DESIGN METHODOLOGY FOR ENGINEERING STRUCTURES FEATURING AUTOMATIC DESIGN CONCEPT GENERATION BY CONSTRAINT PROGRAMMING AND OBJECTIVE RANKING BY MULTI-ATTRIBUTE EVALUATION

CARLOS LÓPEZ[1], PHILIP EYCKENS[1], JAN STROOBANTS[1], VICKY ILIOPOULOU[1],
STIJN JONCKHEERE[2,3], BERT PLUYMERS[2,3], WIM DESMET[2,3], KURT DE GRAVE[4] & JAN GOOS[1]
[1]CodesignS – Flanders Make, Belgium
[2]Core Lab DMMS, Belgium
[3]Department of Mechanical Engineering, KU Leuven, Belgium
[4]DecisionS – Flanders Make, Belgium

ABSTRACT

The industrial design process of engineering products faces the challenge of increasingly complex product requirements for a progressively decreasing time-to-market. Many methodologies and tools exist to optimize design from a reference design, but this is far less the case for the early stages of the design. This early stage is characterized by a large uncertainty on many levels: functional requirements, operational conditions, boundary conditions and production process selection and -design. This paper proposes a two-step methodology for the design of beam-reinforced flat panels, with applications in e.g. automotive-, machine- and aerospace industries. The first step consists of automatic generation of beam configurations with varying geometry. This generation is realized by constraint programming, taking a number of design constraints into account, such as non-overlapping beams, connection angles, and (indirect) connectivity of beams to fixed end joints. The second step evaluates the performance of the generated design concepts, taking into account an extensive set of attributes such as cost, mass, manufacturability, robustness, NVH, strength, stiffness and fatigue. The goal is to extract as much information as possible from the concepts in order to list an objective ranking. This ranking assists the designer in determining the best concept to be further elaborated in later design stages. The proposed methodology has been tested in a car door use case, generating and evaluating more than 1000 concepts.
Keywords: multi-attribute analysis, automatic design, constraint programming, lightweight design.

1 INTRODUCTION

Mechanical design is constantly evolving towards "Design for Excellence" (DfX), and nowadays conceiving a new product focusing mainly or solely on weight and cost reduction is not sufficient to satisfy the strict market demands. On top of being lightweight and cost-efficient, new designs should be optimized for a multitude of extra attributes such as robustness, manufacturability or recyclability, among others. These attributes should be taken into account from the very early stages of the product development process (concept and design phase), since they have the greatest impact in the final cost and time-to-market of the product [1].

On the other hand, manual and traditional design approaches, while useful, might not be effective for early design phases where the design requirements are less strict, and more freedom is given to the designer. The importance of virtual-assisted simulations in such early design iterations is broadly acknowledged. For instance, EARPA, the association of automotive R&D organizations, states that by front-loading design decisions into an early, virtual stage, efforts spent in expensive and time-consuming physical prototyping iterations are drastically reduced [2]. Consequently, automatic approaches can assist the designer in

a more exhaustive exploration of the design space, in order to find the novel and improved solutions that are demanded by industry.

This research aims to provide a design methodology able to automatically generate concepts consisting of straight beams placed in a 2D plane, and automatically provide an objective ranking of their performance, which is derived from the evaluation of multiple different attributes. The methodology can have potential applications in the early conceptual design of products in different industries: automotive (car door), aerospace (aircraft wings) or architecture (roof of a building). The goal is to assist the designer in selecting the most promising concepts that need to be considered and further elaborated in later stages of the design phase.

Several tools have been developed in the past to assist during the concept phase of the development process. However, most of them are developed for very specific sectors (e.g. MODESSA for consumer goods [3] or 2nd-CAD for electro-mechanical systems [4]), or by exploiting existing design knowledge to generate and evaluate conceptual design variants [5] rather than generating new alternatives from scratch. Others, such as the study of La Rocca [6], use a parametric model module to generate models that are suitable for Finite Element Analysis (FEA). In general, the main focus of these tools is on the generation of concept variants, while the evaluation and ranking is still not fully automatized.

The goal of our approach is to have a human centered framework where all the necessary steps to perform concept generation, multi-attribute evaluation and ranking are embedded within a single framework. The automatic generation of concepts is based on constraint programming algorithms, taking into account a number of design constraints specified by the designer. The multi-attribute evaluation involves the automatic creation of 1D-2D Finite Element (FE) models in order to calculate a number of structural attributes. The objective ranking is based on the evaluation results, giving to each concept a performance indicator that scores its overall performance.

2 GENERAL METHODOLOGY

This paper proposes a three step methodology to identify in an objective way the most promising concepts that need to be further elaborated in order to reach the best possible design of a new product. The scope of this study is limited to the design of the beam-reinforcement scheme of flat plates. The three steps that are discussed in this paper are:

1. Automatic generation.
2. Automatic multi-attribute evaluation.
3. Filtering and ranking.

These three steps are discussed in detail in the following sections (Sections 3–5). The design methodology is implemented in Python, and included with the framework of FreeCAD [7], a 3D CAD/CAE parametric modeling application, primarily conceived for mechanical design. FreeCAD is selected since it is a modular, fully open source software that allows to create advanced extensions and customization based on Python scripts. An extra reason to select FreeCAD as a supporting framework is that our methodology is conceived for early design phases, and the ambition is that the most promising architectures are further developed into complex 3D models that will need to be re-evaluated. Therefore, the CAD/CAE capabilities of FreeCAD can be fully exploited by the designer in later phases, and the multi-attribute evaluation of those complex 3D models can be assessed using the methodology exposed in [8], which is also fully implemented in the same FreeCAD framework [9].

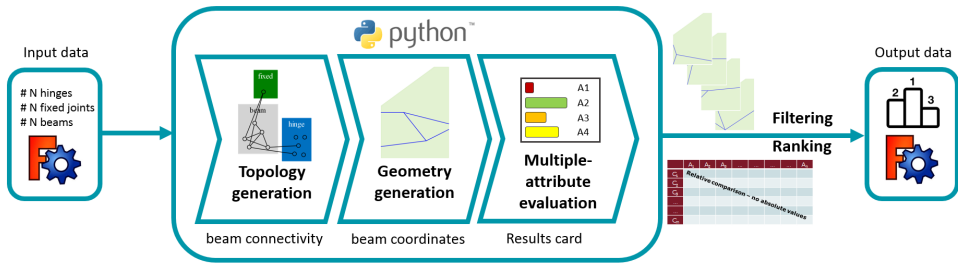Fig. 1 shows a high-level flowchart of the methodology.

Figure 1: Flowchart of the three-step methodology proposed in this paper.

3 AUTOMATIC CONCEPT GENERATION

Beam architectures are automatically generated within a given design space, using the two step approach depicted in Fig. 1: (i) topology generation and (ii) geometry generation. In the first step, topological graphs are generated, that define all the possible ways to connect a given number of joints. Some joints can move freely in the design space, while others are more limited (e.g. a hinge joint on the left side of the door). The second step generates geometries that are compliant with each individual topology graph. Since the focus is on the automatic generation of a car door internal reinforcement, three input parameters are considered:

1. Hinges: Number of joints in the design that are hinges. Hinges are the starting point of the door support frame.
2. Beams: Number of free beams of the reinforcement architecture. More beams increase the number of topological and geometrical variants exponentially, but obviously allow for more freedom in the design.
3. Fixed: Number of points where at least a beam needs to end. This allows the user to steer the design, by forcing at least on beam to end in or pass through a designated zone in the design. This way of reasoning is similar to the "preserve" concept in AutoCAD's generative design tool [10].

While the topology and geometry design can be combined into a single exploration algorithm, they are untangled and treated separately in this approach. Advantages are that the designer can (i) inspect intermediately generated topologies on the fly, and (ii) manually intervene by studying a single topology in more detail, by simply generating more alternative designs. In both topology and geometry generation, we are looking for many different variants. To this end, the design space exploration is not an optimization problem, but rather a satisfaction problem, which suits a constraint programming approach.
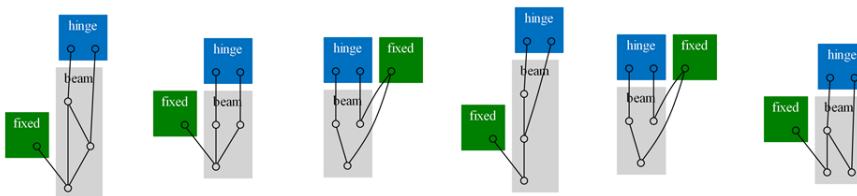

Figure 2: A subset of the possible topologies for 2 hinges, 5 beams and 1 fixed point.

Beams consist of two joints, and all fixed, free and hinge joints touch at least one beam. Topology generation aims to determine many different possible ways to connect the hinges to the beams and finally to the fixed points. Constraint programming is used to compute beam connectivity matrices under the constraint that all beams are indirectly connected to a hinge. For instance, in the case of fixing as input 2 hinges, 5 beams and 1 fixed point, some of the possible configurations are shown in Fig. 2. In these topology graphs, beams are represented by circles. Beams that share joints are connected by lines.

The geometry generation on the other hand, takes the beam connectivity as an additional input, and computes joint coordinates $[x, y]$ (and hence beam coordinates $[x_1, y_1, x_2, y_2]$) by means of a (nonlinear) constraint program. Constraints like

1. unconnected beams do not overlap
2. the minimum angle between beams is $30°$
3. the maximum number of beam connections in a joint (e.g. #beams ≤ 4)
4. minimum beam length (e.g. min $L_{\text{beam}} \geq 10$ cm)

can be used to infuse some design knowledge in the generation step. It is also possible to define *no-go* zones, which are areas where no joints are allowed, and no beams can cross (e.g. in a window). The latter is similar to the "obstacles" concept in AutoCAD's generative design tool [10]. The presented approach is different in the sense that we are reasoning over beam concepts, rather than the local thickness (or absence) of material, which is usually the approach in topology optimization [11]. Especially beam overlap is a challenging constraint to implement efficiently, but it helps designers to understand the constraint program, and it is an important rule to reduce the search space.

The design space is discretized by means of a finite grid, containing the possible joint locations. Even then, many feasible geometries are compliant with a single topology. To this end, the search strategy for design variants is randomized in the constraint program, in order to promote a diverse population [12].

The grid can be used to further direct the design space exploration, by means of promoting some joint locations. Far left Fig. 3 shows an example of a discrete grid of joint locations, where the designer is giving more priority to find beams in the outer part of the door frame. It also shows several geometries that are compliant with a single 5-beam topology.
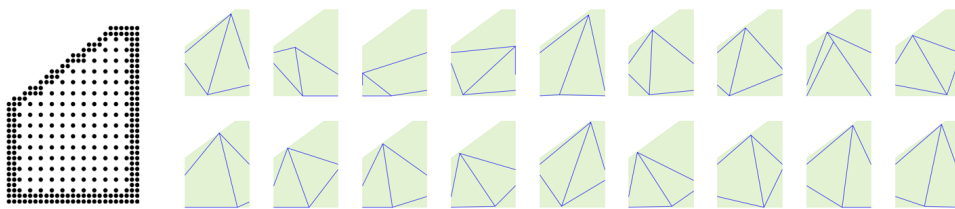

Figure 3: Example of geometries generated for a single topology using the automatic generation tool, given the custom joint grid (left).

The concept generation and evaluation are deliberately decoupled. The drawback is that we do not consider attributes like mass, stiffness, or manufacturability in the design space exploration. However, the individual topology and geometry generation steps are more straightforward, allowing for rapid variant synthesis. Additional design rules can then be implemented as constraints, or as simple filters that can be applied after any generation step.

4 AUTOMATIC CONCEPT EVALUATION

This section details the necessary steps to perform an automatic multi-attribute evaluation of the concepts generated in Section 3. In this study, eight different attributes are studied:

- Cost: includes raw material and production costs.
- Mass: is a key design driver for many applications, in the automotive sector and beyond.
- Manufacturability: gives a relative assessment of how suitable is the design when taken into production.
- Robustness: provides a metric for interconnectivity of the design's components (degree of modularity of the concept), which tells, e.g., how easily the parts in the concept can be changed or modified.
- First eigenfrequency: important to avoid a resonant behaviour of the structure.
- Fatigue: the highest stress that a material can withstand for a given number of cycles without breaking.
- Stiffness: measures the resistance of a structure towards deformation in response to an applied set of forces.
- Strength ratio: gives the highest occurring stress relative to the load-bearing capacity.

Further details regarding the calculation of these attributes can be found in [8]. Fig. 4 shows the menu where the user needs to input all the necessary information for performing the automatic evaluation. In the first block (top part), the user defines the structural concepts that need to be evaluated. The structure consists of two parts: (i) a 2D mesh of the design space, which is common to all the concepts and (ii) a 1D beam model that is built based on the geometry output generated in Section 3. Moreover, the user needs to define a few extra parameters required for the simulation, such as the thickness of the 2D plate (design space) and the unitary price of the joints.

In the second block, the material properties and cross section of the beams are defined. The workbench includes a list of 20 commonly used materials in the automotive sector, where the material properties are already predefined. The only parameter that the user needs to fill in is the price of the material (EUR/kg). Moreover, the workbench offers the option to select three different types of cross-section for the 1D beams: rectangular, circular and tube. The user needs to define the necessary cross-sectional parameters in order to calculate the area and inertia of the beams. It is relevant to highlight that, at the current stage of development of the workbench, a single material and cross-section is considered for all beams. Extension to multi-material and multi-section workflows is however straightforward.

The third block allows the user to define the load cases and the mesh size of the beams. The loads must be defined as "point loads", specifying the location ($[x, y]$ coordinates, since the design is assumed to be flat), direction ($[1, 2, 3]$ corresponding to the $[x, y, z]$ coordinate system) and magnitude. Following the same reasoning, extra boundary constraints together with their degrees-of-freedom (DOF) can be also specified. The load cases are defined as a combination of loads, boundary constraints and the hinges that were previously predefined in the automatic concept generation of Section 3. The fourth block contains information regarding the number of vibration modes that need to be found within a frequency range.

After defining the information required in Fig. 4, the multi-attribute evaluation starts. The evaluation of the non-structural attributes (cost, mass, manufacturability and robustness) is assessed through Python scripts, where the methods detailed in [8] are applied to dedicated models. On the other hand, the evaluation of the structural attributes (stiffness, fatigue, strength ratio and eigenfrequency) require FEA. This study uses the Python-based solver CalculiX [13]. The structural model of each of the alternative concepts is automatically generated, combining the 2D baseline mesh with the 1D beam architecture generated in

Figure 4: Layout menu for automatic evaluation of geometries.

Section 3. The connectivity of the 1D and 2D meshes is imposed with the appropriate connector elements.

The rest of the FE model is automatically generated with extra information that depends whether the analysis to be carried out is static or modal:

1. Static analysis: loads, boundary constraints, load case, material and cross-section properties. One FE model per load case is generated.
2. Modal analysis: number of vibration modes, within an interval defined by initial and final frequency values, as well as material and cross-section properties.

After the evaluation of all the concepts (geometry and FE model), the results are transferred to a spreadsheet document containing a decision matrix, where a number of concepts (rows) are assessed on a number of attributes (columns), in a similar way to Table 1.

Table 1: Example of decision matrix.

Cases	Cost	Mass	Man.	Robust.	1st eigen.	Fatigue	Stiffness	SR
Concept 1								
Concept 2								
...								
Concept n								

5 FILTERING AND RANKING OR CONCEPTS

The results of the multi-attribute analysis carried out in Section 4 can be visualized in the next module of the workbench. Moreover, the different concepts can be filtered and ranked based on the requirements defined for each of the attributes, that need to be specified by the user.

The decision matrix of Table 1 can be visualized, and then a "performance indicator" will be calculated, measuring the performance of each concept taking into account the different attributes. The strategy is to normalize the attributes of each concept with respect to the best attribute of the whole set of concepts, therefore if a concept ranks the best in a specific attribute the normalized value will be 1. The transformation shown in eqn (1) captures the physical trends of the attributes, and it can be implemented in different ways (linear for most attributes, logarithmic for fatigue cycles). The value s in the exponent of the linear transformation becomes 1 or -1 according to if the best metric value is the maximum (i.e. for the attribute robustness, the higher the value the better, therefore $s = 1$) or the minimum one (i.e. for the mass or cost attribute, the lower the better, therefore $s = -1$). Consequently, the better concepts will have a higher score in their performance indicator

$$
X_{norm} = \begin{cases} max\left(\frac{log(X_{attribute})}{log(X_{BEST\,attribute})}, 0 \right), & \text{if } attribute = fatigue \\ \left(\frac{X_{attribute}}{X_{BEST\,attribute}} \right)^s, & \text{otherwise.} \end{cases} \tag{1}
$$

This method retains the information of how much better/worse is one concept with respect to another, is easy to implement and needs only a physical insight of the attributes from the designer. It is a good method for early design phases, since the designer has no previous experience and all attributes are of the same importance.

6 APPLICATION CASE: INTERNAL REINFORCEMENT OF A CAR DOOR

This section focuses on the application of the methodologies exposed in Sections 3–5 to achieve a lightweight design for the internal reinforcement structure of a car door. The constraint programming algorithms described in Section 3 were used to generate different alternative architectures and geometries of the internal reinforcement scheme. The following constraints have been implemented within the constraint programming algorithms:

- Number of beams can be 3, 4, 5 or 6.
- Number of hinges is 2 or 3.
- Number of fixed points is 1 or 2.
- No-go zone for beams (the window area).
- Unconnected beams cannot overlap.
- Minimum angle between beams is 30 degrees.
- Minimum beam length is 10 cm.
- Maximum number of beams concurring in a joint is 4.

The baseline geometry of the door is presented in Fig. 5, as well as the potential points/areas where the hinges and the fixed points are located. The design space is discretized with a grid of points of 5 mm spacing, both in vertical and horizontal direction.
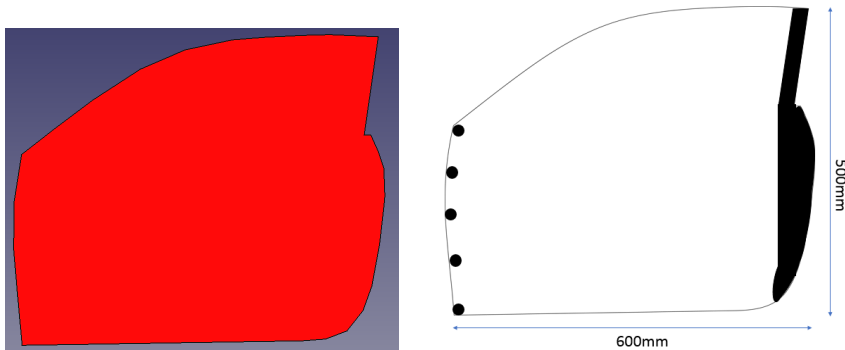

Figure 5: Base geometry for the car door and design space with possible hinges and fixed point locations.

With those parameters, 1885 different concepts have been generated and evaluated. A small sample of 32 concepts, extracted from the 1885, is presented in Fig. 6.
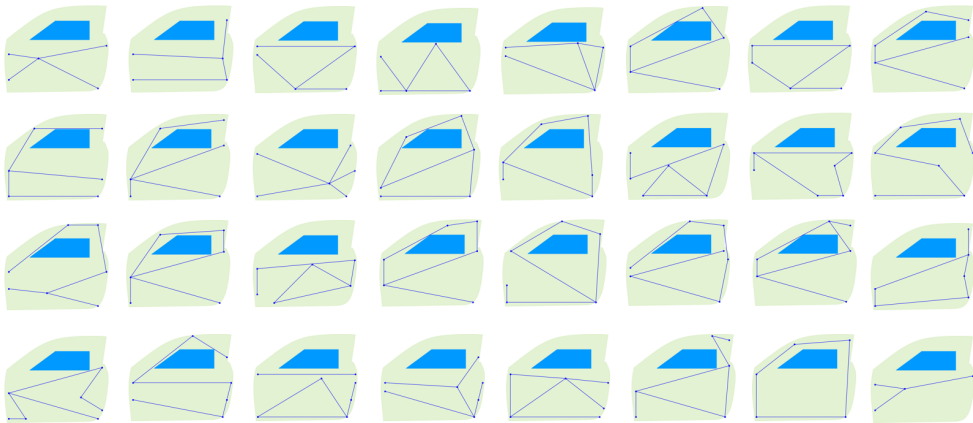
Figure 6: 32 generated architectures for the car door example.

The next step is to perform a multi-attribute evaluation of those 32 concepts, and calculate a performance indicator for each of them. In order to do so, some features need to be defined, such as:

- Material: aluminium, whose material properties are shown in Table 2.
- Beam cross-section: rectangular 10×10 mm.
- Thickness of the baseline plate: 1.2 mm.
- Load case 1 simulates a person of 100 kg hanging on the door (vertical load on the top right of the door). The door is open, so only the hinges act as boundary constraints, allowing exclusively rotation with respect to the vertical axes of the door.

- Load case 2 simulates an external constant pressure of 100 kg/m^2 over the door frame. The door is closed, therefore boundary constraints fixing displacements and rotations are defined over the entire contour of the door.

Table 2: Material properties.

	E [GPa]	ρ [kg/m^3]	Poisson ratio	Yield stress [MPa]	Price [EUR/kg]
Aluminium	210	7800	0.285	255	10
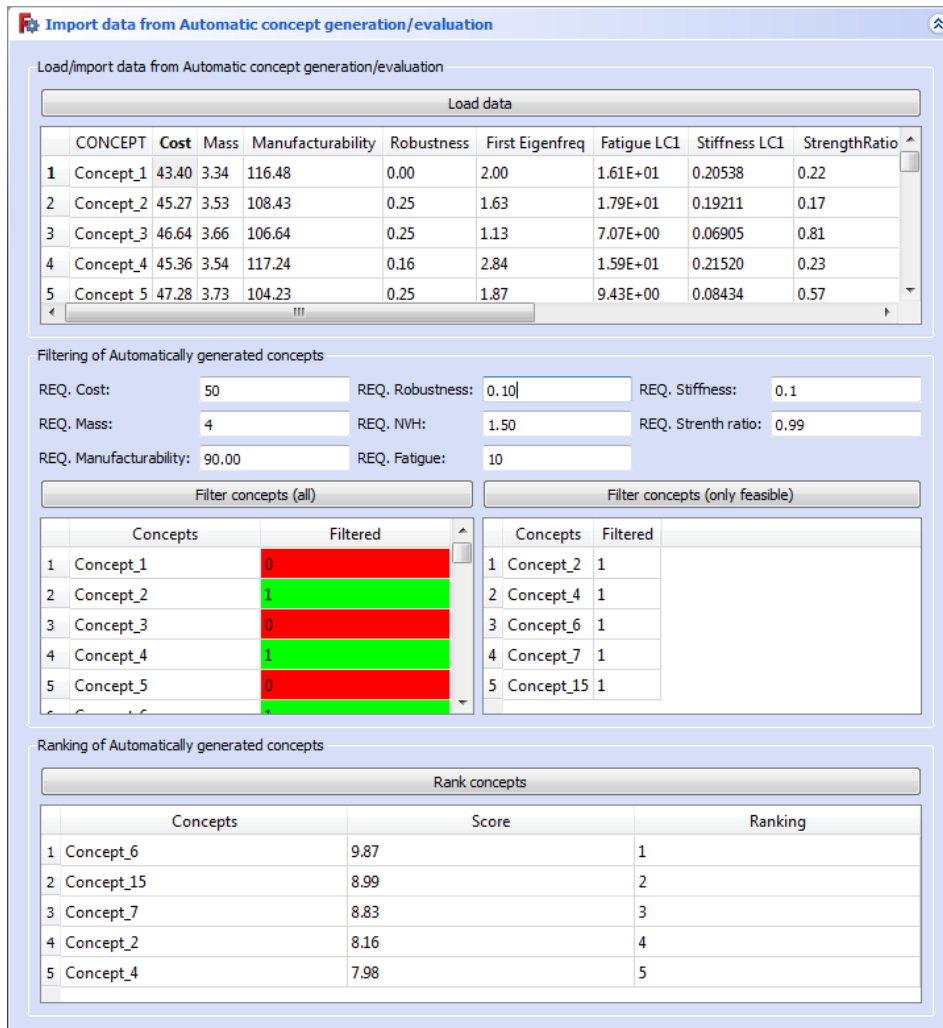
Figure 7: Example of the decision matrix visualization, requirements definition, filtering and ranking.

Finally, the results obtained from the multi-attribute evaluation are organized in a decision matrix, and a performance indicator is calculated for each concept. The concepts are then filtered and ranked based on a set of requirements provided by the user. This information is shown in Fig. 7.

As it can be seen in this particular example, if the requirements are too strict, it is likely that only a few concepts are considered feasible. In this case, only 5 out of 32 concepts are feasible, given that one or more of the attributes is not compliant with the desired requirements. From the feasible results, the ranking algorithms suggest that the most promising concept to be further evaluated is Concept_6 (with a performance indicator score of 9.87), whereas the worst of the feasible concepts is Concept_4 (with a score of 7.98). Both concepts are shown in Fig. 8.
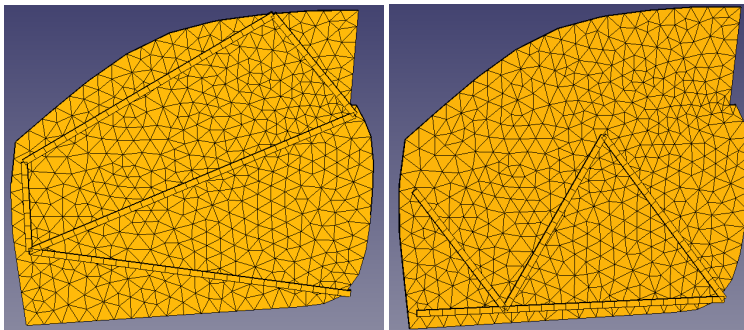
Figure 8: Concept_6 (left) and Concept_4 (right).

Based on the configuration of both concepts, the designer can extract some conclusions that may help in later design phases. For instance, one of the reasons for a better performance of the Concept_6 is its superior performance regarding LC-1, since there are reinforcement beams in the vicinity of the load (located in the top right corner). Moreover, the robustness of Concept_6 is superior as well. In Concept_4, if the bottom bar needs to be replaced, it will affect all the other beams. This situation does not happen in any of the beams of Concept_6. A deeper look on the results can also assist the designer in later design phases, i.e. for defining the importance of each attribute through weighting factors, or to use some of the gained knowledge in an optimization problem (i.e. knowing which attributes are more sensitive to variations, or to define constraints that are not too strict).

The methodology gives also insight over the requirements that the designer needs to define. For instance, if more feasible concepts are desired, the designer has the information that if the mass or cost limit is less strict, more concepts will become feasible because they are compliant with the rest of the attributes. An example is given in Fig. 9, which shows the geometry of a feasible concept (Concept_4) together with the geometries of two non-feasible alternatives (Concept_1 and Concept_3). Moreover, Fig. 10 shows the table with the performance metrics of the three concepts.

From the results it can be extracted that, even though most of attributes of Concept_1 are similar to Concept_4, the former is discarded exclusively due to a robustness value of 0. This means that any modification (i.e. replacement) carried out in one of the beams will affect the rest of the structure, which is something that can be seen in Fig. 10 since all beams converge to a joint. If such an issue is not very relevant for the designer, the robustness requirement could be modified in order to convert Concept_1 into feasible. On the other hand, Concept_3 does not fulfill the minimum requirements in stiffness and fatigue for LC1, nor for the first
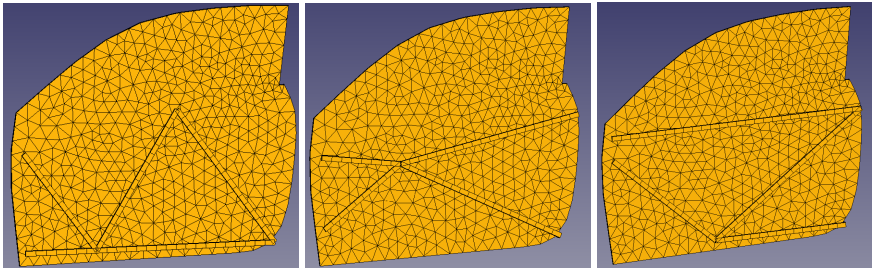
Figure 9: Concept_4 (left), Concept_1 (centre) and Concept_3 (right).

Figure 10: Comparison table of Concept_1, Concept_3 and Concept_4.

eigenfrequency. It is expected that if the two beams could connect to the same hinge and built a close triangle, the dynamic behaviour will improve. Consequently, the tool gives also insight regarding possible design modifications for a given concept.

7 CONCLUSION

This research proposes a design methodology for the automatic generation of 1D architectures, followed by an objective ranking of them based on an automatic multi-attribute analysis. Eight different attributes, both structural (stiffness, first eigenfrequency, strength ratio and fatigue) and non-structural (cost, mass, manufacturability and robustness), are taken into account for evaluation. The methods and algorithms developed for the generation of concepts, the transformation of the concepts to FE models, the attribute evaluation and the filtering and ranking are integrated within a FreeCAD framework, in order to favour the user-friendliness of the methodology and benefit from the CAD capabilities of FreeCAD.

The methodology has been applied to the design of the interior beam reinforcement scheme of a car door structure, where hundreds of alternatives have been automatically generated and ranked. The methodology helped to identify the architectures that are compliant with the requirements of the designer, as well as rank them based on a score that reflect their global better performance with respect to the attributes evaluated. Moreover, the results helped the designer to gain some insight over the definition of the requirements, and to derive the reasons why one concept has a superior performance over another.

REFERENCES

[1] Thomke, S., *Enlighted Experimentation: The New Imperative for Innovation*, Harvard Business Review, pp. 67–75, February 2001.

[2] EARPA, Position paper, *The Key Role of Modeling and Simulation for European Road Transport*, https://www.earpa.eu/docs/2010/EARPA%20M&S%20Position%20 Paper_2010.pdf, 2014.

[3] Kersten, T., "Modessa" a computer based conceptual design support system. *AI System Support for Conceptual Design*, pp. 241–259, 1995.

[4] Vargas-Hernandez, N. & Shah, J.J., 2nd-CAD: A tool for conceptual systems design in electromechanical domain. *Journal of Computing and Information Science in Engineering*, **4**, pp. 28–36, 2004.

[5] Bryant, C.R., McAdams, D.A., Stone, R.B., Kurtoglu T. & Campbell, M.I., *A Computational Technique for Concept Generation* IDETC/CIE 2005, Long Beach, California USA, 2005.

[6] La Rocca, G. & van Tooren, M., A knowledge based engineering approach to support automatic generation of FE models in aircraft design, *45th AIAA Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings*, 2007.

[7] The FreeCAD Team. *FreeCAD documentation*. https://www.freecadweb.org/wiki/ Getting_started.

[8] López, C., Boucher, M., Van Doninck, B., Maurin, F., Iliopoulou, V., Stroobants, J., Jonckheere, S., Deckers, E., Van Poppel, S. & Pluymers, B., Multi-attribute evaluation of concept variants for lightweight structures. *Proceedings of ISMA 2018 – International Conference on Noise and Vibration Engineering and USD 2018 – International Conference on Uncertainty in Structural Dynamics*, pp. 2249–2263.

[9] Iliopoulou, V., López, C., De Grave, K., Dzyuba, V., Van Doninck, B., Stroobants, J., Eyckens, P. & Van Poppel, S., Conceptual design for the manufacturing industry. *Journal of Engineering Design*, Under Review.

[10] https://www.autodesk.com/solutions/generative-design.

[11] Bendsoe, M. & Kikuchi, N., Generating optimal topologies in optimal design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, **71**, pp. 197–224, 1988.

[12] Feydy, T., Goldwaser, A., Schutt, A., Stuckey, P.J. & Young, K.D., Priority search with MiniZinc. *ModRef 2017: The Sixteenth International Workshop on Constraint Modelling and Reformulation at CP2017*, 2017.

[13] Dhondt, G. & Wittig, K., CalculiX documentation. http://www.calculix.de/.