t$   
 $b^l$  --lower bound of  $b$   
 $b^u$  --upper bound of  $b$   
 $t_{\max}$  --maximum shear stress  
 $S_{\max}$  --maximum normal stress  
 $d_{\max}$  --maximum displacement  
 $P_c(\mathbf{x})$  --critic load