Noname manuscript No.
(will be inserted by the editor)

Structural Design using Multi-objective Metaheuristics. Comparative Study and Application to a Real-World Problem

Gustavo Zavala · Antonio J. Nebro · Francisco Luna · Carlos A. Coello Coello

Received: date / Accepted: date

Abstract Many structural design problems in the field of civil engineering are naturally multi-criteria, i.e., they have several conflicting objectives that have to be optimized simultaneously. An example is when we aim to reduce the weight of a structure while enhancing its robustness. There is no a single solution to these types of problems, but rather a set of designs representing trade-offs among the conflicting objectives. This paper focuses on the application of multi-objective metaheuristics to solve two variants of a real-world structural design problem. The goal is to compare a representative set of state-of-the-art multi-objective metaheuristic algorithms aiming to provide civil engineers with hints as to what optimization techniques to use when facing similar problems as those selected in the study presented in this paper. Accordingly, our study reveals that MOCell, a cellular genetic algorithm, provides the best overall performance, while NSGA-II, the *de facto* standard multi-objective metaheuristic technique, also demonstrates a competitive behavior.

Keywords Multi-objective optimization · metaheuristics · structural optimization · real-world problems

Gustavo R. Zavala
Khaos Research Group
University of Málaga
Spain
E-mail: grzavala@gmail.com

Antonio J. Nebro
Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga
Spain
E-mail: antonio@lcc.uma.es

Francisco Luna
Departamento de Sistemas Informáticos y Telemáticos, Centro Universitario de Mérida
Universidad de Extremadura
Spain
E-mail: fluna@unex.es

Carlos A. Coello Coello
CINVESTAV-IPN
Departamento de Computación
México
E-mail: ccoello@cs.cinvestav.mx

1 Introduction

The design of structures such as bridges, buildings, machinery, or any other item that supports or resists loads is a complex problem within the field of civil engineering, where structural integrity affects the safety and functionality. As usually happens in this and other disciplines, structural design problems require several conflicting objectives to be optimized, such as minimizing the total investment cost while maximizing the safety of the final structure.

Problems with more than one objective function to be optimized are known as Multi-objective Optimization Problems (MOPs), and their main characteristic is the non existence of a single solution that can optimize all the objectives at the same time. Instead, the solution to these problems consists of a set of alternative trade-off solutions. This set is usually referred to as *Pareto set* or *Pareto front*, depending on whether it refers to the space of the decision variables defining the problem (solution space) or to the space of the objective function values (objective space). Solutions within these sets are said to be non-dominated, as no solution among them is better than the others for all the objective functions. In we focus on the context of structural design, the implication is that there is no single design that minimizes the structure's weight (to reduce the investment cost as much as possible) and, at the same time, maximizes the stiffness (to provide the highest

possible degree of safety). Instead, there is a set of designs providing a trade-off between these two conflicting optimization criteria.

Addressing a MOP basically consists of two different phases. First, it requires computing the Pareto front of that problem, which in most cases is impractical principally for two reasons: (1) it may contain an infinite number of non-dominated solutions; and (2) there are MOPs that are NP-hard (Knowles and Corne 2003; Lust and Teghem 2010) and, therefore, there is no polynomial time algorithm to solve them. In this situation, the goal when solving a MOP is usually to produce a set of solutions that constitute a reasonably good approximation of the true Pareto front of the MOP being solved. Second, the computed approximation set is given to the decision maker, who is the expert in charge of selecting the final design of the structure to be adopted as the solution. The decision maker selects one solution (or several) from the approximation set in such a way that it best fits his/her desired requirements and preferences. While this second phase clearly depends on user preferences and on other external conditions, computing a high quality approximation is a real challenge in the context of complex problems, as is the optimization of the design of civil structures.

Metaheuristics (Glover and Kochenberger 2003) are a class of approximation algorithms which have become popular alternatives for solving multi-objective optimization problems. For example, Evolutionary Algorithms (EAs) (Bäck et al 1997), which are a specific type of metaheuristic, are highly suitable for solving MOPs due to their ability to compute an approximation to the Pareto front in a single run of the algorithm, mainly because of their population-based nature.

In spite of the popularity of multi-objective metaheuristics, a recent survey on multi-objective optimization applied to structural design (Zavala et al 2013) revealed that the application of metaheuristic techniques in this field has been scarce until very recently. Specifically, their application has been mainly restricted to multi-objective evolutionary algorithms (MOEAs) such as the Nondominated Sorting Genetic Algorithm-II (NSGA-II) (Deb et al 2002) and the Strength Pareto Evolutionary Algorithm 2 (SPEA2) (Zhang and Li 2007), which are two of the most popular MOEAs in the literature. It is worth noting, however, that NSGA-II and SPEA2 date back to 2000 and there have been significant developments since then. For example, new metaheuristic techniques, evolutionary and non-evolutionary based, have been proposed but rarely have they been applied to structural design problems.

Furthermore, the second major conclusion of the aforementioned survey is that many of these previous studies were based either on benchmark problems or on tiny instances of real-world problems, real world instances being largely ignored (see for example (Kaveh and Laknejadi 2011; Kelesoglu 2007)). In many cases, the instances adopted to validate multi-objective metaheuristics do not have any connection with real world applications (see for example (Zapotecas Martínez and Coello Coello 2014)) and, therefore, the conclusions obtained in these studies may have little (or no) relevance to civil engineers responsible for designing real structures.

The idea of this paper is to bridge the existing gap between multi-objective metaheuristics and their application to real world structural design problems. To do so, we use two instances of a real-world problem for the case study: the dimensioning of two cable-stayed bridges. Then, we carry out a study to solve these two instances by applying seven multi-objective metaheuristics which are representative of the state-of-the-art. The selected techniques include both classic and recent algorithms. Our goal is to provide civil engineers with experimental insights regarding the solver which seem promising in this case study, analyzing the advantages and disadvantages of the algorithm adopted in each case. In addition, when comparing these techniques, we follow a rigorous methodology to measure the quality of the approximations computed by the different techniques and to statistically validate the differences between them.

The full list of the scientific contributions of this work can therefore be summarized as follows:

1. To solve two real world instances of a multi-objective structural design problem.
2. Application of a wide set of representative multi-objective metaheuristics, based on different working principles.
3. Detailed and accurate performance comparison between different metaheuristic algorithms
4. Analysis of the obtained solutions from a civil engineering point of view.

The rest of this paper is organized as follows. The next section provides a short review of the preceding related work. Readers interested in a thorough review of this topic, should refer to Zavala et al (2013), where a detailed and comprehensive description of multi-objective techniques for solving structural design problems is presented. Section 3 is devoted to formally describe the problem tackled in this paper. Some background to multi-objective optimization is explained in Section 4. An introduction to multi-objective metaheuristics as well as the description of the algorithms used in our study are included in the next section. Section 6 de-

scribes in detail, the experiments conducted, with an emphasis on the description of the applied metaheuristics and the quality indicators adopted. The results obtained are analyzed in Section 7. Finally, the main conclusions of this paper and some of the possible lines of work for the future are given in Section 8.

2 Related Work

In the survey of multi-objective metaheuristics applied to structural optimization presented in Zavala et al (2013), 51 references were reported and analyzed. With the exception of one, all of these references have been published since the year 2000. The survey classifies structural problems according to the next two criteria:

- Bar or element design
  - Area optimization (1.x)
  - Size optimization (2.x)
  - Shape optimization (3.x)
  - Topological cross-section optimization (4.x)
- Topological design
  - No topological optimization (y.1)
  - Discrete topological optimization (y.2)
  - Continuous topological optimization (3)

Note that a compact notation has been defined (typed at the end of each entry of the previous list) so that each problem type is assigned a code , [X.]Y, which shows the problem category under the first (X) and second (Y) criteria, respectively (please note that the first part is optional as in continuous topological optimization problems the bar or element design does not hold).

The kind of structural problem we are considering in the work presented here fits into category (2.1), i.e., we are concerned with the optimization of the size of the elements of the structure, i.e., the dimensioning of the structure, with no topological optimization.

The analysis carried out in the survey indicated that 17 of the 84 problems found, fall into category 2.1. An in-depth look at the algorithms used to address these problems show that NSGA-II was the most widely used algorithm (in 6 out of 15 papers) and there is a wide spread use of other techniques: SPEA2 (4), the Pareto Archived Evolution Strategy (PAES) (Knowles and Corne 2000) (1), several ad-hoc Genetic Algorithms (4), Particle Swarm Optimization (PSO) (1), Tabu Search (TS) (1), Simulated Annealing (SA) (1), the Population-Based Incremental Learning (PBIL) (Baluja 1994) (1), etc.

One of the main conclusions of the aforementioned survey was that most of the applied algorithms (NSGA-II, SPEA2, PAES) were proposed about fourteen years ago (i.e., around the year 2000) and, therefore, the significantly large amount of research on multi-objective evolutionary algorithms carried out since then has been partially ignored in the papers reviewed in the survey. Furthermore, it also found an evident lack of rigorous comparative studies of state-of-the-art techniques in the papers dealing with the solution of structural problems (including the 2.1 category) using MOEAs. As a consequence, a civil engineer interested in solving a particular problem would not have an up-to-date reference that could help guide her/him towards the most suitable algorithm (or at least, the most promising approaches) to use in a specific application.

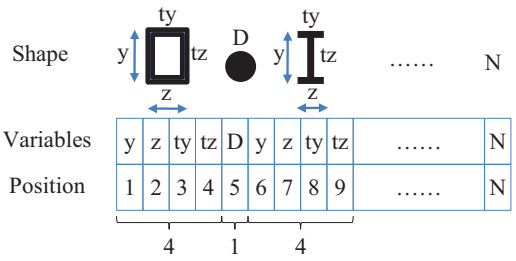


Fig. 1 Types of cross-section elements (hollowed rectangle, circular, I-beam) and parameters to be optimized.

The work presented here, aims to contribute in this way, by studying the optimization of two variants of a cable-stayed bridge, using seven different multi-objective metaheuristics. To the best of our knowledge, no study of this kind has been previously reported in the specialized literature. Furthermore, a closer look at the information in Zavala et al (2013) about the MOEAs that have been applied in the field indicates that basic data about the parameters settings of each technique is, in many cases, missing. This turns the replication of any of these studies into an almost impossible task. In this paper, we provide not only detailed information about the algorithms adopted, but we also have made their source codes available in the public domain, through the website: `http://ebesjmetal.sourceforge.net`.

3 Problem Description

The target problem of our work is the dimensioning of two cable-stayed bridges, formed by elements with different cross-section shapes and sizes. Each element can have one out of three different cross-sections: box-shaped girder (hollowed rectangle), I-shaped girder (I-beam), and circular shape. Each of these element types feature several geometric parameters that need to be determined for each element composing the bridge in order to reach an optimal design (see Figure 1).

We assume the hypothesis that the total bridge load is composed by both invariable and variable loads. The former are the linearly distributed loads formed by the weights of the road and the rolling plate surface, and the effect of the side wind on the suspended main beams; the latter are due to the weight of the elements, which depends on the particular geometric measurements that are subject to the optimization process.

The numerical model for analyzing the structures is the direct stiffness method also known as the matrix stiffness method (Turner et al Setember 1956). This model is based on the use of the first-order conventional linear elastic stiffness matrix, so that the forces and deformations are linear. The second order effects include the geometric stiffness matrix (Przemieniecki 1968) and (Yang and McGuire 1986), which considers the effects of axial load on the bending stiffness of a member. The effect of buckling can also be considered in the calculation.

Below, we detail the features of the two bridges considered in our study.

3.1 Target Problem: 25N_35E Bridge

The smallest of the two problems is called 25N_35E, as it is composed of 25 nodes and 35 elements (see Fig. 2). The bridge has two columns supporting part of the weight of the deck, and the rest of the board weight is supported by the parallel tension of the cables.

It is an asymmetric bridge with a one-way road circulation lane, composed of a single tranche, a total length of $9.00m$ and the deck length and width are $6.00m$ and $3.50m$, respectively. In addition, the bridge contains two pillars (columns) $3.50m$ in height, anchored by two cables to an upper beam. The main longitudinal beam, made of malleable steel, is suspended using ten high resistant tensioners, so two materials with different elastic properties are used. The calculation takes into consideration the entire bridge and considers that the loads are asymmetric.

For the sake of simplifying the construction, we have configured groups of elements, in such a way that the elements composing the groups have the same shape, material and similar position in the structure. The 25N_35E bridge has 8 groups of elements, as shown in Figure 2.

After applying this grouping strategy, the total number of variables to optimize is 26, the total number of geometric constraints is 24, and the sum of mechanical and deflection constraints are 24 and 20, respectively, as summarized in Table 1 (the description and formulation of the constraints are in Section 3.4).

3.2 Target Problem: 133N_221E Bridge

The second problem, referred to as 133N_221E (133 nodes and 221 elements), has a one-way road and a pedestrian circulation lane which is illustrated in Figure 5. The bridge has a total length of $44.00m$ and the deck length and width are $32.00m$ and $6.40m$, respectively. In addition, the bridge contains four pillars $6.00m$ high joined at the top and anchored by cables.

The main longitudinal beam, made of malleable steel, is suspended using ten high resistant tensioners, so two materials with different elastic properties are used like in the 25N_35E instance. From a mechanical point of view, we consider only half of the bridge and symmetric loads. For the sake of simplifying the construction, the grouping strategy has been again used: the 133N_221E bridge has 33 groups of elements, as shown in Figure 3. It results in 108 decision variables, 100 geometric constraints, and the sum of mechanical and deflection constraints are 99 and 68, respectively (see Table 2).

In the middle of the bridge there is a structural joint that allows a controlled descent of up to 0.10m between the two symmetric parts. These two symmetric parts work independently up to this measure and, beyond it, they work together. By assuming that the horizontal and vertical loads are symmetric, the calculation of the structure is also symmetric.

Figure 4 shows a schematic view of the cross and longitudinal sections of the bridge 133N_221E. We can see that the main parts of the structure have variable measurements, which correspond to the measurements of the element shapes (length, width, and thickness) to be optimized.

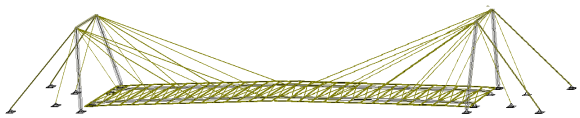


Fig. 5 Bridge 133N_221E.

3.3 Objective Functions to be Optimized

In both cases, the optimization problems are formulated as having two different objective functions: the first (f_1) is to minimize the total weight of the structure and the second (f_2) is to minimize the summation of the deformations in some selected nodes. Given a structure design composed of B elements and D nodes where the deformation has to be checked, the two objective functions to be optimized can be formally defined as follows:

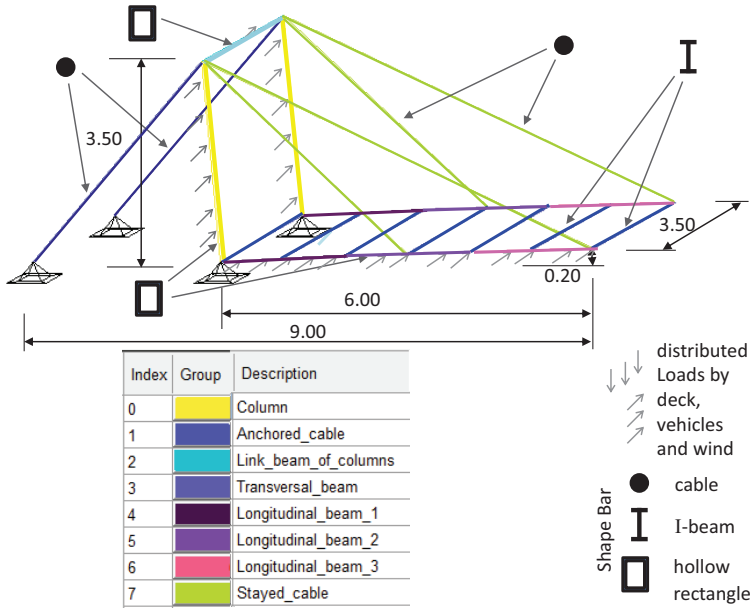


Fig. 2 Bridge 25N_35E and element grouping.

Shape	Elements	Groups (gr)	Variables (var)	Geometric constraints	Mechanic constraints	Deflection constraints
Circle	6	2	1var/gr x 2gr = 2		3const/gr x 2gr = 6	
I-beam	14	1	4var/gr x 1gr = 4	4const/gr x 1gr = 4	3const/gr x 1gr = 3	20
Hollowed rectangle	15	5	4var/gr x 5gr = 20	4const/gr x 5gr = 20	3cont/gr x 5gr = 15	
Total	35	8	26	68		

Table 1 Variables and constraints of the 25N_35E bridge.

Shape	Elements	Groups (gr)	Variables (var)	Geometric constraints	Mechanic constraints	Deflection constraints
Circle	46	8	1var/gr x 8gr = 8		3const/gr x 8gr = 24	
I-beam	39	4	4var/gr x 4gr = 16	4const/gr x 4gr = 16	3const/gr x 4gr = 12	68
Hollowed rectangle	136	21	4var/gr x 21gr = 84	4const/gr x 21gr = 84	3cont/gr x 21gr = 63	
Total	221	33	108	267		

Table 2 Variables and constraints of the 133N_221E bridge.

$$f_1 = \Sigma_{i=1}^{B} \gamma_i l_i \Omega_i \quad (1)$$

$$f_2 = \Sigma_{j=1}^{D} \delta_j \quad (2)$$

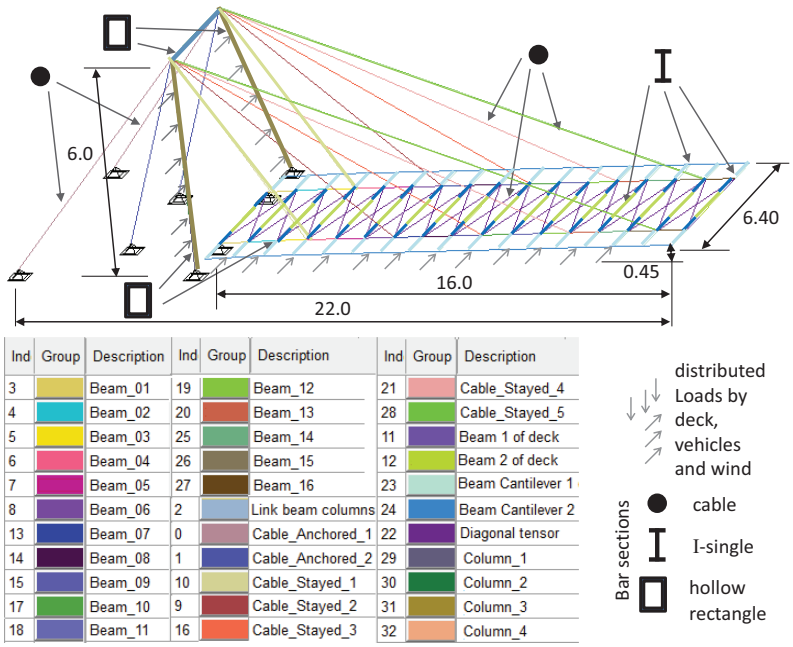
The first equation states that the bridge weight consists of the summation of the weight of each element i, which is computed according to its length (l_i), specific weight (γ_i), and cross-section area Ω_i. The second one indicates that the deformation on the selected nodes is computed as the summation of the allowable deflection (δ_j) for each node j.

3.4 Problem constraints

The side constraints that must be satisfied to obtain acceptable dimensions for element cross sections fall into three categories:

- Geometric proportions.
- Mechanical properties of the materials (strength), which will determine the admissible stresses through the permissible elastic resistance.
- The allowable deformations (deflection) of the structure at predefined nodes.

The geometrical constraints are four, and are defined in Equations (3) to (6). They represent, respectively, the maximum ratio between height and width, the minimum ratio between height and width, the ra-



Ind	Group	Description	Ind	Group	Description	Ind	Group	Description
3		Beam_01	19		Beam_12	21		Cable_Stayed_4
4		Beam_02	20		Beam_13	28		Cable_Stayed_5
5		Beam_03	25		Beam_14	11		Beam 1 of deck
6		Beam_04	26		Beam_15	12		Beam 2 of deck
7		Beam_05	27		Beam_16	23		Beam Cantilever 1
8		Beam_06	2		Link beam columns	24		Beam Cantilever 2
13		Beam_07	0		Cable_Anchored_1	22		Diagonal tensor
14		Beam_08	1		Cable_Anchored_2	29		Column_1
15		Beam_09	10		Cable_Stayed_1	30		Column_2
17		Beam_10	9		Cable_Stayed_2	31		Column_3
18		Beam_11	16		Cable_Stayed_3	32		Column_4

Fig. 3 Bridge 133B_221E and grouping of its elements.

tio between width and plate thickness, and the ratio between height and plate thickness:

$$Constr1_{i=1}^{G} = +\frac{Y_i}{Z_i} - 0.75Ryz_i \quad (3)$$

$$Constr2_{i=1}^{G} = -\frac{Y_i}{Z_i} + 1.50Ryz_i \quad (4)$$

$$Constr3_{i=1}^{G} = Max(-15tb_i + Y_i, 15tb_i - Y_i) \quad (5)$$

$$Constr4_{i=1}^{G} = Max(-10ta_i + Z_i, -20ta_i + Z_i) \quad (6)$$

The terms of all the equations that define the constraints are included in Table 3.

All bridges are subject to primary shear and bending moment effects as well as vertical deflections and end rotations. The constraints related to the strength of materials are defined in Equations (7) and (8) (primary normal stresses, as illustrated for the I-beam elements in Figure 6), and Equation 9 (primary shear and torsional stresses):

$$Constr5_{i=1}^{G} = \sigma_{-} = (-\omega_i\frac{Nxx_i}{\Omega_i} - Mxz_i\frac{Y_i}{Iz_i} - Mxy_i\frac{Z_i}{Iy_i}) + \sigma_{ac} \quad (7)$$

$$Constr6_{i=1}^{G} = \sigma_{+} = -(\frac{Nxx_i}{\Omega} + Mxz_i\frac{Y_i}{Iz_i} + Mxy_i\frac{Z_i}{Iy_i}) + \sigma_{as} \quad (8)$$

$$Constr7_{i=1}^{G} = \tau = -(\frac{Qxy_i\ Az_i}{Z_i\ Iz_i} + \frac{Qxz_i\ Ay_i}{Y_i\ Iy_i} + Mxx_i\frac{Z_i}{It_i}) + \tau_a \quad (9)$$

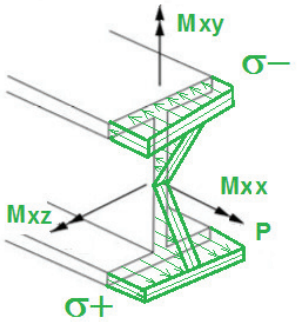


Fig. 6 I-beam total normal stress.

Finally, there is also a deflection constraint, which is specified in Equation (10):

$$Constr8_{j=1}^{D} = -\delta_j^n + \delta\ a_j^n \quad (10)$$

So as to provide an illustrative example, Figure 7 shows two displacement restricted nodes for the 133N_221E bridge.

Term	Description
G	number of element groups
Y_i	section height in y-axis for the elements in the i^{th} group
Z_i	section height in z-axis for the elements in the i^{th} group
tb_i	plate thickness in y-axis
ta_i	plate thickness in z-axis
Ryz_i	constant value relating the proportions of the geometrical shape (defined by design)
σ_{as}	allowable stress
σ_{ac}	allowable compression
τ_a	allowable shear stress
ω_i	buckling coefficient
$\pm Nxx_i$	maximum and minimum axial force
$\pm Mxz_i$	maximum and minimum bending moment with respect to the z-axis
$\pm Mxy_i$	maximum and minimum bending moment with respect to the y-axis
Iz_i	inertia moment with respect to the z-axis
Iy_i	inertia moment with respect to the y-axis
Qxy_i	shear force with respect to the y-axis
Qxz_i	shear force with respect to the z-axis
Az_i	statical moment with respect to the z-axis
Ay_i	statical moment with respect to the y-axis
It_i	a torsional moment with respect to the x-axis
j	constraint number for selected nodes
D	maximum number of nodes with constraint
δ_j^n	deflection in j^{th} node belonging to position n
$\delta\ a_j^n$	maximum allowed deflection j^{th} node belonging to position n

Table 3 Description of the terms in Equations (3) to (6), (7)- (9), and (10)

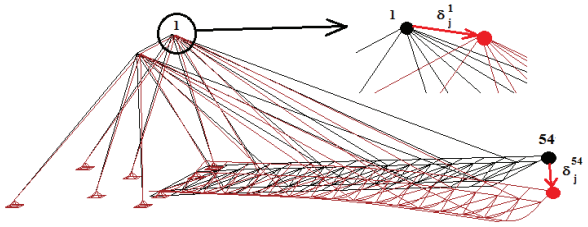


Fig. 7 Sample of displacement constraint nodes for Bridge 133N_221E

4 Multi-objective Optimization Background

The bridge design problem we are dealing with is composed of two contradictory objectives that have to be optimized at the same time, so it belongs to the discipline known as multi-objective optimization. This section is devoted to presenting some background to this field.

In a formal way, the formulation of a multi-objective optimization problem (MOP) extends the classic definition in single-objective optimization by considering the existence of a vector of (at least two) objective functions, as follows:

Definition 1 (MOP) A MOP is defined as a tuple (S, f, g, h), where $S \neq \emptyset$ is called the solution space (or search space), $f = [f_1, f_2, \ldots, f_k]$ is a vectorial function, where $f_i : S \to \mathbb{R}$, are the objective functions, and $g = [g_1, g_2, \ldots, g_m]$ and $h = [h_1, h_2, \ldots, h_p]$ are also vectorial functions, where $g_i : S \to \mathbb{R}$ and $h_i : S \to \mathbb{R}$ are the constraint functions. Thus, solving an optimization problem, consists in finding a set of solutions $X^* \subseteq S$ such that, for all $x^* \in X^*$:

$$f_j(x^*) \leq f_j(x), \quad \forall\ x \in S\ . \tag{11}$$

for some $1 \leq j \leq k$, subject to:

$$g_i(x^*) \leq 0 \quad i = 1, 2, \ldots, m \tag{12}$$
$$h_i(x^*) = 0 \quad i = 1, 2, \ldots, p \tag{13}$$

where $g_i, h_j : S \to \mathbb{R}$, $i = 1, ..., m$, $j = 1, ..., p$ are the constraint functions of the problem.

The set of all values satisfying the constraints defines the *feasible region* Ω and any point $\mathbf{x} \in \Omega$ is a *feasible solution*. For simplicity, in the following definitions we consider $\Omega = S$.

A key concept in muti-objective optimization is *Pareto dominance*, which is defined next:

Definition 2 (Pareto dominance) Given two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^k$, we say that $\mathbf{x} \leq \mathbf{y}$ if $x_i \leq y_i$ for $i = 1, ..., k$, and that $\mathbf{x}$ **dominates** $\mathbf{y}$ (denoted by $\mathbf{x} \prec \mathbf{y}$) if $\mathbf{x} \leq \mathbf{y}$ and $\mathbf{x} \neq \mathbf{y}$.

Pareto dominance can be used as a binary operator, in such as way that when applied to two solutions

in objective space it returns the solution that *dominates* the other one unless neither of the compared solutions dominate each other (i.e., when both they are *non-dominated* solutions). Many multi-objective metaheuristics apply Pareto dominance as a basic component of their search strategies.

Solving a MOP can be viewed as the process of finding the set of solutions that dominate every other point in the solution space; this means that the solutions in that set are Pareto optimal for that problem. Formally:

Definition 3 (Pareto Optimality) We say that a solution $x^* \in \mathcal{S}$ is **Pareto optimal** if it is non-dominated by any other solution $x' \in \mathcal{S}$.

The set of solutions composed of all the Pareto optimal solutions is known as the *Pareto optimal set* (or simply the Pareto set):

Definition 4 (Pareto Optimal Set) The **Pareto Optimal Set** $\mathcal{P}^*$ is defined by:

$$\mathcal{P}^* = \{x \in \mathcal{S} | x \text{ is Pareto optimal}\}$$

While the solutions in the Pareto set belong to the variable decision space (S), their correspondence in the objective function space ($\mathbb{R}^k$) leads to a set known as the *Pareto front*:

Definition 5 (Pareto Front) The **Pareto Front** $\mathcal{PF}^*$ is defined by:

$$\mathcal{PF}^* = \{f(x) \in \mathbb{R}^k | x \in \mathcal{P}^*\}$$

The solutions in the Pareto front are usually referred to as *non-inferior*, *acceptable*, or *efficient*. The Pareto front is also known in some contexts as *efficient frontier*.

As discussed in the introduction, the Pareto front of a given MOP could contain a large (even infinite) number of points so, in practice, we only aim for an approximation of the Pareto front containing a limited (predefined) number of points. Thus, an important goal of multi-objective optimization is to provide accurate approximations to the Pareto front. In this context, accurate means that the approximation set should fulfill two properties: be as close as possible to the Pareto front, to ensure that it contains optimal or quasi-optimal solutions, and be uniformly spread along the Pareto front, meaning that a good exploration of the search space has been carried out, without leaving unexplored areas of the Pareto front. These two properties are called *convergence* and *diversity*, respectively.

To assess the performance of multi-objective algorithms, quality indicators are applied to measure the degree of convergence and/or diversity of Pareto front approximations (Knowles et al 2006).

5 Metaheuristics and Multi-Objective Optimization

Metaheuristics are a family of non-exact optimization methods for finding high quality solutions to complex optimization problems which cannot be solved effectively by exact techniques. Although in general, metaheuristics do not guarantee optimal solutions will be found, they usually compute near-optimal solutions within a reasonable amount of time and resources. Examples of metaheuristic techniques are Evolutionary Algorithms (or EAs, which are by far the most well-known and widely used metaheuristics), and many others, such as Particle Swarm Optimization (PSO), Scatter Search, Simulated Annealing, Tabu Search, etc.

Algorithm 1 Template of a metaheuristic

```
1: P(0) ← GenerateInitialSolutions()
2: t ← 0
3: Evaluate(P(0))
4: while not StoppingCriterion( ) do
5:     Q(t) ← Variation(P(t))
6:     Evaluate(Q(t))
7:     P(t + 1) ← Update(P(t), S(t))
8:     t ← t + 1
9: end while
```

From a behavioral point of view, a metaheuristic typically follows an iterative process in which a set of tentative solutions are manipulated by a number of variation operators, aiming to progressively generate better solutions. Algorithm 1 shows the pseudo-code of a generic metaheuristic. In that code, a set P of some initial solutions (P may be eventually initialized to $\emptyset$), is iteratively updated by generating a set Q of new solutions from it until a stopping condition is met.

Algorithm 1 can be considered as a template that can be instantiated to yield most metaheuristic algorithms in current use. For example, in the case of EAs, the solutions and the set P are called, respectively, *individuals* and *population*, and the variation operators are crossover and mutation. If we focus on PSO, the solutions and the set P are referred to as *particles* and *swarm*, respectively, and the variation operators are equations that modify the position and velocity of the particles.

Metaheuristics have become very popular techniques for solving both single- and multi-objective optimization problems. In the latter case, one reason is that a Pareto front approximation can be obtained in a single run of the algorithm. Multi-objective metaheuristics also follow the template of Algorithm 1 incorporating a few extra components to deal with the fact that an approximation to the Pareto front, instead of a single

point, is aimed for. An example may be the use of an external archive to keep only the non-dominated solutions found during the search, which is a policy adopted by many state-of-the-art multi-objective solvers.

An important issue that must be considered in the multi-objective domain is that comparing solutions is not as trivial as in single-objective optimization. The dominance relationship provides a partial order, which makes it difficult to compare different solutions.

To cope with this issue, there are currently three main different approaches used by multi-objective metaheuristics:

1. *Pareto dominance + density estimator*. This is the most popular approach. The idea is to make use of Pareto dominance relationships to compare solutions and, when two solutions are non-dominated, a density estimator is applied to break the tie.
2. *Decomposition*. The multi-objective optimization problem is decomposed in a number of single-objective problems by applying an aggregation-based scheme. Algorithms based on this approach compare solutions based on the aggregations.
3. *Quality indicator based*. A current trend is to use a quality indicator to compare solutions. Therefore, a solution a is better than b if a provides a better indicator value than b.

In the work presented here we will use metaheuristics from the three categories: NSGA-II, PAES, MOCell, GDE3, and SMPSO belong to 1, while MOEA/D and SMS-EMOA fit into the categories 2 and 3, respectively. We briefly describe these techniques next.

5.1 Description of the Metaheuristics Considered

In this section, we briefly describe the algorithms we have used in our study. The criterion for selecting the techniques is to cover a range of representative techniques, covering both classic algorithms and modern techniques.

With regard to classical techniques, we have selected NSGA-II and PAES. The first one has become the *de facto* standard multi-objective metaheuristic, being the most widely used algorithm to date. In fact, this is the case for structural design problems as indicated before in the related work section. The reason for selecting PAES lies in its simplicity. This approach only requires the user to set a few control parameter which makes it easy to configure.

The next group of techniques comprises MOCell, SMPSO, and GDE3 as examples, respectively, of cellular evolutionary algorithms, PSO, and differential evolution. These metaheuristics have proven to be very effective in many studies.

Finally, as stated in the previous section, we have selected MOEA/D and SMS-EMOA as representatives of decomposition and quality indicator based algorithms, respectively.

We provide a description of all these algorithms in the following paragraphs (detailed information about them can be found in the included references).

The *Non-dominated Sorting Genetic Algorithm - II* (NSGA-II) was proposed in (Deb et al 2002). It works with a (parents) population of solutions which is used to create an offspring population in each iteration of the algorithm using the typical genetic operators (selection, crossover, and mutation). Both populations are merged at the end of each iteration and the former population is updated by considering the best solutions from this union. Pareto dominance is used in NSGA-II to rank the solutions, and a density estimator called *crowding distance* is applied to diversify the set of solutions generated by the algorithm.

The PAES (*Pareto Archive Evolution Strategy*) is a simple (1+1) evolution strategy (Knowles and Corne 1999), which uses a single individual that undergoes mutation (no crossover operator is adopted in this case). PAES uses an external archive where the non-dominated solutions found during the search are stored. To promote diversity within the archive, it uses an adaptive grid which partitions objective function space into several hypercubes. When a solution has to be discarded from the archive, higher priority is given to the hypercubes containing a higher number of solutions.

MOCell (Nebro et al 2006) is a multi-objective cellular genetic algorithm (cGA). The main characteristic of this type of algorithms is that each solution belongs to a cell and can only be recombined with a reduced number of solutions (the surrounding cells or neighbors). The main idea of this limitation is to perform a greater exploration of the search space. MOCell also considers an external archive to store non-dominated solutions. This archive is bounded and uses the crowding distance of NSGA-II to maintain a diverse set of solutions. We have used here an asynchronous version of MOCell, called aMOCell4 in Nebro et al (2007), in which the cells are updated sequentially (asynchronously).

SMS-EMOA (Beume et al 2007) (*S Metric Selection Evolutionary Multi-Objective Algorithm*) is an algorithm based on NSGA-II which adopts the hypervolume (Zitzler and Thiele 1999) quality indicator to select the best solutions. In the selection scheme of this approach, given two solutions, the one contributing the most to the hypervolume of the current population is preferred. SMS-EMOA can become a very slow algo-

rithm due to the high computational cost involved in the calculation of the hypervolume contribution.

The *Multi-Objetive Evolutionary Algorithm based on Decomposition* MOEA/D (Zhang and Li 2007) is the multi-objective metaheuristic that popularized the use of a decomposition of MOPs into several single-objective problems (subproblems) by using a scalarization approach. By combining this idea with the use of neighboring relationships between sub-problems, MOEA/D has been shown to be very competitive when solving difficult problems, and its version based on differential evolution (Li and Zhang 2009) has become very popular. The original MOEA/D does not incorporate any constraint handling mechanism, so we have used the variant described in (Asafuddoula et al 2012).

SMPSO (*Speed-constrained Multi-Objective PSO algorithm*) is a particle swarm optimization algorithm for solving MOPs (Nebro et al 2009). From a high level of abstraction, in a PSO algorithm a set (swarm) of candidate solutions (particles) to the problem navigates through the search space of an optimization problem. This navigation takes place based on a velocity equation, which rules the way in which particles change their position. Among the factors that govern that velocity equation, two of them can be highlighted: the current position of the particle and the best positions visited so far, also referred to as leaders. Usually, the best position visited by a particle (local leader) and the best particle visited by the whole swarm (global leader) are considered. The main innovation of SMPSO is the incorporation of a constriction velocity mechanism, already applied in single-objective PSO algorithms, which modulates the speed at which particles fly (Clerc and Kennedy 2002).

The Generalized Differential Evolution (GDE) algorithm (Kukkonen and Lampinen 2005) is also based on NSGA-II, but the crossover and mutation variation operators are changed to use the differential evolution operator. Another difference is that GDE3 modifies the crowding distance of NSGA-II in order to provide a better distributed set of solutions.

6 Experimentation

In this section, we detail the experimentation methodology we have used in our study. First, we describe the quality indicators called Hypervolume and Two Set Coverage, which are used to assess the quality of the computed Pareto front approximations. Second, we summarize the way in which we compared the different metaheuristics. After that, we describe the tools that we used to produce the results included in this work. Finally, we include the parameters settings used by the evaluated techniques.

6.1 Quality Indicators

In single-objective optimization, assessing the performance of a metaheuristic principally requires observing the best value yielded by an algorithm (i.e., the lower the better, in case of minimization problems). However, in multi-objective optimization, performance assessment is a more difficult matter, because multi-objective metaheuristics aim to converge on a set of solutions (our approximation of the Pareto optimal set), and not on a single one. As we have stated, two properties are usually required: convergence and a uniform distribution along the Pareto front. A number of quality indicators for measuring these two criteria either separately or together have been proposed in the literature. For the purposes of this paper, we have chosen the Hypervolume (I_{HV}) (Zitzler and Thiele 1999) quality indicator, which assesses both convergence and maximum spread at the same time, and Two Set Coverage (I_C), which provides a relative comparison between two solution sets based upon the dominance relationship (i.e., it assesses convergence).

6.1.1 Hypervolume (I_{HV})

The I_{HV} calculates the volume, in objective function space, enclosed between the computed approximation to the Pareto front and a reference point. Figure 8 clarifies how this indicator works in the case of a minimization problem. In this example, the approximation to the optimal Pareto front (the continuous curve) consists of the points $Q = \{A, B, C\}$ and the reference point is W. Usually, the latter is chosen using the highest possible value (minimum for a maximization problem) known for each objective function, although it does not have to be done this way. The closer the approximation is to the Pareto optimal front, the higher the value is for this indicator. Conversely, the wider the spread of the solutions along the Pareto front, the higher the value of this indicator. Therefore, approximations that produce the highest possible values of this indicator are our aim.

Given the approximation and the reference point included in that figure, I_{HV} can be computed as follows. For each solution $i \in Q$, a hypercube v_i is constructed with the reference point W and the solution i as the diagonal corners of the hypercube. Thereafter, the I_{HV} consists in the union of all these hypercubes. Mathematically:

$$I_{HV} = \text{volume}\left(\bigcup_{i=1}^{|Q|} v_i\right). \quad (14)$$

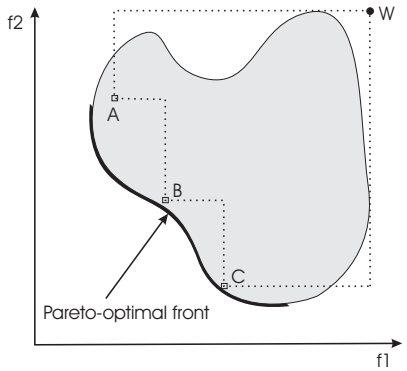


Fig. 8 The hypervolume enclosed by the set of non-dominated solutions $\{A, B, C\}$.

In some cases, the I_{HV} of an approximation may take a value equal to 0, meaning that all the points are outside the limits defined by the reference point W.

In our work, we consider W as a point consisting of the highest value of each objective in the Pareto front. To do this, it is necessary to know beforehand the Pareto front of the problem. As in the problems analyzed in this study these fronts are unknown, we have built for each instance a *reference Pareto front* as indicated in Section 6.2.

6.1.2 Two Set Coverage (I_C)

When comparing two Pareto fronts produced by different metaheuristics, it is normally desirable to know which of them has a better convergence towards the Pareto front. This can be quantitatively measured by the Two Set Coverage metric or I_{SC} (Zitzler et al 2000), which provides a relative comparison between two solution sets based upon the dominance relationship.

Given two sets of solutions X' and X'', $I_{CS}(X', X'')$ gives the ratio of points of X'' that are dominated by X' according to the following formula:

$$I_C(X', X'') = \frac{|\{\mathrm{a}'' \in X''; \exists \mathrm{a}' \in X' : \mathrm{a}' \preceq \mathrm{a}''\}|}{|X''|} \quad (15)$$

If all the solutions in X' dominate or are equal to all the points in X'' then $I_C(X', X'') = 1$, and $I_C(X', X'') = 0$ if none of the solutions in X'' are dominated by X'. Both $I_C(X', X'')$ and $I_C(X'', X')$ have to be considered for performance assessment because $I_C(X', X'')$ is not necessarily equal to $1 - I_C(X'', X')$.

When comparing a set of fronts of two algorithms after performing a number of independent runs, the reported value of I_C is the mean of applying this performance measure to the fronts obtained by each technique taken in pairs.

6.2 Experimentation Methodology

This section describes the steps followed to analyze the performance of the evaluated metaheuristics.

For each of the two problem instances considered we have run each of the 7 algorithms 30 times, resulting in a total of $7 \times 30 = 240$ runs per instance. The non-dominated solutions from these 240 runs have been taken as the reference Pareto front in each case. After this, I_{HV} has been computed for each of the runs using the reference front, and the results are reported using boxplot representations.

In addition, when comparing the values yielded by two algorithms on a given problem, we check whether differences in the computed I_{HV} by different algorithms are statistically significant. To address this issue, we have applied the unpaired Wilcoxon rank-sum test, a non-parametric statistical hypothesis test, which allows us to make pairwise comparisons between algorithms to analyze the significance of the data obtained (Demšar 2006). A confidence level of 95% (i.e., significance level of 5% or p-value under 0.05) has been used in all cases, meaning that the differences are unlikely to have occurred by chance with a probability of 95%. The results of these tests have been summarized in the form of a table for each instance. Each cell in these tables contains the results of this test for a pair of algorithms. These results are summarized using three different symbols for the sake of clarity: "–" indicates that there is no statistical significance between these algorithms, "▲" means that the algorithm in the row has yielded better results than the algorithm in the column with statistical confidence, and "▽" is used when the algorithm in the column is statistically better than the algorithm in the row.

6.3 Implementation Details

In this section, we describe the tools we have considered in the work presented here, providing interested readers with the chance to reproduce our study, or even to extend it to other domains.

We have used two Open Source tools that can work together. On the one hand, the design of the structures

has been carried out using Ebes, a software for the design and analysis of spatial bar structures, which we developed. On the other hand, all the algorithm implementations adopted in this paper are available in jMetal (Durillo and Nebro 2011), a Java based multi-objective optimization framework widely known in the field of multi-objective optimization, which was also developed by us.

The structures designed with Ebes are stored in text files which contain all the information describing them. These files can be loaded into jMetal so that any of the algorithms provided by this can later be applied to optimize the associated problem. The output of jMetal consists of two files: one containing the computed approximation to the Pareto set, and another one containing the corresponding approximation to the Pareto front. These files can be loaded again in Ebes, allowing an engineer or person with expertise in the area to explore all the obtained solutions and choose the most promising one(s) taking into consideration specific design goals.

Both tools, Ebes and jMetal, work as an integrated software (Zavala et al 2014). The codes (both source and binary codes), as well as the Ebes files describing the problems solved in this paper, can be found at `http://ebesjmetal.sourceforge.net`.

6.4 Parameter Settings

The adopted parameterization of the algorithms is summarized in Table 4. These parameters have been chosen after an initial experimental phase where a sensitivity analysis for each parameter for the 25N_35E instance was carried out. To ensure a fair comparison, all the algorithms have to find a Pareto front approximation of 100 individuals. The total number of evaluations has been set to 150,000 for the smallest bridge and to 500,000 for the 133N_221E.

7 Analysis of Results

This section includes the analysis of the results obtained after evaluating the seven metaheuristic techniques adopted. The section is divided into three parts. The first and the second are devoted to the algorithm comparison from the numerical point of view; and the third contains a discussion from a civil engineer's point of view.

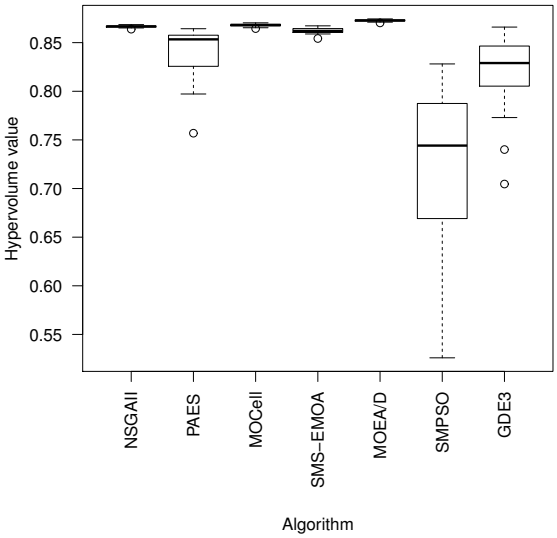


Fig. 9 Boxplot representation of the I_{HV} values obtained for the 25N_35E bridge.

7.1 Metaheuristics Comparison

We now provide the analysis of the results from each bridge by assessing the performance of the metaheuristics by using the I_{HV} quality indicator.

7.1.1 Bridge 25N_35E

The values obtained after applying I_{HV} to each of the fronts gathered from the 30 independent runs performed by the algorithms when solving the 25N_35E bridge are summarized in Figure 9 in the form of boxplots. It's important to keep in mind that the bigger the hypervolume the better.

The analysis of the boxplots shows that there is a group of four techniques (MOEA/D, MOCell, NSGA-II, and SMS-EMOA), which are clearly ahead of the rest. The algorithms in that group not only provide the best median values, but they are also robust in the sense that the differences between the maximum and minimum values represented by each boxplot are very small.

The ranking of the best performing algorithms is led by MOEA/D, followed by MOCell and NSGA-II. To have a clearer view of the statistical significance of the results we can take a look at the output of applying the Wilcoxon rank-sum test, which is included in Table 5. The results reported by the test indicate that there are significant differences in all the pair-wise comparisons between all the algorithms.

Parameterization used in NSGA-II	
Population Size	100 individuals
Selection of Parents	binary tournament
Recombination	simulated binary crossover, $p_c = 0.9$
Recomb. distribution index	40
Mutation	polynomial, $p_m = 1.0/L$
Mutation distribution index	20
Parameterization used in PAES	
Mutation	polynomial, $p_m = 1.0/L$
Mutation distribution index	5
Archive Size	100
Parameterization used in MOCell	
Population Size	100 individuals (10×10)
Neighborhood	8 surrounding neighbors
Selection of Parents	binary tournament
Recombination	simulated binary crossover, $p_c = 0.9$
Recomb. distribution index	40
Mutation	polynomial, $p_m = 1.0/L$
Mutation distribution index	10
Archive Size	100 individuals
Parameterization used in SMS-EMOA	
Population Size	100 individuals
Selection of Parents	binary tournament
Recombination	simulated binary crossover, $p_c = 0.9$
Recomb. distribution index	20
Mutation	polynomial, $p_m = 1.0/L$
Mutation distribution index	5
Offset	100
Parameterization used in SMPSO	
Particles	100 particles
Inertia weight	0.1
Mutation	polynomial (to 1/6 of the population)
Mutation distribution index	40
Leaders Size	100
Parameterization used in MOEA/D	
Population Size	100 individuals
Neighborhood parameters	$T = 20$, $delta = 0.9$, $nr = 2$
Mutation	polynomial, $p_m = 1.0/L$
Mutation distribution index	40
Recombination	Diff. Evolution, $CR = 1.0$, $F = 0.5$
Parameterization used in GDE3	
Population Size	100 individuals
Recombination	Diff. Evolution, $CR = 0.1$, $F = 0.5$

Table 4 Parameterization of the evaluated algorithms (L = solution length).

	PAES	MOCell	SMS-EMOA	MOEA/D	SMPSO	GDE3
NSGAII	▲	▽	▲	▽	▲	▲
PAES		▽	▽	▽	▲	▲
MOCell			▲	▽	▲	▲
SMS-EMOA				▽	▲	▲
MOEA/D					▲	▲
SMPSO						▽

Table 5 Statistical comparison (Wilcoxon rank-sum test) summary for the I_{HV} values of the 25N_35E bridge instance. The symbols mean: the algorithm in the row is statistically better than the one in the column ('▲'), the opposite ('▽'), and no statistically significant difference ('–').

The performance analysis reveals the numerical performance of the algorithms, but does not give all the information to the civil engineer about the pros and cons of the techniques. To provide an insight into the expected results, we include, in Figure 10, the fronts corresponding to those with the median I_{HV} value of MOEA/D, MOCell and NSGA-II. We can see that MOEA/D provides solutions from the extremes of the reference Pareto front, with a progressive concentration of solutions towards the center of the fronts, which presents a high degree of convergence. The front of MOCell is characterized by having a uniform spread of solutions, but the far right of the reference Pareto front is not covered, and this is the reason why MOEA/D

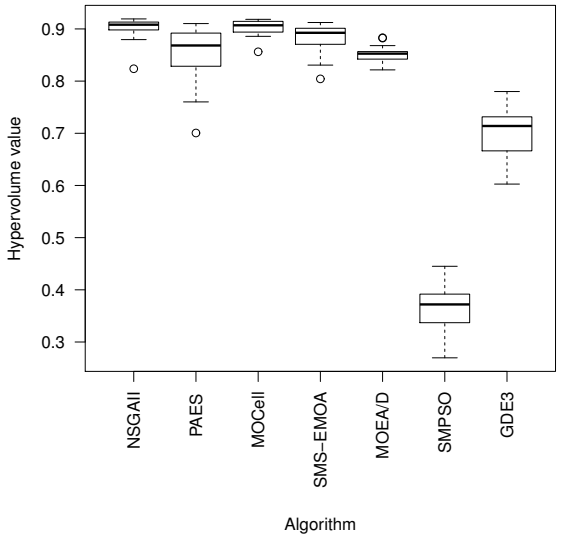


Fig. 11 Boxplot representation of the I_{HV} values obtained for the 133N_221E bridge

provides a better I_{HV} value than MOCell in this problem. Finally, NSGA-II covers the full reference Pareto front except for the far left (we can observe that the leftmost solution does not achieve the value 0.5 in the y-axis). In this case, the diversity of the solutions is not as uniform as with MOCell, and we can also see that the solutions in the middle of the front have not fully converged. These two facts mean that MOEA/D outperforms NSGA-II with respect to I_{HV}.

7.1.2 133N_221E

The boxplots summarizing the I_{HV} values obtained when solving the 133N_221E bridge are included in Figure 11.

The results in this case are slightly different to the ones obtained for the 25N_35E bridge. First, it is remarkable that none of the best performing algorithms have been as robust as in the previous instance. This lack of robustness is an indication of the difficulty of the problem, so we can guess that none of the algorithms have fully converged towards the optimal Pareto front after evaluating 500,000 solutions. This suggests that there is room for improvement if we can afford spending more computing time. Second, the best algorithm in the 25N_35E bridge, MOEA/D, has suffered a downgrade in its performance, so it has not scaled well with the size of the problem.

The leading algorithms are MOCell and NSGA-II which, according to the results of the Wilcoxon rank-sum test, included in Table 6, do not yield significant differences.

	NSGAII	MOCell
25N_35E bridge		
NSGAII	–	0.26
MOCell	0.41	–
133N_221E bridge		
NSGAII	–	0.38
MOCell	0.43	–

Table 7 $I_C(X', X'')$ metric values when comparing MOCell and NSGA-II (X' and X'' represent the algorithm in the row and in the column, respectively).

As in the case of the 25N_35E bridge, we analyze the Pareto front approximations obtained for MOEA/D, MOCell and NSGA-II which have the median I_{HV} value. The fronts are included in Figure 12. It is easy to see that MOEA/D has difficulty in converging on the left side of the reference Pareto front, hence its poor I_{HV} value. MOCell and NSGA-II present a front with similar features to the one obtained in the 25N_35E bridge: good diversity and lack of coverage of the far right of the front (MOCell), as well as not such a good diversity, lack of solutions in the far left, and an inappropriate convergence in the medium region (NSGA-II). As a consequence, MOCell yields a more accurate front than NSGA-II.

7.2 Discussion

The comparative study conducted in the previous section indicates that, in the context of the considered structural problems and the parameter settings applied in the selected metaheuristics, there is a group of two solvers, MOCell and NSGA-II, that are worth considering when dealing with structural problems such as those adopted in our study, regardless of the problem size.

A view of the fronts of these two metaheuristics in Figures 10 and 12 seems to indicate that the Pareto front approximations of MOCell have a higher degree of convergence than NSGA-II. In order to provide a quantitative confirmation of this claim we have applied I_C to the 30 Pareto front approximations generated by the two algorithms in the two considered problems. The results (see Table 7) indicate that in both problems MOCell outperforms NSGA-II, although in the case of the 133N_221E bridge, the difference is small.

Regarding the rest of the techniques adopted, MOEA/D achieved the best result in the small instance of the problem, but had trouble as the size of the problem grew. SMS-EMOA, the indicator-based algorithm, ranked fourth and third in the two problems, respectively, so it can be considered as a possible choice after

	PAES	MOCell	SMS-EMOA	MOEA/D	SMPSO	GDE3
NSGAII	▲	–	▲	▲	▲	▲
PAES		▽	▽	–	▲	▲
MOCell			▲	▲	▲	▲
FastSMSEMOA				▲	▲	▲
cMOEAD					▲	▲
SMPSO						▽

Table 6 Statistical comparison (Wilcoxon rank-sum test) summary for the I_{HV} values of the 133N_221E bridge instance. The symbols mean: the algorithm in the row is statistically better than the one in the column ('▲'), the opposite ('▽'), and no statistically significant difference ('–').

MOCell and NSGA-II. The performance of PAES shows that it is behind SMS-EMOA. Finally, GDE and, in particular, SMPSO performed poorly in the two problem instances, so they should be discarded in favor of the other algorithms.

It is worth considering the issue of the total running time of the algorithms. The execution of NSGA-II over 150,000 evaluations to solve the 25N_35E bridge is about six minutes in a MacBook Pro (2.2 GHz Intel Core i7 processor, 8 GB 1333 MHz DD3, 256 GB SSD) running MacOS X 10.9.2 and using Java 1.8.0. The computing time increases to 4.4 hours when solving the 133N_221E bridge on the same machine. These times are certainly affordable but it is clear that, as the running time of the solvers is problem dependent, more complex designs could require more computing power. We addressed the design of a high dimensional cable-stayed bridge, with 1584 elements and 837 nodes, in Luna et al (2015). This problem instance is indeed so large that we were forced to develop parallel versions of the multi-objective algorithms (NSGA-II and SMS-EMOA), which were able of using up to 420 processing elements at the same time. Even with such a computing platform, the algorithms required more than 18 hours to compute 300,000 function evaluation and to reach an accurate solution to the problem instance.

7.3 Evaluation from a Decision Maker's Point of View

In real-world applications, the decision maker is normally interested in certain types of trade-offs. From this point of view, the fronts produced by MOCell and NSGA-II in Figures 10 and 12 appear similar, but if designs with the minimum weight are desired, then MOCell is preferable. As analyzed in the previous section, this algorithm also generates solution sets with better convergence than NSGA-II, although the differences are not so remarkable as to discard this algorithm, which has the advantage of being very well-known.

To demonstrate the kind of trade-off designs that can be obtained after the multi-objective optimization process, Figures 13 and 14 show five possible designs for each bridge than can be selected from a set of solutions yielded by MOCell. The figures allow us to clearly see how the resulting bridges become robust (and heavier) when taking solutions from the left to the right part of the front. To illustrate the final outcome, we have considered the criterion of choosing the least heavy solutions from this front. The values of the decision variables and the objective functions for the 25N_35E and 133N_221E bridges are included in Tables 8 and 9, respectively.

The resulting optimized design of the 133N_221E instance can be observed in Figure 15, which shows the values of the cross-sections of the elements of the bridge. Furthermore, using a tool such as Ebes, the performance of the design can be studied according to different properties, such as elasticity of the structure (Figure 16), bending moments (Figure 17), shear (Figure 18), and axial forces (Figure 19). In the figures, the associated color ramps are intended to represent the internal strength intensity associated with each element.

8 Conclusions and Future Work

In this paper, we have studied the performance of seven state-of-the-art multi-objective metaheuristics when solving two instances of a real-world civil engineering problem, namely, the design of two variants of a cable-stayed bridge. The design problem has been formulated as having two conflicting objectives: reduce the overall structure weight and minimize the deformation in selected nodes.

After describing the problem formulations in detail and presenting the chosen algorithms, we carry out a rigorous comparative study to assess the performance of the seven techniques. The results obtained when solving the simpler design problem (the so-called 25N_35E bridge) show that the best performing algorithm is MOEA/D, followed by MOCEll and NSGA-II.

The second instance, referred to as the 133N_221E bridge, is more complex than the 25N_35E, which is evidenced by the fact that the algorithms have not fully converged. The results indicate that the best performing techniques are NSGA-II and MOCell, and, to a lesser extent, SMS-EMOA, while MOEA/D has been

Shape	Description	Gr.	Y m	Z m	ty mm	tz mm	σ_{max} MPa	σ_{min} MPa	τ MPa	σ_a MPa	τ_a MPa
I-single	Transversal beam	3	0.26	0.11	9	6	146.8	-146.9	0.5	±150	90
Hollowed rectangle	Longitudinal beam 1	4	0.39	0.19	15	13	64.4	-120.3	26.7		
	Longitudinal beam 2	5	0.33	0.16	11	8	56.7	-144.8	11.8		
	Longitudinal beam 3	6	0.29	0.12	10	6	52.4	-147.7	2.6		
	Column	0	0.52	0.21	18	11	37.8	-144.2	11.2		
	Link beam of columns	2	0.33	0.20	13	11	0.4	-79.2	0.5		
Circle	Anchored cable	1	0.07				134.4			750	
	Stayed cable	7	0.04				189.7				
$f_1 = 0.030293MN$, $f_2 = 0.387m$											

Table 8 Decision variables and function values from a solution returned by NSGA-II for the 25N_35E bridge.

Shape	Description	Gr.	Y m	Z m	ty mm	tz mm	σ_{max} MPa	σ_{min} MPa	τ MPa	σ_a MPa	τ_a MPa
I-single	Beam Cantilever 1 of deck	22	0.20	0.14	13	13	39.23	-39.23	7.22	±150	90
	Beam Cantilever 2 of deck	23	0.20	0.14	13.0	13.0	17.11	-18.64	6.68		
	Beam 1 of deck	11	0.22	0.16	8.5	9.3	147.9	-146.08	0.52		
	Beam 2 of deck	12	0.25	0.17	9.3	8.7	149.92	-148.25	0.34		
Hollowed rectangle	Longitudinal beam 1	3	0.48	0.38	20.0	22.1	0.00	-102.9	0.46		
	Longitudinal beam 2	4	0.48	0.33	26.9	22.1	0.00	-95.17	0.32		
	Longitudinal beam 3	5	0.50	0.39	17.6	21.8	0.00	-100.27	0.65		
	Longitudinal beam 4	6	0.39	0.42	20.0	23.2	0.00	-87.49	1.10		
	Longitudinal beam 5	7	0.46	0.48	18.5	27.3	0.00	-80.90	1.06		
	Longitudinal beam 6	8	0.52	0.42	20.0	23.2	0.00	-126.08	0.70		
	Longitudinal beam 7	13	0.43	0.42	20.8	22.5	2.03	-121.17	0.28		
	Longitudinal beam 8	14	0.520	0.46	18.7	29.7	0.00	-58.54	0.31		
	Longitudinal beam 9	15	0.41	0.28	16.3	21.6	0.00	-132.11	2.27		
	Longitudinal beam 10	17	0.43	0.30	14.6	16.7	0.00	-136.06	0.55		
	Longitudinal beam 11	18	0.40	0.28	20.7	23.0	0.00	-114.50	2.58		
	Longitudinal beam 12	19	0.55	0.39	20.0	20.0	28.99	-78.08	1.69		
	Longitudinal beam 13	20	0.41	0.36	19.6	18.2	14.29	-77.18	0.55		
	Longitudinal beam 14	25	0.31	0.32	17.1	22.8	0.0	-59.95	0.69		
	Longitudinal beam 15	26	0.40	0.28	15.7	22.3	0.7.95	-60.46	1.22		
	Longitudinal beam 16	27	0.35	0.25	12.0	17.4	55.01	-55.44	3.69		
	Link beam of columns	2	0.46	0.34	15.7	18.7	0.00	-70.71	0.07		
	Column 1	29	0.61	0.46	24.5	23.6	0.00	-145.04	0.00		
	Column 2	30	0.54	0.55	18.8	29.6	0.00	-148.67	0.00		
	Column 3	31	0.57	0.48	29.6	25.2	0.00	-150.70	0.00		
	Column 4	32	0.59	0.49	23.7	25.6	0.00	-149.37	0.00		
Circle	Diagonal tensor	22	0.08				2.43	-146.72			
	Anchored cable 1	0	0.12				256.67				
	Anchored cable 2	1	0.02				183.15				
	Stayed cable 1	10	0.04				198.47				
	Stayed cable 2	9	0.06				259.41			750	
	Stayed cable 3	16	0.02				272.30				
	Stayed cable 4	21	0.07				209.18				
	Stayed cable 5	28	0.07				175.13				
$f_1 = 0.28312\ MN$, $f_2 = 3.118\ m$											

Table 9 Decision variables and function values from a solution returned by MOCell for the 133N_221E bridge.

unable to properly scale with the problem's complexity.

By considering the two problems together, MOCell has emerged as the best performing algorithm, followed by NSGA-II. The worst algorithms in the comparative study were GDE3 and SMPSO, which yielded a very poor performance.

At the end of our study, we have selected a solution from each problem and provided details of the designs obtained.

We can conclude that multi-objective metaheuristics are a very useful tool for civil engineers, as they are able to provide a wide range of trade-off designs when applied to problems such as those addressed in this paper. From our study, we can make some hints as to what techniques are more promising to find the solutions to similar design problems. MOCell and NSGA-II are very competitive in this context, and they could be the primary choice over the other algorithms analyzed.

Extending our study to other structural design problems is an interesting research path that we intend to explore in the near future. Another interesting research line has to do with the fact that the variation operators used in the metaheuristics operate without any knowledge of the problem, with suggests that ad-hoc operators could be designed to provide more effective and efficient search capabilities, which could speed up the optimization process.

Acknowledgements This work is partially funded by Grants TIN2014-58304-R (Ministerio de Ciencia e Innovación) and P11-TIC-7529 and P12-TIC-1519 (Plan Andaluz

de Investigación, Desarrollo e Innovación). The last author gratefully acknowledges support from CONACyT project no. 221551.

References

Asafuddoula M, Ray T, Sarker R, Alam K (2012) An adaptive constraint handling approach embedded moea/d. In: Evolutionary Computation (CEC), 2012 IEEE Congress on, IEEE, pp 1–8

Bäck T, Fogel D, Michalewicz Z (eds) (1997) Handbook of evolutionary computation. Oxford University Press

Baluja S (1994) Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning. Tech. Rep. CMU-CS-94-163, Carnegie-Mellon University, Pittsburgh, Philadelphia, USA

Beume N, Naujoks B, Emmerich M (2007) Sms-emoa: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research 181(3):1653–1669

Clerc M, Kennedy J (2002) The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 6(1):58–73

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2):182–197

Demšar J (2006) Statistical Comparisons of Classifiers over Multiple Data Sets. J Mach Learn Res 7:1–30

Durillo JJ, Nebro AJ (2011) jmetal: A java framework for multi-objective optimization. Adv Eng Softw 42(10):760–771

Glover F, Kochenberger GA (2003) Handbook of metaheuristics. Springer

Kaveh A, Laknejadi K (2011) A Hybrid Multi-Objective Optimization and Decision Making Procedure for Optimal Design of Truss Structures. Iranian Journal of Science and Technology–Transactions of Civil Engineering 35(C2):137–154

Kelesoglu O (2007) Fuzzy multiobjective optimization of truss-structures using genetic algorithm. Advances in Engineering Software 38(10):717–721

Knowles J, Corne D (1999) The pareto archived evolution strategy: A new baseline algorithm for multiobjective optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation, IEEE Press, Piscataway, NJ, pp 9–105

Knowles J, Corne D (2003) Instance generators and test suites for the multiobjective quadratic assignment problem. In: Fonseca C, Fleming P, Zitzler E, Deb K, Thiele L (eds) Evolutionary Multi-Criterion Optimization, Second International Conference, EMO 2003, Faro, Portugal, April 2003, Proceedings, Springer, no. 2632 in LNCS, pp 295–310

Knowles J, Thiele L, Zitzler E (2006) A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. Tech. Rep. 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich

Knowles JD, Corne DW (2000) Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. Evolutionary Computation 8(2):149–172

Kukkonen S, Lampinen J (2005) GDE3: The third evolution step of generalized differential evolution. In: IEEE Congress on Evolutionary Computation (CEC'2005), pp 443 – 450

Li H, Zhang Q (2009) Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. IEEE Transactions on Evolutionary Computation 12(2):284–302

Luna F, Zavala GR, Nebro AJ, Durillo JJ, Coello CA (2015) Distributed multi-objective metaheuristics for real-world structural optimization problems. The Computer Journal

Lust T, Teghem J (2010) The multiobjective traveling salesman problem: A survey and a new approach. In: Coello Coello C, Dhaenens C, Jourdan L (eds) Advances in Multi-Objective Nature Inspired Computing, Studies in Computational Intelligence, vol 272, Springer Berlin Heidelberg, pp 119–141

Nebro A, Durillo J, Luna F, Dorronsoro B, Alba E (2006) A cellular genetic algorithm for multiobjective optimization. In: Pelta DA, Krasnogor N (eds) Proceedings of the Workshop on Nature Inspired Cooperative Strategies for Optimization (NICSO 2006), Granada, Spain, pp 25–36

Nebro A, Durillo J, Luna F, Dorronsoro B, Alba E (2007) Design issues in a multiobjective cellular genetic algorithm. In: Obayashi S, Deb K, Poloni C, Hiroyasu T, Murata T (eds) Evolutionary Multi-Criterion Optimization. 4th International Conference, EMO 2007, Springer, Lecture Notes in Computer Science, vol 4403, pp 126–140

Nebro AJ, Durillo JJ, Garca-Nieto JM, Coello CAC, Luna F, Alba E (2009) SMPSO: A new PSO-based metaheuristic for multi-objective optimization. In: 2009 IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making, pp 66 – 73

Przemieniecki J (1968) Theory of Matrix Structural Analysis. Dover, New York, USA

Turner M, Clough RW, Martin HC, Topp LJ (Setember 1956) Stiffness and deflection analysis of complex structuures. Tech. rep., Journal of the Aeronautical Sciences (Institute of the Aeronautical Sciences) 23 (9) pp. 805-823

Yang Y, McGuire W (1986) Stiffness matrix for geometrix nonlinear analysis. J. Struct. Eng. ACE, 112(4), 879

Zapotecas Martínez S, Coello Coello CA (2014) A Multi-objective Evolutionary Algorithm based on Decomposition for Constrained Multi-objective Optimization. In: 2014 IEEE Congress on Evolutionary Computation (CEC'2014), IEEE Press, Beijing, China, pp 429–436, iSBN 978-1-4799-1488-3

Zavala G, Nebro A, Luna F, Coello Coello C (2013) A survey of multi-objective metaheuristics applied to structural optimization. Structural and Multidisciplinary Optimization 49(4):1–22

Zavala G, Nebro A, Durillo J, Luna F (2014) Integrating a multi-objective optimization framework into a structural design software. Advances in Engineering Software 76:161–170

Zhang Q, Li H (2007) MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evolutionary Computation 11(6):712–731

Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans Evolutionary Computation 3(4):257–271

Zitzler E, Deb K, Thiele L (2000) Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation 8(2):173–195

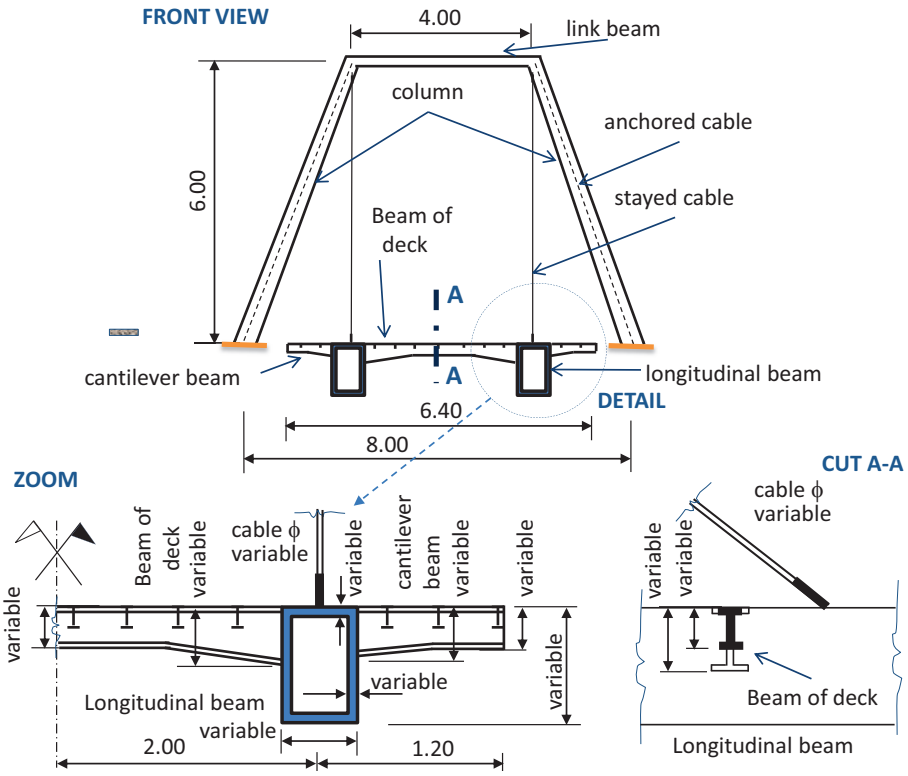


Fig. 4 Scheme of beams in bridge 133N_221E.

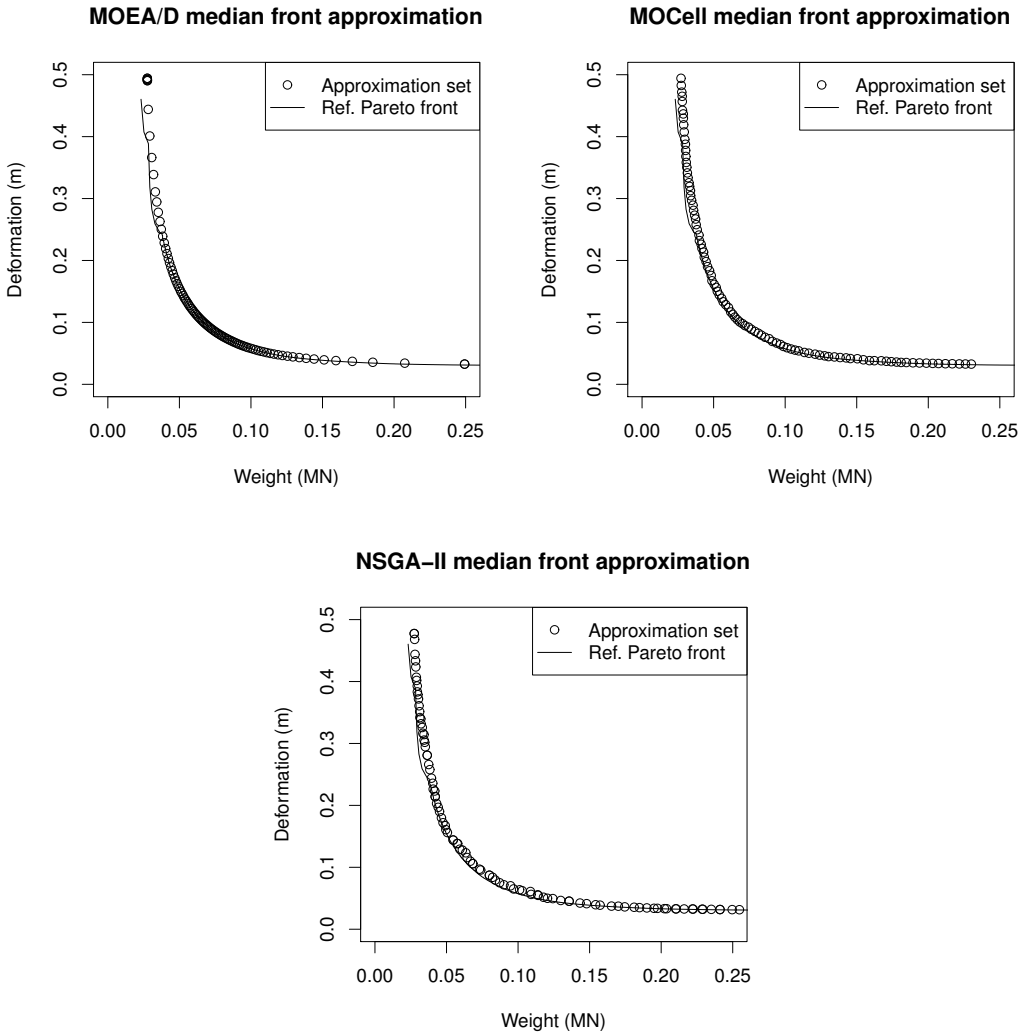


Fig. 10 Pareto front approximations corresponding to the median I_{HV} for the 25N_35E Bridge.

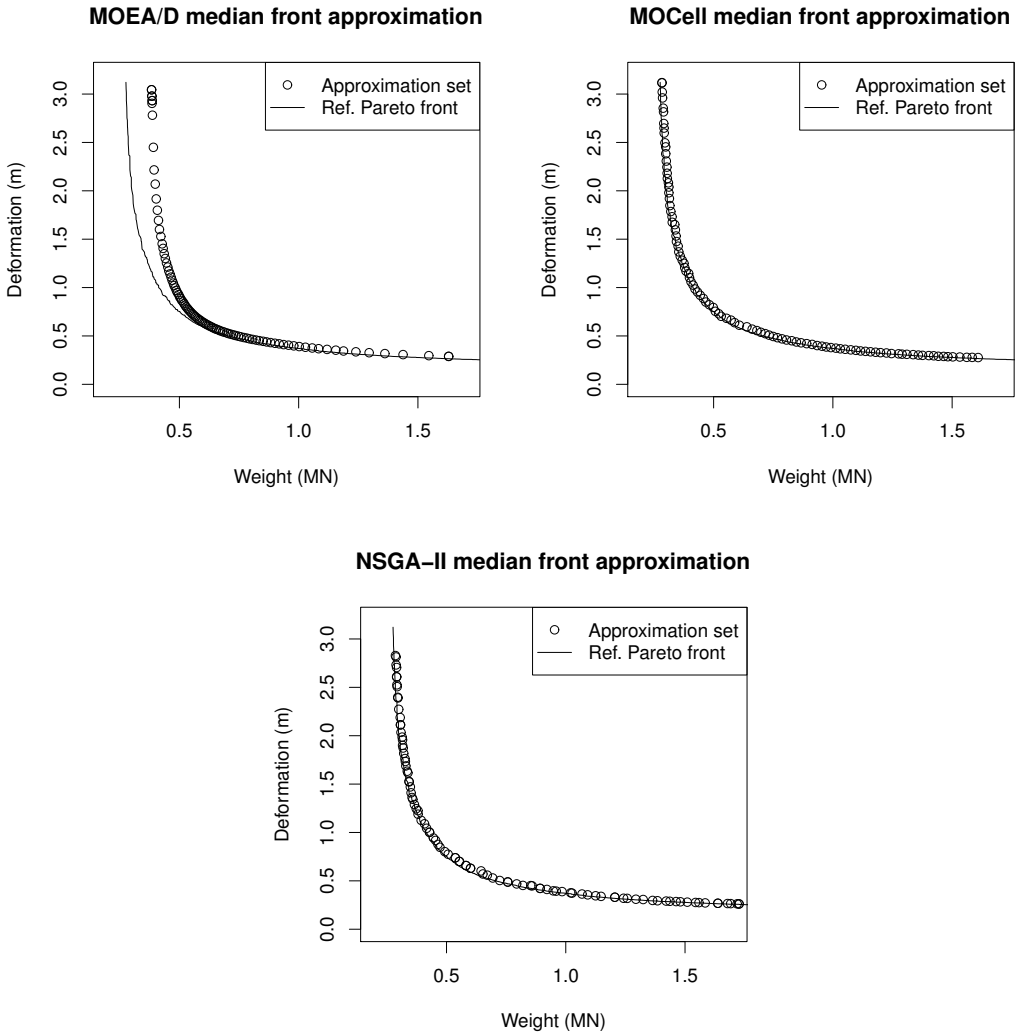


Fig. 12 Pareto front approximations corresponding to the median I_{HV} for the 133N_221E bridge

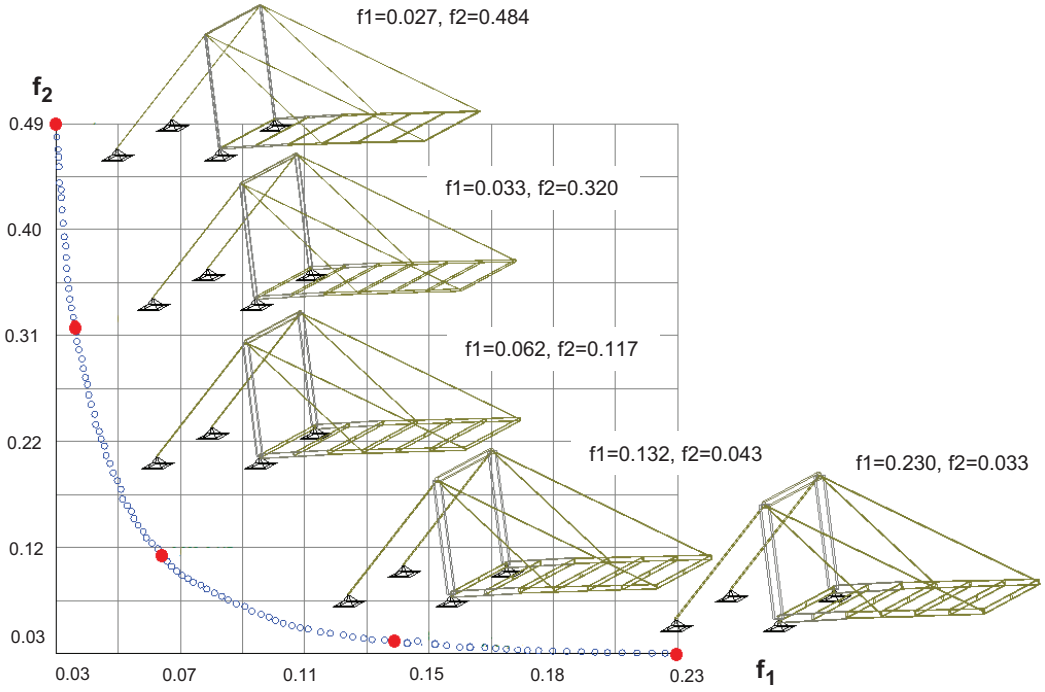


Fig. 13 Designs obtained by selecting different solutions of the Pareto front approximation for the 25N_35E bridge.

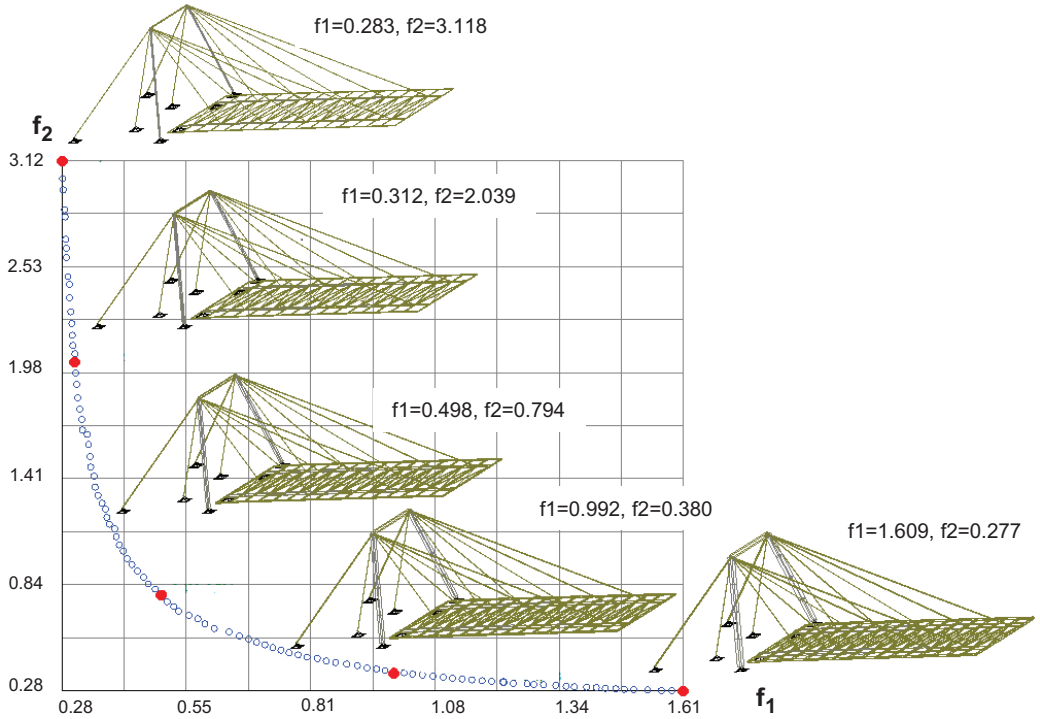


Fig. 14 Designs obtained by selecting different solutions of the Pareto front approximation for the 133N_221E bridge.

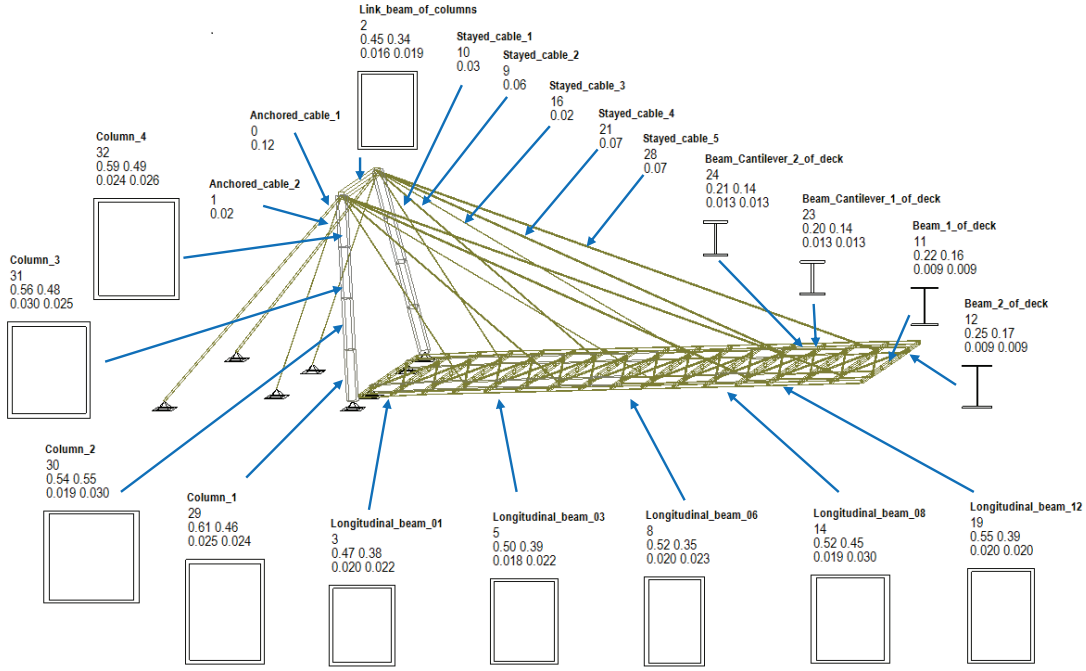


Fig. 15 Cross-section values for the selected design of the 133N_221E bridge.

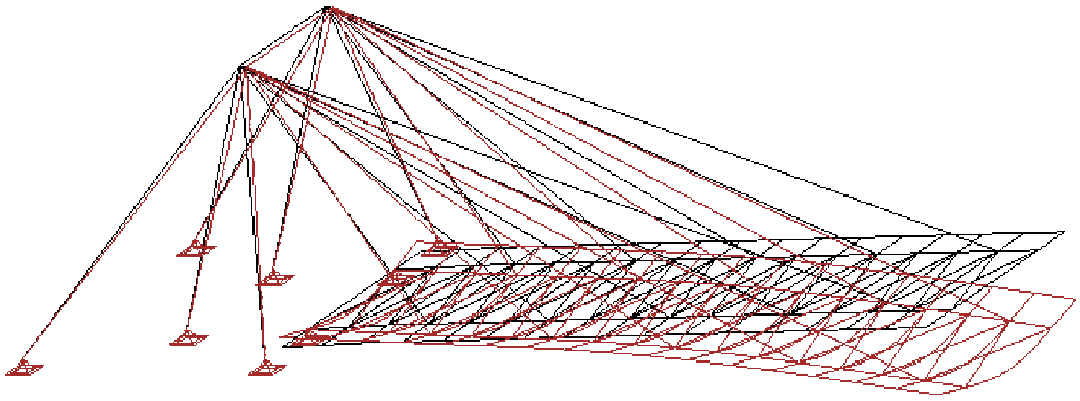


Fig. 16 Elasticity of the 133N_221E bridge

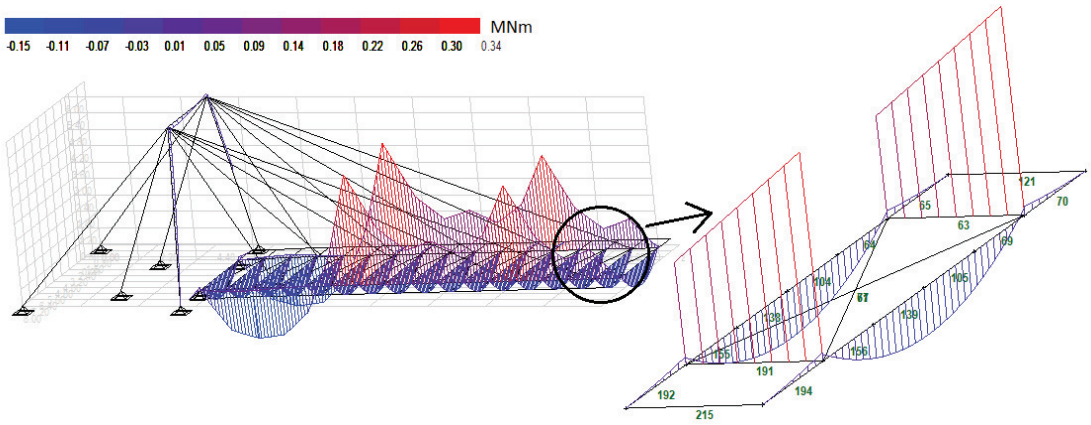


Fig. 17 Bending moment diagram of the 133N_221E bridge

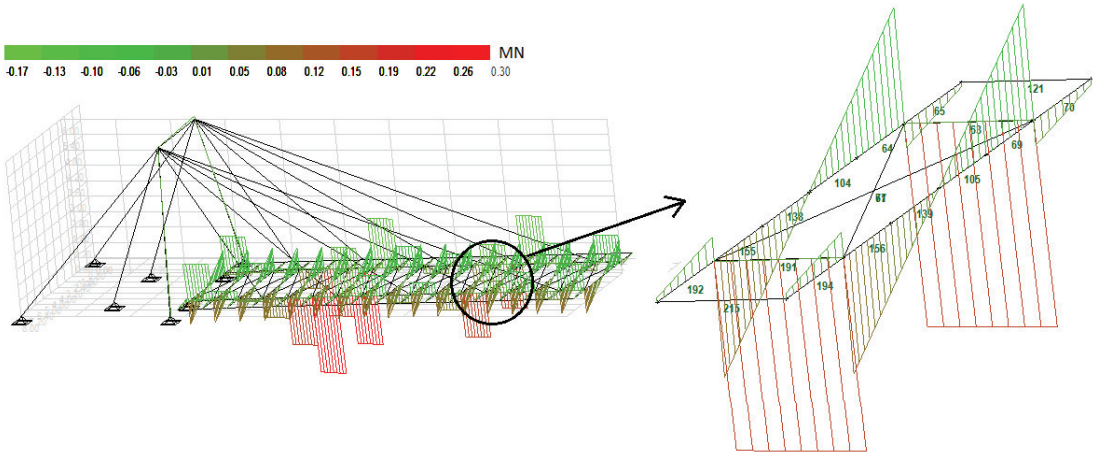


Fig. 18 Shear diagram of the 133N_221E bridge

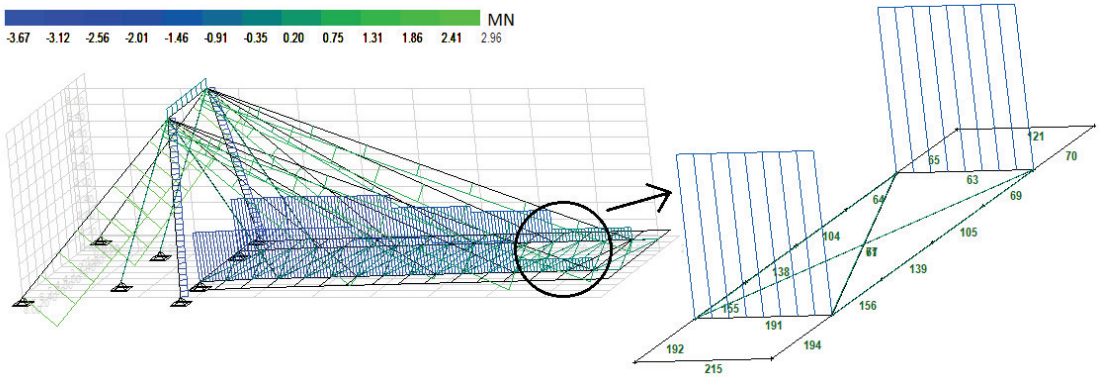


Fig. 19 Axial forces diagram of the 133N_221E bridge