COMPDYN 2015
5th ECCOMAS Thematic Conference on
Computational Methods in Structural Dynamics and Earthquake Engineering
M. Papadrakakis, V. Papadopoulos, V. Plevris (eds.)
Crete Island, Greece, 25–27 May 2015

MULTI-OBJECTIVE OPTIMIZATION UNDER UNCERTAINTY WITH REAL-TIME INTEGRATED DECISION MAKING APPLIED TO STRUCTURAL ENGINEERING

Mariapia Marchi[1], **Mauro Munerato**[1], **Luca Rizzian**[1] **and Stefano Costanzo**[1]

[1] ESTECO S.p.A.
Padriciano 99, 34149 Trieste (Italy)
e-mail: marchi,munerato,rizzian,costanzo@esteco.com

Keywords: Multi-Objective Optimization, Multi-Criteria Decision Making, Interactive Optimization, Robust Design Optimization, Reliability-Based Design Optimization, Reliability Assessment of Structures

Abstract. *One of the major tasks of structural engineering design optimization is the handling of uncertainties (such as variations in material properties, loading conditions, unknown environmental conditions or even uncertainties in modeling assumptions), which affect system performance in terms of robustness and reliability (or, in other words, the ability to respond to input variations with minimal alteration, loss of functionality or damage). This task is usually tackled with Optimization Under Uncertainty (OUU) methods[1], like robust design optimization and reliability-based design optimization. In most cases, the optimization has to deal with multi-objective problems (such as maximizing the performance while minimizing costs, system response variations, etc). These problems do not have a unique solution, but a set of tradeoff optimal solutions (the so-called Pareto front). The action of a decision maker (DM) is necessary for choosing the final optimal design according to some (pre-defined) preferences or criteria. Multi-Criteria Decision Making (MCDM) techniques[2] have been developed over the past years to try to make these choices objective and rational. In most MCDM methods, the preferences are usually taken into account during some a-posteriori analyses of the optimization outcomes.*

Here we address both OUU and MCDM problems with an approach that integrates directly the action of the DM with the optimization process. The DM is asked to express their preferences (based on their previous experience) to drive the optimization towards the most preferred regions of the Pareto front. This can lead to a more efficient exploration of specific regions of the Pareto front and reduce the computational cost of finding desirable solutions. Interactive MCDM approaches have been recently given more attention in the multi-objective optimization community [3, 4, 5]. A validation of this approach on simple test-cases is shown as well as its application to the design of a simple building structure under uncertainties with seismic hazard and snow loads.

1 INTRODUCTION

Some of the main challenges in the field of structural engineering design include the handling of uncertainties arising, e.g., from variations in material properties, loading conditions, unknown environmental conditions or even uncertainties in modeling assumptions, and the multi-objective nature of design problems requiring a trade-off between several (conflicting) objectives (such as performance maximization, cost minimization, variation minimization, etc.).

Uncertainties affect system robustness and reliability, i.e. the ability to respond to input variations with minimal alteration, loss of functionality or damage. Thus in the past decades a number of Optimization Under Uncertainty (OUU) methods[1], like Robust Design Optimization (RDO) and Reliability-Based Design Optimization (RBDO), have been developed to find designs, which are robust and reliable unlike those sometimes found with deterministic optimization methods.

Multi-objective optimization (MO) problems do not have a unique solution, but a set of tradeoff optimal solutions (the so-called Pareto front). Hence a compromise between conflicting requirements is sought. The intervention of a decision maker (DM) is necessary for choosing the final optimal design according to some (pre-defined) preferences or criteria. Multi-Criteria Decision Making (MCDM) techniques[2] have been developed over the past years to make these choices objective and rational. In many MCDM methods, the preferences are taken into account during a-posteriori analyses of the optimization results. Other techniques include the preferences directly in the optimization phase, usually by rewriting the multi-objective problem as a single-objective problem, where a function of the preferences has to be optimized.

The importance of MO in real-world structural engineering problems was highlighted for instance by Papadrakakis et al. in Ref. [6]. Structural sizing examples were studied and optimized with Evolutionary Algorithms (EAs). EAs are more efficient than gradient-based methods for these kinds of problems, because they rely on the evolution of a population of optimal candidate designs. Furthermore EAs are robust global search algorithms (they need no derivatives), which are less likely to be trapped in local optimal regions of large and complex search spaces. Ref. [7] addressed a reliability-based multi-objective sizing optimization of a multi-storey steel frame under seismic loading, where probabilistic constraints were introduced to account for the system reliability. Failure probabilities were computed with the Latin Hypercube Sampling (LHS)[8]. Beck et al. [9] developed a software to find the tradeoff between conflicting design criteria while accounting for modeling and loading uncertainties. They turned the MO into a Single-objective Optimization (SO) of the aggregation of the preference functions. Frangopol[10] reviewed the relevance of using MO and decision making in civil infrastructure maintenance management under uncertainty, while Caterino et al. [11] focused on *a posteriori* MCDM techniques applied to the specific problem of seismic structural retrofitting. Jensen et al. [12] developed a method for efficiently obtaining compromise design solutions and tradeoff information between objectives in the optimization of structural systems under stochastic excitations by introducing an auxiliary variable to solve the MO problem as a SO by means of line search techniques. In Ref. [13] certain authours of this paper studied a structural sizing MO under uncertainty (MOUU) problem by using Polynomial Chaos[14] Expansion (PCE) techniques to determine the system reliability with a method already applied in Refs. [15, 16].

Motivated by the growing interest shown by structural engineering specialists in MCDM and MO techniques, in this paper we address both MOUU and MCDM problems with an approach that integrates directly the action of the DM with the optimization process. For this purpose, a novel algorithm has been developed: the DM is asked to express their preferences (based on

their previous experience) as target points to drive the optimization towards the most preferred regions of the Pareto front. In principle this can lead to a more efficient exploration of specific regions of the Pareto front and reduce the computational cost of finding desirable solutions.

Interactive MCDM techniques have been recently given more attention in the MO community. Two approaches have been explored: direct integration of the DM in the optimization process[3, 4], by giving them the possibility to insert some reference point, and real-time interaction with the optimization algorithm[5], by expressing their preference each time after a defined number of iterations. The target point method follows the first approach. A validation of this approach on simple test-cases is shown as well as its application to the design of a simple building structure under uncertainties with seismic hazard and snow loading. A comparison with the reference-point method of Refs. [3, 4] is also provided.

All the simulations presented below were done with the modeFRONTIER[17] integration platform for multiobjective and multidisciplinary optimization. A significant advantage of this software is that it enables the automation and execution of complex design simulation process chains represented with graphical workflows.

In Section 2 we summarize the key ideas of the MOUU method used in the paper and introduce the MCDM techniques used in the present study as well as a benchmark of the MCDM methods on mathematical test cases. In Section 3 we describe the structural engineering application and the MO we performed (deterministic, with integrated DM and under uncertainties). *A posteriori* checks on the reliability of the solutions are also presented. Conclusions are discussed at the end.

2 TECHNIQUES

In this section we highlight a few key concepts of the theoretical background behind this work and summarize the main ideas of the computational techniques used in this paper.

2.1 Multi-Objective Optimization under Uncertainty

A general multi-objective optimization problem can be formulated in mathematical terms as (see e.g. Ref. [18])

$$\begin{cases} \text{Minimize} & \mathbf{F}(\mathbf{x}) = f_1(\mathbf{x}), ..., f_I(\mathbf{x}), \\ \text{subject to} & g_j(\mathbf{x}) \leq 0,\ j = 1, ..., J \\ & h_k(\mathbf{x}) = 0,\ k = 1, ..., K \\ \text{with} & x_m \in [x_m^L, x_m^U],\ m = 1, ..., M \end{cases} \tag{1}$$

where $\mathbf{x} = (x_1, ..., x_M)$ represents a vector of M independent input variables x_m with lower and upper bounds respectively equal to x_m^L and x_m^U, $\mathbf{F}$ is a vector of I objective functions f_i, while g_j and h_k respectively are a set of J inequality and K equality constraints that define the feasible design space region (i.e. designs satisfying the constraints). If $I > 1$ and the objectives are in contrast, then the problem is multi-objective and has an infinite set of Pareto optimal solutions. A point $\mathbf{x}^*$ is Pareto optimal if and only if there is no other point $\mathbf{x}$ such that $\mathbf{F}(\mathbf{x}) \leq \mathbf{F}(\mathbf{x}^*)$, and $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$ for at least one function, that is: a point is Pareto optimal if there is no other point that improves at least one of the objectives without deteriorating another. The vector of objectives $\mathbf{F}(\mathbf{x}^*)$ is said to be non-dominated.

Multi-objective optimization problems can be directly solved by means of evolutionary and genetic algorithms. A set of optimal solutions is found starting from an initial set of candidate

solutions (population), which evolve in an iterative process by following mechanisms inspired by biological evolution, like selection, crossover, mutation (see e.g. [18] and references therein).

In MOUU different types of uncertainties can be encountered: random or probabilistic (due to an inherent variability of physical systems and consequently irreducible) and epistemic or imprecise uncertainties (due to a lack of knowledge or information and, in principle, reducible at some stage of the modeling activity) (see e.g. Ref. [1]). Here we consider only the first type. Probabilistic uncertain input parameters are modeled by random input variables following certain probability density functions (PDFs), which represent the probability of occurrence of an event. Because of the input stochasticity, the system response is also stochastic, but its PDF is not known *a priori*. Given that objectives and constraints in optimization problems are often defined in terms of output variables, quantifying the uncertainty on system responses becomes extremely important. This task is accomplished with so-called Uncertainty Quantification (UQ) techniques. Distribution moments, like mean and variance that can be used as robustness measures in RDO, can be estimated with many techniques, e.g. Monte Carlo (MC) or LHS, or the more efficient PCE[14, 19]. The latter is accurate and usually requires a smaller number of function evaluations than the sampling techniques, at least in the case of small-medium problem dimensions. This is a crucial advantage, since computational time is one of the major bottlenecks in design optimization processes. In RBDO instead one is more interested in finding probability tails to determine the failure probability w.r.t. predefined performance or limit state functions (LSFs). The failure probability can be computed with different approaches: (advanced) sampling strategies, FORM and SORM approximations, etc. (see e.g. Ref. [1]). Then it can be minimized or used as a constraint in RBDO problems (see e.g. [1] but also [20, 21] that highlight the effectiveness of using evolutionary algorithms in both single or multi-objective RBDO problems).

For this work we used an OUU method recently introduced by the authors. For a more detailed description, see Refs. [15, 16, 13] and references therein. For each optimization step, a nested UQ flow is executed to compute the statistical output properties to be used as objectives or constraints of the main optimization process. To be specific, in the UQ loop we can compute mean values and standard deviations of output properties with MC, LHS or PCE methods. We can also use the PCE as a stochastic response surface to determine the cumulative distribution function (CDF) and percentile values of output responses that represent probabilistic (or chance) constraints to be optimized in the main loop within a given reliability threshold (with the so-called performance-measure RBDO approach). The use of the PCE for reliability assessments has been growing in recent years (see e.g. Refs. [22, 23, 24, 25]).

For each optimization step, the principal steps in the nested uncertainty quantification flow with percentile calculations are[15, 16]:

- Generation of an MC or LHS sample
- Evaluation of the sample (*real* function evaluations)
- Computation of PCE coefficients via least square minimization
- Estimation of mean and variance
- Generation of an LHS sample for percentile calculations
- Evaluation of the sample (PCE *virtual* function evaluations)
- Determination of CDF and percentiles

The determination of the CDF and percentiles on empirical samples evaluated with the PCE as a stochastic response surface model (*virtual* function evaluation) is clearly an advantage in terms of computational time unlike a pure sampling approach requiring *real* function evaluations, which in real-world applications are often made with calls to finite element or other computationally demanding simulation solvers.

The important parameters for the above-described UQ process are the chaos order k (polynomial degree of the truncated polynomial chaos expansion), the size N of the first sample to determine the PCE coefficient[1], and the size N_{perc} of the sample for the computation of the CDF.

This technique has been benchmarked in Refs. [15, 16, 13, 26] on several test–cases with particular attention to the optimal choice of the sample sizes and k. Our method does not depend on the optimization algorithm used, which can be chosen on the basis of the optimization problem and other necessities. However, evolutionary multiobjective algorithms (e.g. genetic algorithms) are quite beneficial due to their robustness and ability of finding accurate Pareto fronts.

2.2 Optimization with user preferences

One of the major optimization challenges is to generate solutions in a precisely defined region of the Pareto frontier, without wasting computational time to identify the entire set of optimal solutions. For this reason, we tried to directly integrate user preferences (in our case the decision maker, or DM) in the optimization process by implementing an *a priori* approach based on the DM's knowledge of the location of the desirable region. In this way the generated designs can be automatically ordered according to user preferences.

In this paper we studied two different approaches:

NSGA-III, reference point-based algorithm: NSGA-III[3, 4] is a many-objective[2] evolutionary optimization algorithm. NSGA-III does not add any new genetic operator to NSGA-II[27]. Its fundamental quality is the use of reference points that act as attractors driving the optimization towards optimal regions. The *reference points* are objective space points distributed on a grid. They can either be provided by the user or computed by the algorithm. Since the authors of NSGA-III released no official source code, we developed our own prototype following the quoted references. To allow for more freedom in the exploration of *a priori* unknown objective spaces, we chose to provide no explicit reference points. Their number H is

$$H = \binom{M + p - 1}{p}, \tag{2}$$

where M is the number of objectives and p the number of divisions of each objectives (we took $p = 16$).

Since NSGA-III is expected to exhibit better performance on many-objective problems with respect to NSGA-II, we decided to apply it to a structural sizing problem with a discretized Pareto frontier (see Section 3). The expectations were entirely justified.

[1] This sample is used for the least square minimization and can be arbitrarily chosen, with the exception of its minimum size N_{min}, which must be equal to the number of unknown parameters in the PCE, i.e. $N \geq N_{min} = \frac{(k+d)!}{k!d!}$. Thus in the single–variate case the number of coefficients equals $k + 1$, but in the multivariate case it also depends on the stochastic input space dimension d and grows fast as d increases. This is the real bottleneck of this method.

[2] An optimization problem is *many-objective* when the number of objectives $n \geq 3$.

Target point-based algorithm: we modified the NSGA-II algorithm in such a way to ask the user before the start of the optimization to provide a *target point*, i.e. an objective space point that should be reached by the optimizer. Moreover, we modified the solution-sorting mechanism as explained below. The algorithm steps, which initially require an interaction with the user, are the following:

- the user is asked to provide a target point objective values
- the user is asked to provide a tolerance value for each objective in order to define a tolerance hypervolume
- the user is asked to provide a target number of points that the algorithm has to find inside the tolerance hypervolume. Here the interaction with the user ends
- the optimization process starts. It consists in two phases, i.e. reaching the target point and an NSGA-II based optimization for further improvements:
  - In the first phase, to reach the region of the target point, all the solutions generated in the current generation are first ordered using the Pareto sorting criterion according to the value of their absolute distance from the target point. Within each front we apply a modified selection mechanism, based on the so-called controlled elitism[28], which avoids a premature convergence of the algorithm by keeping a percentage of information from lower rank fronts. For each new point, the algorithm checks whether it falls within the tolerance hypervolume. If it does, the target point number counter is decreased. When it becomes zero, the first phase stops
  - In the second phase the relative distance from the target point is used to sort new designs until the specified number of generations is achieved. In our implementation all objectives have to be minimized. Hence, a solution with a negative relative distance is better than a solution with a positive distance.

The target point initially drives the optimization. In particular, if the target point belongs to the Pareto front, then in the second phase the algorithm will start populating the front further. We developed this strategy to prevent the natural tendency of EA evolution to be driven by the simplest objectives rather than those in which the DM is most interested.

We believe that there is a slight but important difference between target and reference points. A target point is a point that the DM wants to reach with the optimization, whereas reference points are points distributed in the objective space that determine only the direction of the optimization, i.e. they have the role of attractors.

These two methods are not mutually exclusive. In fact, the target point is useful when the user wishes to find more points in a "gap" region, while our use of NSGA-III, without any provided reference points, helps finding the whole Pareto frontier.

More in general, within the Multi-Criteria Decision Making field different approaches to design optimization problems are possible as described in Ref. [29]:

1. $a\ priori$: preferences are asked before the optimization;
2. $a\ posteriori$: preferences are expressed after the optimization;
3. interactive: the optimization algorithm asks the DM to express their preferences during the optimization process.

Historically *a posteriori* approaches have been the standard choice for decision making: the DM runs the optimization and then uses MCDM techniques to find the best Pareto front solution in accordance with their preferences. However, *a priori* and (above all) interactive methods have been recently stirring up great interest in the MCDM community.

We believe that interactive methods, like the one presented in Ref. [5], are in principle the most accurate, because the algorithm is able to calculate the most representative value function with respect to the user preferences. DM preferences are specified at each interaction. The more frequent the interactions, the more accurate the algorithm. However each interaction requires a lot of effort from the DM's side, since, for example, an optimization with many generations implies many interactions. In our opinion, this is the most important disadvantage of interactive approaches. Hence we decided to investigate the above-described *a priori* approaches to reduce the DM effort and stress.

2.3 Mathematical test problem

Both the target point-based and the reference point-based algorithms are prototypes. In order to see differences between them, we tested them on two mathematical benchmark problems: the Constrained Test Problem 7 (CTP7)[30] and the unconstrained Zitzler Deb Thiele 1 (ZDT1) problem[31].

The first problem is construed as follows:

$$\textbf{CTP7:}\begin{cases}\text{Minimize } f_1(\mathbf{x}) = x_1 \\ \text{minimize } f_2(\mathbf{x}) = g(\mathbf{x})\left(1 - \dfrac{f_1(\mathbf{x})}{g(\mathbf{x})}\right) \\ \text{subject to } c(\mathbf{x}) = \cos(\theta)(f_2(\mathbf{x}) - e) - \sin(\theta) f_1(\mathbf{x}) \geq \\ \quad a|\sin(b\pi(\sin(\theta)(f_2(\mathbf{x}) - e) + \cos(\theta) f_1(\mathbf{x}))^c)|^d\end{cases} \tag{3}$$

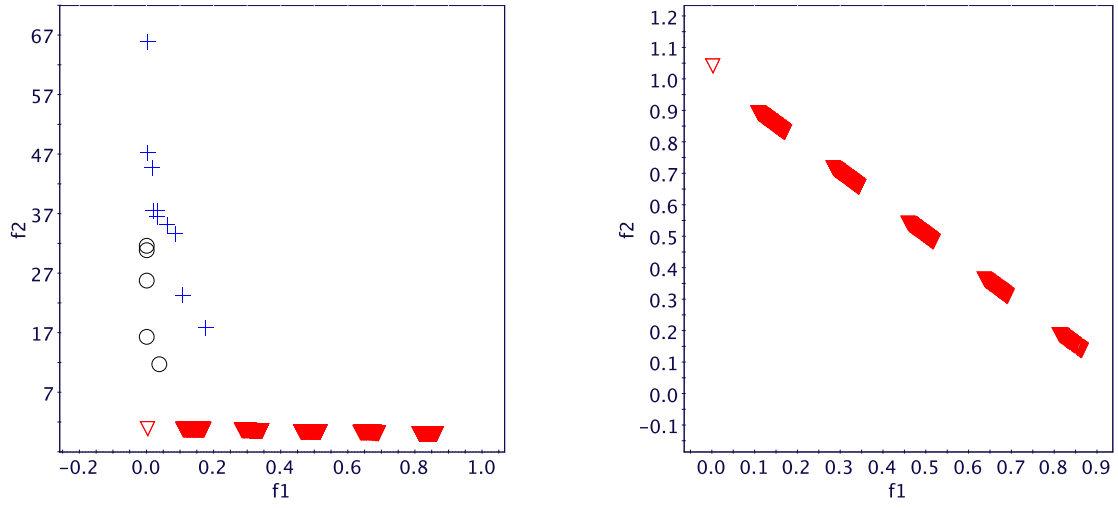
where $\theta = -0.05\pi, \quad a = 40, \quad b = 5, \quad c = 1, \quad d = 6, \quad e = 0$. CTP7 has a discontinuous Pareto frontier.

We performed three optimization runs: one with the standard NSGA-II algorithm and the other two with the target-point and the reference-point algorithms respectively. In each simulation we started from an initial population of 100 individuals and set the number of generations to 1000. We used crossover and mutation probabilities respectively equal to 0.9 and the reciprocal of the number of input variables, and distribution index values set to 20.

On Figure 1 we can see the efficiency of the reference-point NSGA-III algorithm to find the entire Pareto frontier owing to the ability of the reference points to attract the optimization towards the disconnected Pareto front regions. On the other hand NSGA-II and the target-based algorithm are not able to find any points on the true Pareto front, even though we set a target point located on the exact Pareto front. For this problem, the target-point algorithm is still influenced by NSGA-II.

The second test problem is unconstrained and has a continuous Pareto front. It is construed as follows:

$$\textbf{ZDT1:}\begin{cases}\text{Minimize } f_1(\mathbf{x}) = x_1 \\ \text{minimize } f_2(\mathbf{x}) = g(\mathbf{x})\left(1.0 - \sqrt{\dfrac{f_1}{g}}\right) \\ \text{with } g(x_1, x_2, \cdots, x_n) = 1.0 + \dfrac{9.0}{n-1}\displaystyle\sum_{i=2}^{n} x_i\end{cases} \tag{4}$$



(a) (b)

Figure 1: (a): Pareto front designs found with a deterministic optimization of the CTP7 problem with NSGA-II (circles), NSGA-III (downward triangles) and the target-point (plus symbols) algorithm described in Section 2.2. (b): detail of the NSGA-III Pareto front.

We used ZDT1 to test the algorithms with a smaller number of generations in order to see if the target-point method is able to focus on the desired region from the very beginning. We performed again three optimization runs with the same three algorithms and parameters as for CPT7, but with a number of generations equal to 40. Results are shown on Figure 2. We started with the original NSGA-II simulations and we noted that a gap appears in the Pareto frontier. With the target-point algorithm we tried to drive the optimization precisely towards the middle of the gap region (target point values set to the middle of the gap, tolerance values around 20% and maximum points in the tolerance hypervolume equal to 50). As a result, the target point-based algorithm was able to find several designs in the desired region at the expense of the exploration of the Pareto frontier tails. NSGA-III performed slightly better than NSGA-II both in the upper left tail of the Pareto region and in the gap region, but it was not able to extensively explore the right bottom corner with the given number of generations. This is not entirely unexpected because that region is influenced by the most complex objective of the problem (minimization of f_2) and NSGA-III does not use the crowding distance strategy to increase the population diversity. Finally, we observe that both new algorithms present small convergence problems (the attainment of the true Pareto front is not perfect), which can be overcome by increasing the number of generations.

A comparison of the performance of these algorithms on a many-objective optimization problem is provided in Section 3.

3 STRUCTURAL ENGINEERING APPLICATION

In this paper we further analyzed a structural engineering design problem, presented in Ref. [13], with the aim of showing the effects of the dynamic use of MCDM algorithms on the optimal solutions found during the design optimization phase (with or without uncertainty).

We considered the same building structure and model as in Ref. [13] with minor changes. It

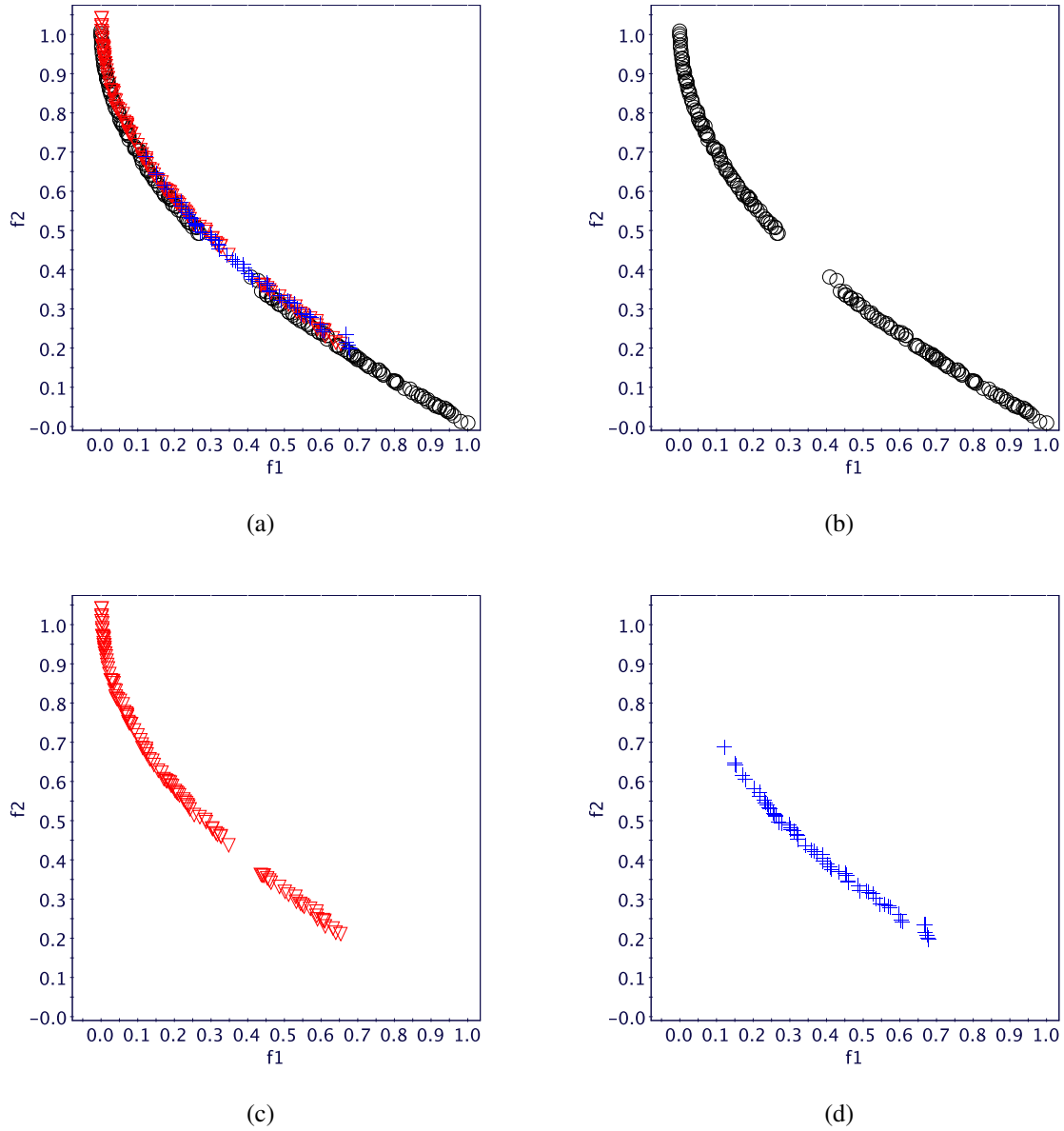


Figure 2: Pareto front designs found with a deterministic optimization of the ZDT1 problem with NSGA-II (b), NSGA-III (c) and the target-point algorithm described in Section 2.2 (d). All the Pareto frontiers are shown on the same graph in (a).

is a two-storey building formed by four 3-bay frames made of reinforced concrete[3] beams and columns, and ceilings/floors made of reinforced concrete joists and hollow flooring blocks. The three-dimensional building frame is shown on Figure 3.

As the location we chose the Italian town of Forni di Sopra (UD) that is characterized by a maximum horizontal ground acceleration $a_g = 0.175g$ and an elevation greater than 1500 m above mean sea level. This enables the statistical analysis of the snow load present on the building flat roof (the worst possible case in the presence of snow) on a site with significant seismic hazard. Wind loads are not relevant for this location and type of structure.

[3]Concrete of type C25/30, i.e. confined characteristic compressive strength $f_{ck} = 25$ MPa and design compressive strength $f_{cd} = 14.17$ MPa.

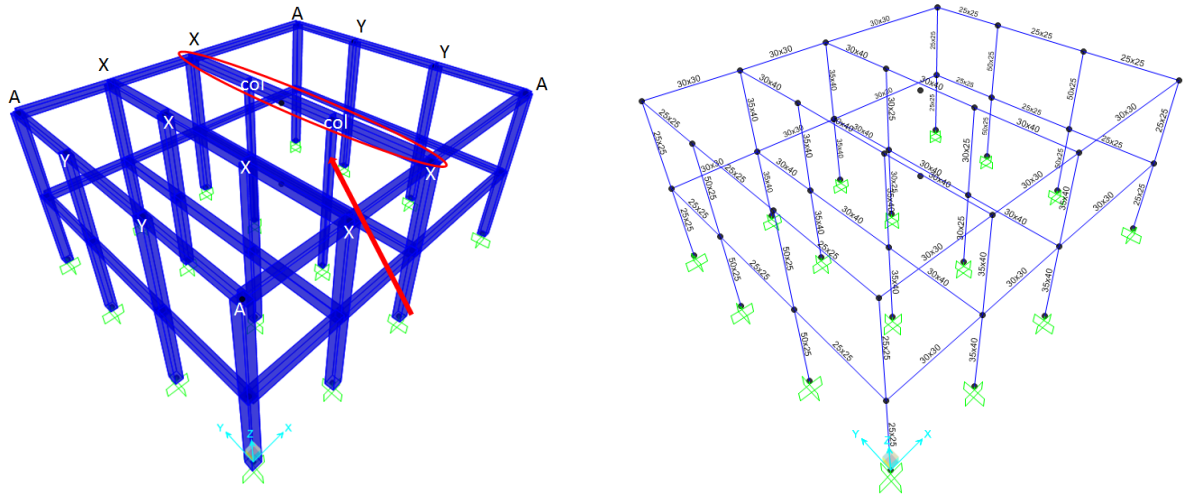


Figure 3: 3D view of the building frame. On the left: extruded view (the different labels denote the different types of column sections. The investigated most loaded beam and column are respectively circled and marked with an arrow). On the right: standard view, with an example of design sections.

To reduce the computational effort, we fully designed only the most loaded coverage beam and the most stressed column (out of the four columns supporting the beam) of the structure (see Figure 3) with the aim of determining the optimal size of their sections and reinforcement content to achieve minimum weight and seismic displacement while respecting the constraints required by the Italian law and considering variations in other column sections.

The stresses for the vertical loads and seismic actions were simulated with SAP2000[32], a finite element analysis general-purpose civil-engineering software. We considered the envelope[4] of vertical load (permanent, variable and snow) combinations at ultimate limit state (ULS) with the seismic combinations in the x and y directions as required by the Italian law for construction engineering (NTC2008: DM 14.1.2008).

For the seismic displacements we used the serviceability limit state (SLS) with an "SLD" (i.e. "Damage Control" performance level) seismic elastic response spectrum.

The framed supporting structure composed of beams and columns was represented with one-dimensional finite elements. Since the perimeter walls were not directly included in the structure model, we introduced them in terms of an additional mass contribution for the seismic actions in the SAP model, also to mimic the effect of any masses not explicitly taken into account. However, the results illustrated below are only intended as guidelines for designers, since we did not consider the shear reinforcements of the structure. In fact, our work was mainly focused on the applicability (and effect) of MOUU and real-time integrated MCDM techniques to structural design optimization problems.

The following sections illustrate the details of the optimization problem, simulations and results.

[4]The envelope merged four terms: two vertical load combinations (one with a dominant variable load and the other with a dominant snow load) and two ULS seismic load combinations (with seismic components $E_X + 0.3E_Y$ and $E_Y + 0.3E_X$, where E_Y and E_Y denotes respectively the seismic action along x and y). Here, we used the seismic project response spectrum "SLV" (corresponding to a "Life Safety" performance level) for the town considered.

3.1 Optimization problem and workflow

To illustrate the effectiveness of the above-illustrated MCDM algorithms on many-objective problems we chose *three objectives*: weight minimization of the most loaded beam and stressed column (see Figure 3) and seismic displacement minimization in the x direction (for a "LSD" seismic spectrum with dominant action along x) by varying the sections of the two elements and their content of reinforcing bars, as well as the sections of all the other columns.

We considered *twelve input variables* and *three constant input parameters*. The input variables are: (1) the most loaded beam section, (2) four types of column sections (denoted with different labels on Figure 3), (3) the three longitudinal reinforcements in the three lower spans of the beam, (4) the three upper longitudinal reinforcements in the three spans of the beam, and (5) the longitudinal reinforcement of the column. The reinforcements are given in terms of steel sections and their number. All the mentioned inputs have discrete values (from a list of sections available in commercial catalogs). The three constant input parameters are: (1) the snow load value provided by the law for 1500 m above sea level, equal to 5.83 kN/m^2 - this value, like the other vertical loads of the two floors, was transformed into linear load and applied to the beams; (2) the additional mass term of 600 kN; (3) the structure factor q = 3.9 (necessary for the conversion from the elastic to the project spectrum) as expected for "Frame structures with multiple floors and spans, of class B" (see NTC2008: DM 14.1.2008).

The most loaded beam and column sections and reinforcement contents were subject to *fourteen input constraints* as required by the law (see NTC2008, paragraphs 7.4.6.2.1 and 7.4.6.2.2).

In output, we monitored several properties, such as the beam and column weights, their total weight, seismic displacements, first-mode eigenperiod, bending moments, etc.

We also considered *nine output constraints*: six ULS compliance checks of load and resistance effects on the beam (beam moment resistances greater than stress moments), one constraint on the column (compliance with the limits on compressive and flexural strengths due to the loads) and two constraints for the seismic displacements along x and y (relative inter-store displacement in both directions smaller than the maximum allowed displacement).

Given the high number of constraints present on both input (geometric constraints for section sizes and reinforcing bar content) and on output variables (seismic displacements, constraints on the most loaded beam and column loads), this is a hyper-constrained problem. The presence of discrete input variables makes its solution even harder from a mathematical point of view. Thus, this example is an interesting benchmark for the performance of the algorithms described in Section 2 on complex and realistic structural design optimization problems.

All simulations were performed with the modeFRONTIER integration platform and optimization software. As said in the previous paragraph, we computed the stresses caused by vertical and seismic loads with SAP2000. The SAP2000 inputs are the structure geometry and constraints, the frame sections, the linear loads, the seismic analysis type, etc. modeFRONTIER was integrated with SAP2000 by means of an Excel file, which controlled the SAP2000 model with a Visual Basic macro. The macro was used for a dual purpose: to update the SAP2000 model parameters (the central beam and column variable sections) with the values provided by modeFRONTIER for each proposed configuration and to extract the output variable values calculated by SAP2000 and write them in the Excel file to be read by modeFRONTIER.

Figure 4 shows the modeFRONTIER optimization workflow. From top to bottom we can follow the optimization data flow: at the top we find input variables, constant parameters and constraints, whereas at the bottom we find the monitored output variables, objectives and constraints on the output variables. The green icons (square boxes with an incoming arrow) in the

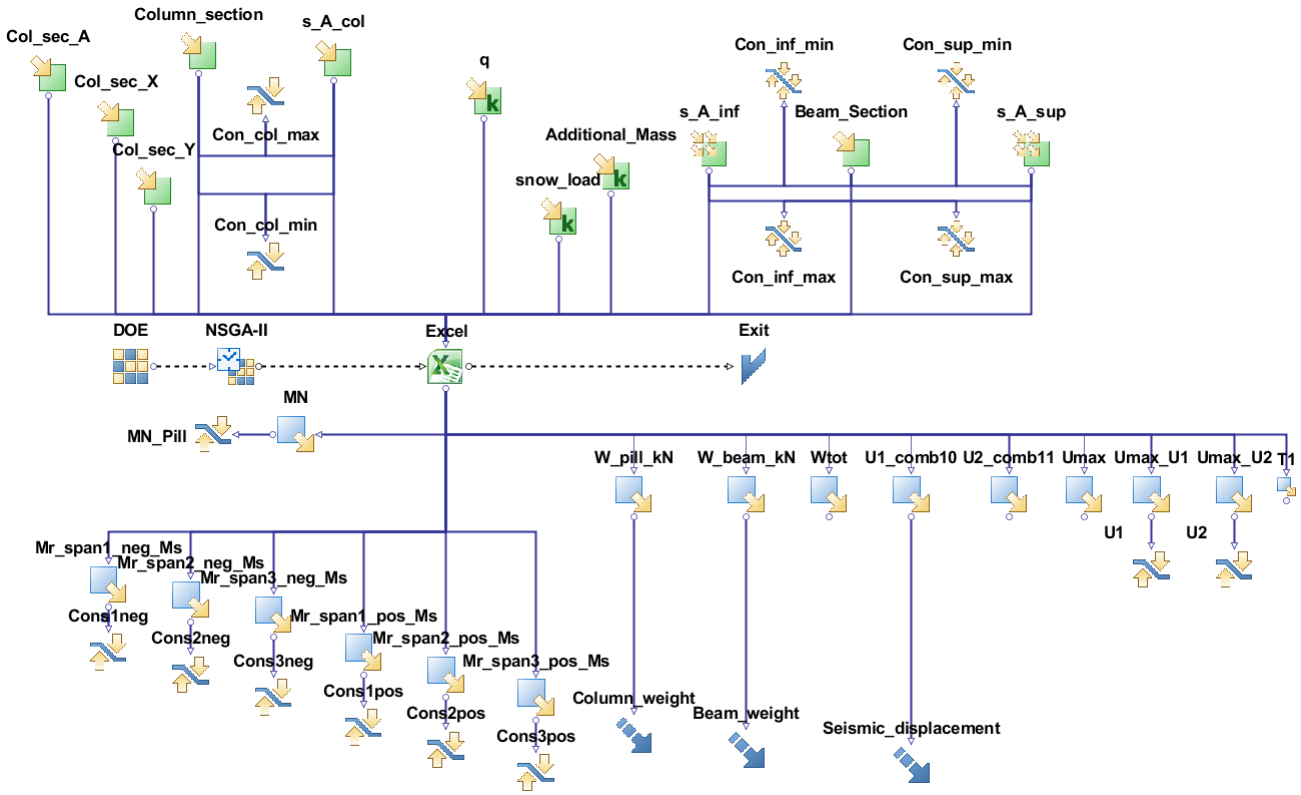


Figure 4: Optimization workflow.

top part of the figure represent the input variables and constants, while the blue icons (square boxes with an outcoming arrow) at the bottom represent the output variables. Constraints are represented by opposing arrows separated by a line, while objectives are represented by an arrow icon. In the center, from left to right we find icons representing the logic flow of the process, with the generation of the initial sample by means of a design of experiment algorithm node (DOE), followed by the optimization algorithm node, an Excel node (interface between modeFRONTIER and SAP) and the logic end of the process.

3.2 Simulation details

We ran the following simulations:

1. deterministic optimization with NSGA-II
2. deterministic optimization with the target point-based algorithm (described in Section 2.2) starting from the deterministic Pareto front of point 1.
3. stochastic optimization starting from the Pareto front found in point 2.
4. stochastic optimization with the target point-based algorithm to improve the stochastic Pareto front found in 3.
5. reliability assessment of representative solutions found in the previous points
6. deterministic optimization with the reference point-based algorithm (i.e. NSGA-III based, see Section 2.2) starting from the Pareto front found in 1.

7. encouraged by the results obtained in the previous point, we used the NSGA-III based algorithm starting from the same random DOE as in 1. to compare its performance against NSGA-II

More details are provided in the following paragraphs.

3.2.1 Deterministic optimization with standard NSGA-II

The problem described in Section 3.1 was optimized with the NSGA-II genetic algorithm, with an initial input population of 360[5] candidate designs (generated with a uniform random algorithm) crossover and mutation probability values respectively equal to 0.9 and the reciprocal of the number of input variables, and an allowed budget of 50 generations.

During the optimization, many unfeasible designs (i.e. designs that violate at least one constraint) were found. Remarkably, no seismic displacement constraints were violated within the explored input variable ranges. The most critical constraints appeared to be those on the most loaded beam.

The Pareto front is shown on Figure 5 (cross symbols). By inspecting the variable values of Pareto front designs, we noticed that all the designs had the same beam section, a square section of dimensions equal to 0.30×0.30 m^2. The range of the column section spanned over almost all the available values in the catalog list, but the majority of designs preferred either square columns or along the x direction (with only few designs preferring the rectangular section directed along y). The first mode eigenperiod appeared to be almost perfectly linearly correlated with the seismic displacement objective.

3.2.2 Deterministic optimization with target-point integrated MCDM

We decided to apply the target point-based algorithm to drive the optimization toward low values of the beam weight, or to be specific toward the objective space region close to beam and column weight and seismic displacement values of respectively 26.35 kN, 14 kN and 0.0203 m^2. We used a maximum number of points inside the tolerance hypervolume equal to 20.

We started the algorithm with an initial population composed of the Pareto front designs found in Section 3.2.1 and other randomly selected feasible designs from that previous run in order to reach 200 elements (since the starting points were already good, it was not necessary to apply the rule of thumb described in footnote 5).

The objective values of the Pareto front designs found after 45 generations are shown on Figure 5 (empty squares), where they are compared with the Pareto front designs found with the standard NSGA-II algorithm in Section 3.2.1 (cross symbols). It can be observed that the target-point algorithm found at least three new points in the target region of the objective space. These objective space points correspond to several designs in the input variable space that differ for their reinforcement content. The new designs tended to have square column sections, while they all had a beam section equal to 0.30×0.30 m^2 as in Section 3.2.1.

Once again, no violation of the seismic displacement was found. The most violated constraints were still those on the beam compliance checks.

[5]This high value was chosen by a rule-of-thumb according to which the input population size should be greater than the number of input variables multiplied by 2 and by the sum of the number of objectives and constraints.

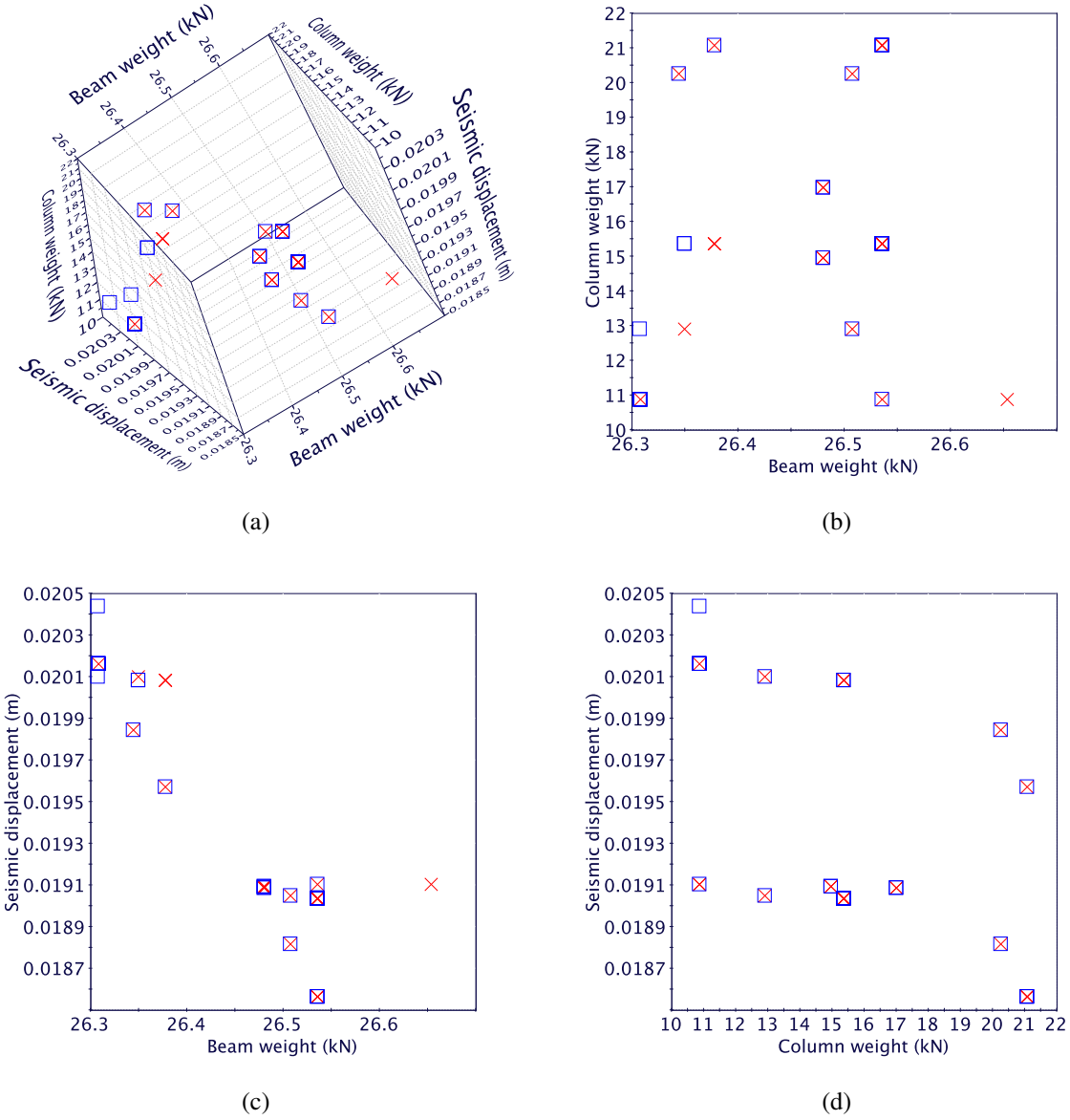


Figure 5: Pareto front designs found with a deterministic optimization with NSGA-II (cross symbols) and with the target-point algorithm described in Section 2.2 (empty squares).

3.2.3 Optimization under uncertainty

To improve the reliability of Pareto optimal designs w.r.t. to the most violated constraints found in the previous sections, we decided to perform a reliability-based design optimization with the technique explained in Section 2.1.

In all the OUU simulations we considered the snow load and the additional mass as uncertain parameters respectively defined by an exponential PDF, with location parameter 5.83 and scale parameter 0.5, and a log-uniform PDF, with location parameter equal to 600 kN and scale parameter of 60 kN (i.e. the underlying uniform distribution is in the range from 600 kN to 660 kN).

To determine the optimal PCE sample size we performed some uncertainty quantification checks with sizes $N = 11,\ 13,\ 20,\ 30,\ 50,\ 100$ and PCE degree $k = 3$ and compared the results obtained with LHS data found with sample sizes $N = 13,\ 20,\ 30,\ 50,\ 100,\ 200,\ 500,\ 1000$.

Figure 6 shows results for two of the most important constraints on the beam, denoted with Cons1neg and Cons3neg in the workflow of Figure 4. As is evident, for these properties the PCE reached convergence with the smallest sample size for the mean values and the second smallest for the standard deviation, at variance with LHS data, which converged for $N = 200$ or 500. Therefore, we chose to perform the stochastic optimization with a PCE sample of 13 designs.

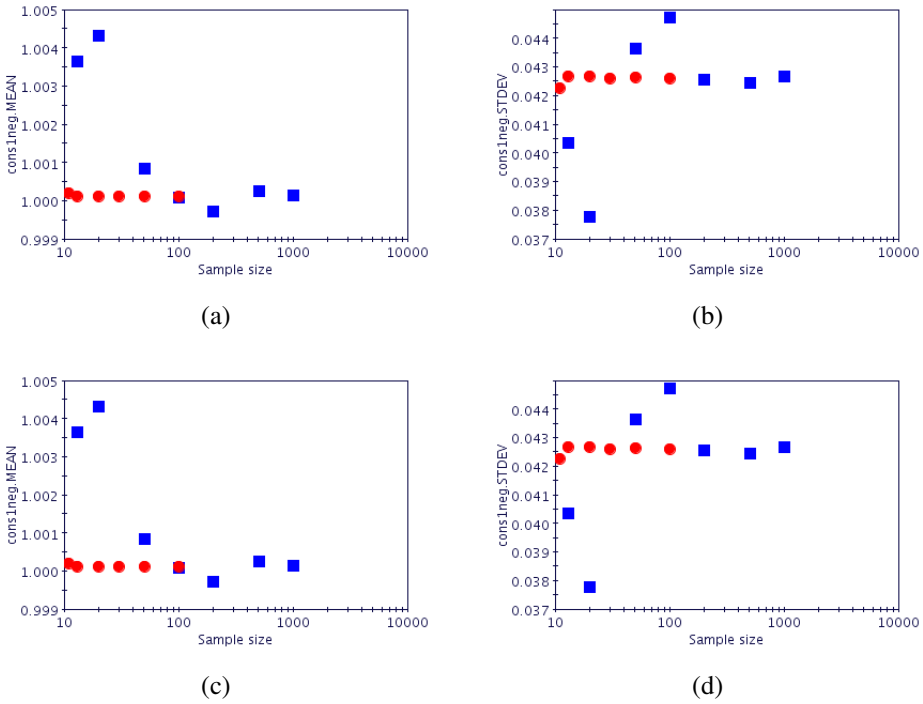


Figure 6: (a) and (b): estimates of the mean values and standard deviation of ultimate limit state compliance checks for the first span of the most loaded beam; (c) and (d): same properties for the third span. Mean values are shown on the left, while standard deviations on the right. PCE data are represented with circles, while LHS data with squares.

We considered the same optimization problem as in the previous sections, but for the introduction of probabilistic constraints. We aimed at achieving a reliability of 99.9%. Therefore we computed the 99.9-th percentile of the six constraints on the beam (called Cons1pos, Cons1neg, Cons2pos, Cons2neg, Cons3pos, Cons3neg in the workflow of Figure 4). All other constraints were treated as deterministic and checks on the design feasibility w.r.t. them were made *a posteriori*.

As an optimizer, we used the NSGA-II algorithm with the same parameters as in the deterministic simulations and an initial population of 200 designs composed of the Pareto front designs found in Section 3.2.2 plus other randomly selected feasible designs. For each design we computed its statistical properties with a third-order PCE determined on a sample of 13 elements generated around the current design by a LHS based on the PDFs of the snow load and additional mass parameter. The computation of these samples requires real function evaluations (i.e. calls to the SAP software). Percentiles were determined with negligible computational cost on new LHS samples with 1000 designs evaluated with the so determined PCE functions. We ran 8 generations of NSGA-II, which correspond approximately to the total number of real function evaluations made in the deterministic optimization in Sections 3.2.1 and 3.2.2.

In the RBDO run, the seismic displacement constraint along the x direction was occasionally violated, due to the output dispersion introduced by the input stochasticity (in practice, violations sometimes occurred in the internal samples of the 13 designs used for the determination of the PCE). However, no design in the samples drawn around the Pareto front designs turned out to be unfeasible.

The column constraint was also occasionally violated for geometric reasons. In fact, in most cases, if a design was violated, all the designs in the sample generated to compute the PCE were also unfeasible.

The stochastic Pareto front is shown in Figure 7 (empty diamonds), where it is compared with the Pareto front found with the target-point deterministic optimization in Section 3.2.2 (empty squares). Like the deterministic optimization, all the OUU Pareto front designs have the same beam section, which this time is rectangular and equal to 0.30×0.40 m^2. Unlike in the case of the deterministic simulations (especially the one of Section 3.2.1) where almost the entire column section range was explored, here most designs have square column sections (of dimension 0.25×0.25 m^2) and only few of them are rectangular (directed along y and of dimension 0.30×0.25 m^2). Apparently the reliability requirement has driven the optimization to this input space region. In the objective space it is striking that the reliability has been reached at the expense of the beam weight (determined not only by the beam section but also by its reinforcement content), which notably increased in the stochastic Pareto front.

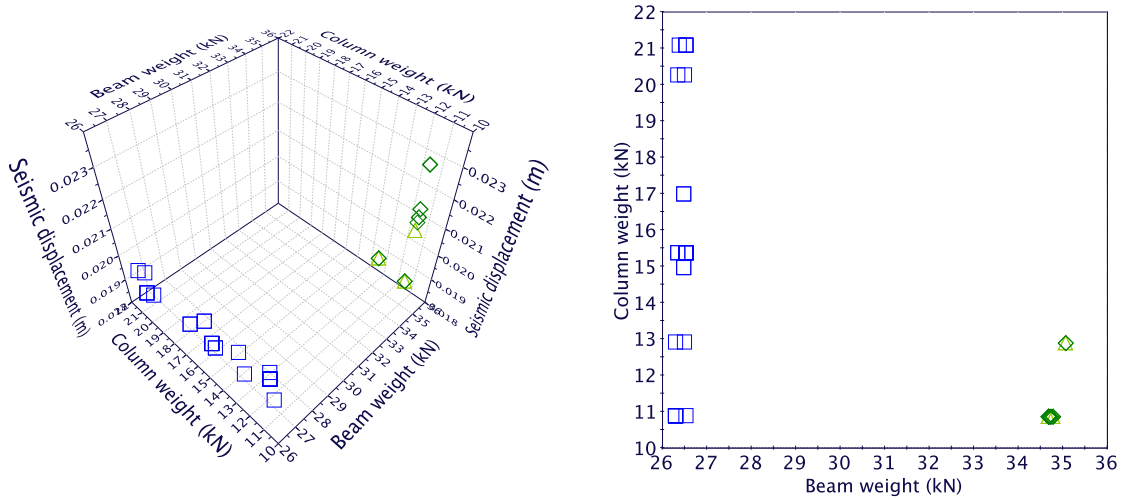
3.2.4 Stochastic run with target-point algorithm

To drive the optimization in the gap which appears in the stochastic Pareto front in the seismic displacement range $[0.0195, 0.0215]$ m (see Figure 7), we ran the target point-based algorithm on the OUU problem for 8 generations starting from an initial population of 200 designs (composed of the Pareto front designs found in Section 3.2.3 plus other randomly selected feasible designs). The new stochastic Pareto front (empty triangles) is compared with the stochastic Pareto front found with the standard NSGA-II algorithm (empty diamonds) and with the deterministic target-point results (empty squares) on Figure 7. The Pareto front found with the target point optimization is located in a smaller region w.r.t. the other fronts. The new front presents only one new seismic displacement value in the desired region, given by two different designs in the input variable space.

3.2.5 Reliability assessments

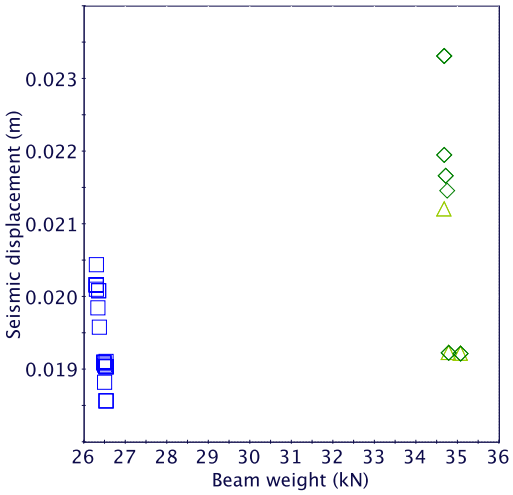
To double check the effectiveness of our MOUU method for finding reliable optimal solutions, we made a posteriori uncertainty quantification checks. In this section we show outcomes for selected solutions, though the trend found can be considered generally valid. Table 1 contains input/output values of four representative optimal designs found in Sections 3.2.1 (**DN**), 3.2.2 (**DT**), 3.2.3 (**SN**) and 3.2.4 (**ST**). The objective-space representation of these points is shown on Figure 8.

We generated samples of 200 points centered on the selected designs with the same uncertainty distributions as those considered in Sections 3.2.3 and 3.2.4. Then we checked whether any of the constraints was violated. The amount of designs violating some of the most critical constraints can be seen for instance by looking at the empirical PDFs of those constraints computed on the generated samples. An example for each representative design is shown on Figure 9. PDFs for the deterministic Pareto-optimal designs are shown at the top of the figure.

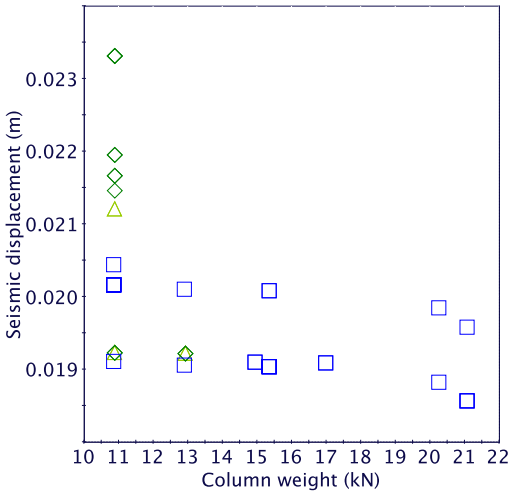


(a)

(b)



(c)



(d)

Figure 7: Pareto front designs found with a stochastic optimization with NSGA-II (empty diamonds), stochastic optimization with the target-point algorithm (empty triangles) and results from Section 3.2.2 (empty squares).

For these designs the empirical PDFs of the most violated constraints are shown. The constraint violation is indicated by the light-colored (orange online) part of the histograms. PDFs for the stochastic optimization optimal solutions are shown at the bottom. Since no constraint violation was observed in the samples around **SN** and **ST**, we randomly chose to show Cons2neg for **SN** and Cons3neg for **ST**. This confirms our expectations. Optimal solutions found with deterministic MO methods (with or without MCDM) have a non-negligible probability to violate critical constraints (indeed the largest part of the empirical PDFs lies below the constraint threshold, which equals 1). On the other hand, optimal solutions found with the RBDO methods have a negligible probability of violating the chance constraints.

	DN	**DT**	**SN**	**ST**
Beam_section (m^2)	0.30×0.30	0.30×0.30	0.30×0.40	0.30×0.40
Column_section (m^2)	0.35×0.25	0.25×0.30	0.25×0.25	0.25×0.25
s_A_sup (1) (m^2)	0.001521	0.001357	0.001521	0.001357
s_A_inf (1) (m^2)	0.000763	0.000603	0.000763	0.000603
s_A_sup (2) (m^2)	0.001357	0.001357	0.001357	0.001357
s_A_inf (2) (m^2)	0.001257	0.00076	0.000804	0.00076
s_A_sup (3) (m^2)	0.001257	0.001257	0.001257	0.001257
s_A_inf (3) (m^2)	0.000603	0.000603	0.000763	0.000603
s_A_col (m^2)	0.002714	0.002281	0.001885	0.001885
Col_sec_A (m^2)	0.40×0.40	0.40×0.40	0.40×0.40	0.35×0.35
Col_sec_X (m^2)	0.25×0.50	0.35×0.40	0.25×0.50	0.35×0.40
Col_sec_Y (m^2)	0.40×0.35	0.40×0.35	0.40×0.35	0.40×0.30
T1 (s)	0.598	0.628	0.603	0.667
Beam_weight (kN)	26.48	26.307	34.77	34.659
Column_weight (kN)	14.951	12.911	10.871	10.871
Seismic_displacement (m)	0.019	0.02	0.019	0.021
MN	0.751	0.482	0.514	0.493
Cons1neg	1.157	1.069	1.623	1.504
Cons2neg	1.027	1.006	1.436	1.467
Cons3neg	1.0004	1.029	1.417	1.458
Cons1pos	1.217	1.066	1.522	1.347
Cons2pos	1.853	1.094	1.613	1.469
Cons3pos	1.026	1.139	1.617	1.445
U1 (m)	0.005	0.004	0.005	0.004
U2 (m)	0.006	0.006	0.006	0.007

Table 1: List of some input and output values for the designs chosen for the reliability checks. **DN** is a Pareto optimal design found with the deterministic NSGA-II optimization of Section 3.2.1, **DT** is one of the target point solutions of Section 3.2.2, **SN** was found with the NSGA-II MOUU of Section 3.2.3 and **ST** with the refinement target point stochastic optimization in Section 3.2.4. In the first column, input variable names are written in italics. From top to bottom: beam section, column section, reinforcement content of the three beam bay frames ("*s_A*" stands for "steel area"), reinforcement content for the column, sections of the other type of columns (see labels in Figure 3), first mode eigenperiod, the three objectives, and the constraints (from MN to U2: the constraint on the column, the six constraints on the beam and the seismic displacement constraints).

3.2.6 Deterministic optimization with the reference-point algorithm based on NSGA-III

We used the structural optimization problem described in the previous sections to test the reference-point algorithm based on the NSGA-III algorithm[3, 4] (see Section 2.2). This algorithm is expected to perform better than NSGA-II on many-objective optimization problems.

We began by applying the reference-point algorithm to refine the results found with the deterministic optimization with the standard NSGA-II algorithm in Section 3.2.1. We started from an initial population composed of 200 elements (the NSGA-II Pareto front designs and other randomly selected feasible designs) and ran the reference point algorithm for 20 generations (with the same crossover and mutation probability parameters as in Section 3.2.1).

On Figure 10 we compare the deterministic Pareto front designs found with the reference-point algorithm (downward triangles) and the Pareto front found with the target-point algorithm

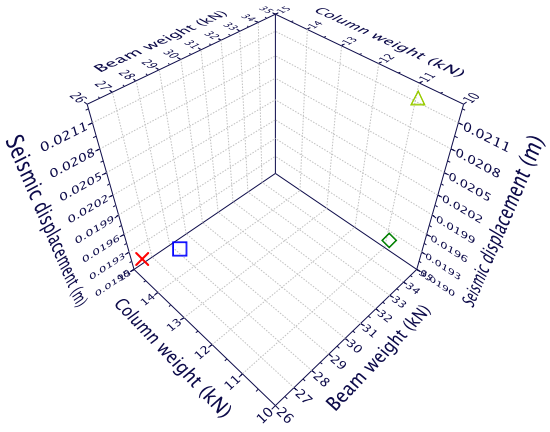


Figure 8: Designs selected for the reliability checks. The cross symbol was found with the deterministic optimization with NSGA-II (**DN** in Table 1 notation), the empty square with the deterministic target point-based optimization (**DT**), the empty diamond with the stochastic optimization with NSGA-II (**SN**) and the triangle with the stochastic target-point optimization (**ST**).

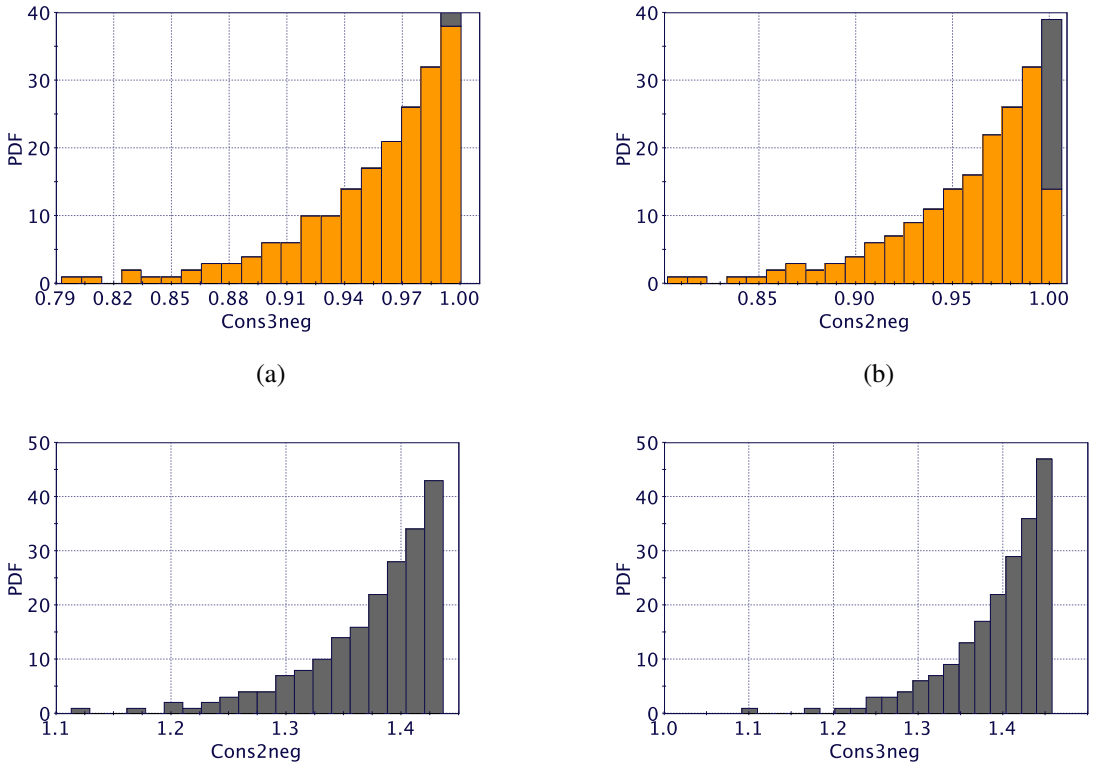


Figure 9: Empirical probability density functions (PDFs) of critical output constraints of selected designs. (a): **DN**, (b): **DT**, (c): **SN**, (d): **ST**. The light-colored (orange online) boxes denote violated constraint values.

in Section 3.2.2 (cross and square symbols, where the crosses represent the points already found with the initial NSGA-II simulation in Section 3.2.1 and the squares the new designs found with

the target-point refinement). The reference point algorithm was capable of finding a well distributed Pareto front. This illustrates well the difference of aims of the target-point algorithm (that drives the simulation process towards a target region) and the reference-point algorithm. It also confirms the good behavior of NSGA-III on many-objective problems. The reference-point algorithm was capable of finding some solutions with better values for the beam weight objective than NSGA-II. An inspection of the input variable values of the reference-point algorithm optimal solutions revealed that all designs had the same beam section as the solutions obtained with the other deterministic optimization runs (hence the smaller weight was obtained with a smaller reinforcement content). However, most designs have either square or rectangular column sections along x.

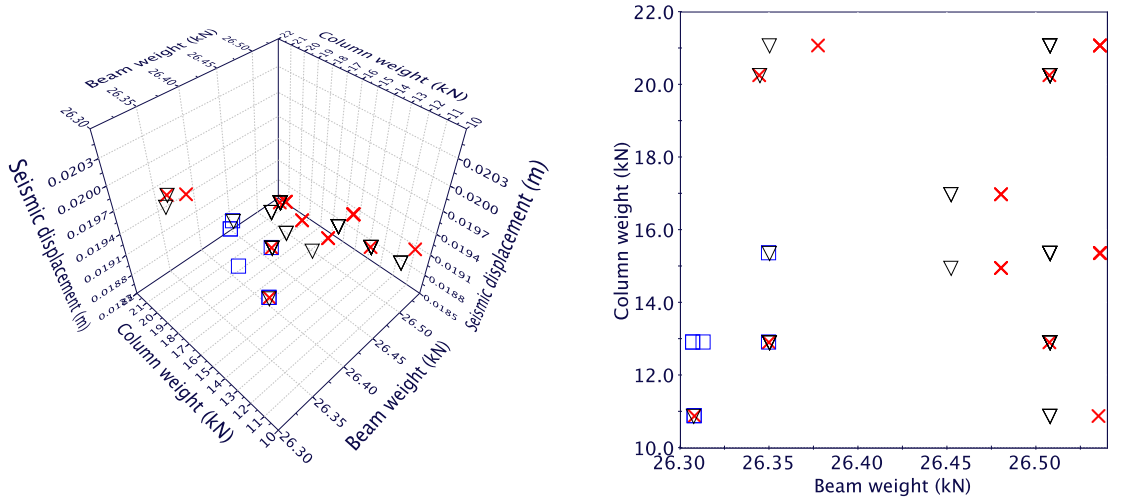
Encouraged by the good results, we decided to rerun the reference-point algorithm from scratch to compare its performance with the deterministic optimization made with NSGA-II and check whether the results shown on Figure 10 depended on the already good initial population used for that run. We started from the same random DOE used for NSGA-II in Section 3.2.1 and ran the algorithm for the same number of iterations, i.e. 50 generations. Results are shown on Figure 11. It is even clearer from these figure than from Figure 10 that the NSGA-III based algorithm is capable of finding nondominated solutions with much smaller weight values for the beam. This demonstrates the very good performance of the reference-point method for this problem.

An analysis of the input variable values of the Pareto front designs found with this simulation from scratch shows that other beam sections were taken into consideration as well (0.30×0.25 m^2 and 0.35×0.25 m^2). However, there seemed to be no preference between the different types of column sections.

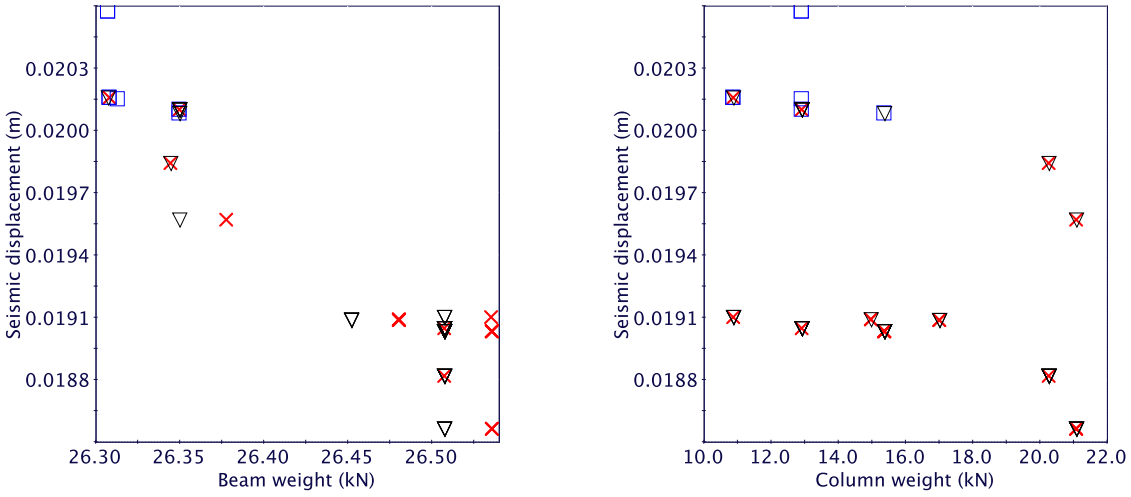
4 CONCLUSIONS

In this paper we have addressed a structural optimization problem by combining recently developed strategies for multi-objective optimization under uncertainty and multi-criteria decision making incorporated in the optimization process following the *a priori* approach. A new decision making strategy based on target-points has been developed and benchmarked in combination with the NSGA-III algorithm (based on reference points) of Ref. [3, 4] on mathematical test-cases. We applied these strategies to a real-world case study, i.e. the sizing optimization of a three-dimensional building structure under seismic and snow loads. Many-objective deterministic and stochastic optimizations were performed, both with and without the integrated user preference methods.

The different strategies considered confirmed our expectations even on the structural optimization case, which is very challenging due to its discrete input variable space and a consequently discretized Pareto front. The MOUU provided reliable solutions with a negligible probability of violating critical chance constraints unlike the solutions obtained with the deterministic optimization. The target-point algorithm was used to refine the NSGA-II results and to drive the optimization towards specific regions of the objective space, while with the reference point method we obtained well-distributed optimal solutions, with some striking improvements over the standard NSGA-II algorithm. However, both user-preference algorithms are prototypes and there is still space for further improvements, from both the algorithmic and the methodology points of view. Our goal is to investigate the Pareto frontier as thoroughly as possible to achieve better decision making using the new solutions with the aim of reducing the computational cost which arises from the exploration of undesired objective space regions, and concentrating on specific regions. Another important line of investigation is the development of techniques to



(a) (b)



(c) (d)

Figure 10: Comparison of the Pareto front designs found with the target-point algorithm and the reference-point (NSGA-III based) algorithm described in Section 2.2. Reference-point results are represented by empty downward triangles. Target-points results are represented by cross symbols (which are points already found with the original NSGA-II algorithm, before launching the MCDM optimization) and empty squares (which represent the *new* designs found with the target point algorithm).

efficiently determine the PCE coefficients also in the case of a large number of uncertain input variables, like e.g. Ref. [24].

Acknowledgments

L. Rizzian and ESTECO S.p.A. gratefully acknowledge CSI Italia s.r.l. for providing the SAP2000 license. We thank Maja Engel (ESTECO S.p.A.) for the manuscript revision.

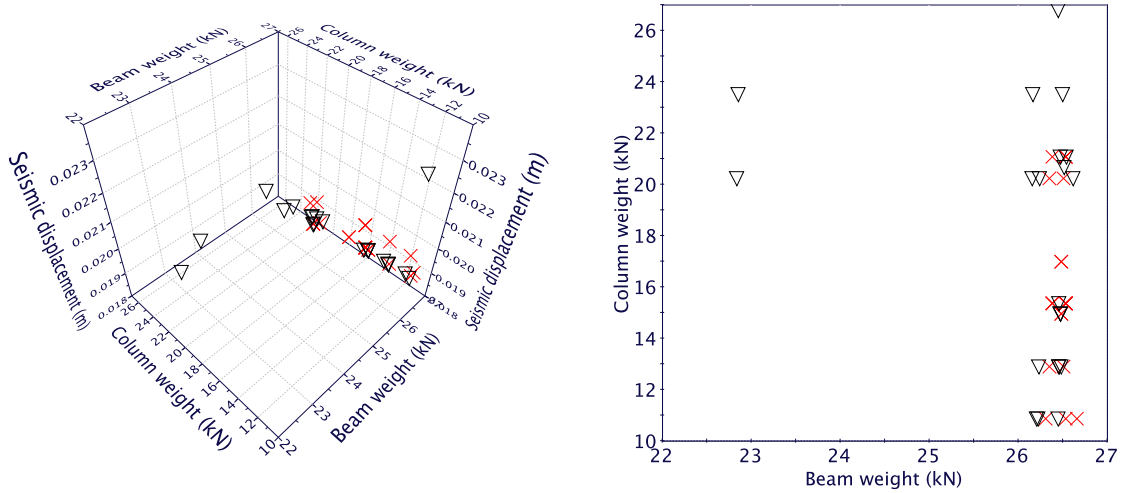


Figure 11: Pareto front designs found with NSGA-II (cross symbols) and with the reference-point algorithm described in Section 2.2 (empty downward triangles).

REFERENCES

[1] G.I. Schuëller and H.A. Jensen, Computational methods in optimization considering uncertainties - An overview. *Comput. Methods Appl. Mech. Engrg.*, **198**, 2–13, 2008.

[2] J. Figueira, S. Greco, M. Ehrgott (eds.), *Multiple criteria decision analysis: state of the art surveys*. Springer Verlag, Berlin, Heidelberg, New York, 2005.

[3] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE transactions on Evolutionary Computation*, **18**(4), 577–601, 2014.

[4] H. Jain, K. Deb, An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: handling constraints and extending to

an adaptive approach. *IEEE transactions on Evolutionary Computation*, **18**(4), 602–622, 2014.

[5] J. Branke, S. Greco, R. Slowinski, P. Zielniewicz, Learning value function in interactive evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, **19**(1), 88–102, 2014.

[6] M. Papadrakakis, N.D. Lagaros, V. Plevris, Multi-objective optimization of skeletal structures under static and seismic loading conditions. *Engineering Optimization*, **34**, 645–669, 2002.

[7] M. Papadrakakis, Y. Tsompanakis, N.D. Lagaros, M. Fragiadakis, Reliability based optimization of steel frames under seismic loading conditions using evolutionary computation. *Journal of Theoretical and Applied Mechanics*, **42**(3), 585–608, 2004.

[8] M.D. McKay, W.J. Conover, R.J. Beckman, A comparison of three methods for selecting values of input variables in the analysis of output from a computer Code. *Technometrics*, **21**(2), 239–245, 1979.

[9] J.L. Beck, E. Chan, A. Irfanoglu, C. Papadimitriou, Multi-criteria optimal structural design under uncertainty. *Earthquake Engineering and Structural Dynamics*, **28**, 741–761, 1999.

[10] D.M. Frangopol, Life-cycle performance, management, and optimisation of structural systems under uncertainty: accomplishments and challenges. *Structure and Infrastructure Engineering*, **7**(6), 389-413, 2011.

[11] N. Caterino, I. Iervolino, G. Manfredi, E. Cosenza, Comparative analysis of multi-criteria decision-making methods for seismic structural retrofitting. *Computer-Aided Civil and Infrastructure Engineering*, **24**, 432–445, 2009.

[12] H.A. Jensen, D.S. Kusanovic, M.A. Valdebenito, Compromise design of stochastic dynamical systems: A reliability-based approach. *Probabilistic Engineering Mechanics*, **29**, 40–52, 2012.

[13] M. Marchi, L. Rizzian, E. Rigoni, R. Russo, A. Clarich, Combining robustness and reliability with polynomial chaos techniques in multiobjective optimization problems: use of percentiles. A. Cunha, E. Caetano, P. Ribeiro, G. Müller eds. *Proceedings of the 9th International Conference on Structural Dynamics, EURODYN 2014*, Porto, Portugal, 30 June-2 July 2014.

[14] N. Wiener, The homogeneous chaos, *Amer. J. Math.* **60**, 897–936, 1938.

[15] A. Clarich, R. Russo, M. Marchi, E. Rigoni, Reliability-based design optimization applying polynomial chaos expansion: theory and applications. *10th World Congress on Structural and Multidisciplinary Optimization*, Orlando, Florida, USA, 19–24 May, 2013.

[16] M. Marchi, E. Rigoni, R. Russo, A. Clarich, Percentile via polynomial chaos expansion: bridging robust optimization and reliability. *International Conference EVOLVE 2013, A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation*, July 10–13, Leiden, NL, Extended Abstract Proceedings, ISBN 978-2-87971-118-8, ISSN 2222-9434, 2013.

[17] www.esteco.com

[18] R.T. Marler, J.S. Arora, Survey of multi-objective optimization methods for engineering. *Structural Multidisciplinary Optimization*, **26**, 369–395, 2004.

[19] D. Xiu, G.E. Karniadakis, The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.* **24**, 619–644, 2002.

[20] K. Deb, D. Padmanabhan, S. Gupta, A.K. Mall, Handling uncertainties through reliability-based optimization using evolutionary algorithms, *Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, *LNCS* **4403**, 66–68, 2007.

[21] K. Deb, S. Gupta, D. Daum, J. Branke, A.K. Mall, D. Padmanabhan, Reliability-based optimization using evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, **13**, 1054–1074, 2009.

[22] B. Sudret, A. Der Kiureghian, Comparison of finite Element Reliability Methods, *Probabilistic Engineering Mechanics*, **17**, 337–348, 2002.

[23] B. Sudret, M. Berveiller, M. Lemaire, A stochastic finite element procedure for moment and reliability analysis. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, **15**(7-8), 825–866, 2006.

[24] G. Blatman, B. Sudret, An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis. *Probabilistic Engineering Mechanics*, **25**, 183–197, 2010.

[25] R. Filomeno Coelho, P. Bouillard, Multi-objective reliability-based optimization with stochastic metamodels. *Evolutionary Computation*, **19**(4), 525–560, 2011.

[26] M. Marchi, E. Rigoni, R. Russo, A. Clarich, Guideline identification for optimization under uncertainty through the optimization of a boomerang trajectory. To appear in *Seventh International Conference on Evolutionary Multi-Criterion Optimization (EMO 2015).*

[27] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, **6**, 182–197, 2002.

[28] K. Deb, T. Goel, Controlled elitist non-dominated sorting genetic algorithms for better convergence. *First International Conference On Evolutionary Multi-Criterion Optimization (EMO)*, *LNCS*, **1993**, 67–81, 2001.

[29] M. Luque, F. Ruiz, K. Miettinen, Global formulation for interactive multiobjective optimization. *OR Spectrum*, **33**(1), 27–48, 2011.

[30] K. Deb, A. Pratap Mathur and T. Meyarivan, Constrained test problems for multi-objective evolutionary optimization, *First International Conference On Evolutionary Multi-Criterion Optimization (EMO)*, *LNCS*, **1993**, 284–298, 2001.

[31] E. Zitzler and K. Deb and L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results. *Evolutionay Computation*, **8**(2), 173–195, 2000.

[32] SAP2000 Ultimate 16.0.1, 2013, www.csiamerica.com