

## Research Article

# An Automated Structural Optimisation Methodology for Scissor Structures Using a Genetic Algorithm

Aushim Koumar,<sup>1</sup> Tine Tysmans,<sup>2</sup> Rajan Filomeno Coelho,<sup>3</sup> and Niels De Temmerman<sup>4</sup>

<sup>1</sup>Department of Mechanics of Materials and Constructions (MeMC) and Department of Architectural Engineering ( $\alpha$ -lab), Vrije Universiteit Brussel (VUB), Brussels, Belgium

<sup>2</sup>Department of Mechanics of Materials and Constructions (MeMC), VUB, Brussels, Belgium

<sup>3</sup>BATir Department, Université Libre de Bruxelles (ULB), Bruxelles, Belgium

<sup>4</sup>Department of Architectural Engineering ( $\alpha$ -lab), VUB, Brussels, Belgium

Correspondence should be addressed to Aushim Koumar; [akoumar@vub.ac.be](mailto:akoumar@vub.ac.be)

Received 2 September 2016; Revised 20 December 2016; Accepted 26 December 2016; Published 18 January 2017

Academic Editor: Vahid Hajipour

Copyright © 2017 Aushim Koumar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We developed a fully automated multiobjective optimisation framework using genetic algorithms to generate a range of optimal barrel vault scissor structures. Compared to other optimisation methods, genetic algorithms are more robust and efficient when dealing with multiobjective optimisation problems and provide a better view of the search space while reducing the chance to be stuck in a local minimum. The novelty of this work is the application and validation (using metrics) of genetic algorithms for the shape and size optimisation of scissor structures, which has not been done so far for two objectives. We tested the feasibility and capacity of the methodology by optimising a 6 m span barrel vault to weight and compactness and by obtaining optimal solutions in an efficient way using NSGA-II. This paper presents the framework and the results of the case study. The in-depth analysis of the influence of the optimisation variables on the results yields new insights which can help in making choices with regard to the design variables, the constraints, and the number of individuals and generations in order to obtain efficiently a trade-off of optimal solutions.

## 1. Introduction

*Context.* During the last decades, there has been an important increase in disasters caused by nature or people. When a disaster occurs, the response of humanitarian organisations must happen quickly and transitional settlement is one of the basic needs that must be provided as quickly as possible to the affected population.

During the first emergency phase, there is a need not only for emergency shelters for the local population, but also for larger constructions, such as warehouses and dispensaries. The latter are generally called collective service tents (CSTs) [1]. The current CSTs are, however, structurally complex and it takes a substantial amount of time to build them. An alternative to the current kit-of-part systems, which is faster to erect, is the use of deployable scissor systems [1].

Scissor units, pantographs [2], or scissor-like elements (SLEs) [3] are structural elements composed of two straight beams, connected through an intermediate point (a pivotal connection) that enables a single axis rotation in the plane of the structural element [4]. By hinging scissor units to each other at their end points, a two-dimensional transformable linkage is formed. By modifying the shape of the beams and the location of the pivotal connection, different units can be obtained, the most important units for this research being the translational units and polar units [4].

The advantage of using scissor structures for emergency tents is the easy transportability, the ease and speed of erection and folding (skilled labour is thus not needed for the installation), and the high volume increase between compact and deployed state [5, 6]. Furthermore, scissor structures can be disassembled and their structural elements can be

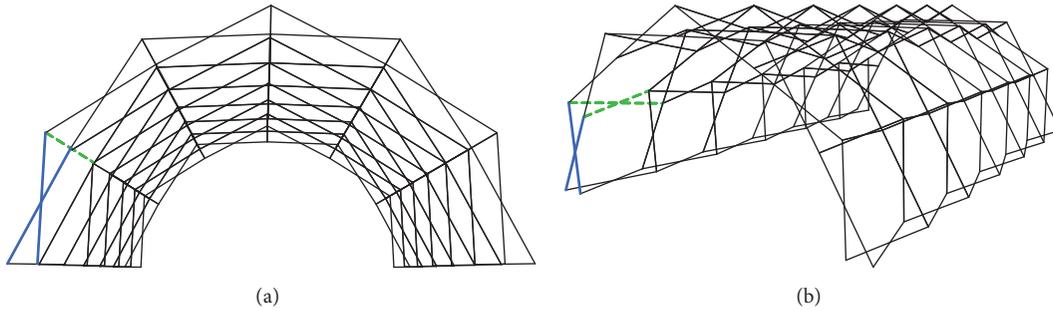


FIGURE 1: A barrel vault structure is composed of arches with polar units (blue bold solid line) connected through translational units (green bold dashed line).

combined into a building kit for use in a later stage of disaster relief [7], which emphasises the sustainable aspect of the solution.

*Problem Statement.* Conventional design of scissor structures starts with a geometrical design followed by limited structural design, most of the time combined with common sense and know-how. The novelty of this research presented in this paper is to couple both geometrical and structural design phases to obtain optimised scissor structures. Optimisation is particularly beneficial for disaster relief shelters, as, unlike common civil constructions, they are produced in very large quantities. In order to reduce cost and to improve transportability, keeping the weight to a minimum is crucial. Additionally, minimising the space needed for transport is important: by making the scissor structures as compact as possible, more of them could be sent in the same container (transport by ship). A design for the two conflicting objectives mass and compactness is therefore essential for emergency scissor structures. Multiobjective optimisation methods allow generating a range of optimal solutions for multiple objectives and distinguish themselves from traditional design methodologies which provide one single solution whose dimensions are optimised a posteriori (after the geometrical design).

The structural optimisation of scissor structures is however not trivial, because of the different types of design variables. The type of units and the way they are connected have an influence on the shape (shape optimisation). A second type of variable is the number of units which does not influence the shape but the typology of the structure (typology optimisation). The variation in the number of units affects the whole structure because it must meet the geometric and kinematic constraints of scissor structures. Lastly, size optimisation is required for the third type of design variables, which are the cross-sectional dimensions.

In this research, a barrel vault scissor structure shape is chosen because it can be composed of polar and translational units and has therefore a straightforward design and high compactness in its undeployed state (Figure 1). On this structure, both a typology (number of scissor units) and a size (cross section) optimisation are performed. Furthermore, because the aim is to have a compact and lightweight

structure, a two-objective optimisation must be performed yielding not one solution but a range of solutions.

In this paper, a general methodology using GAs is developed to obtain a set of structural optimal solutions for barrel vault shaped scissor structures in a fully automated framework without interaction with the user. The methodology behind this framework is illustrated by applying it to a case study: the optimisation of a barrel vault structure for disaster relief. The originality of the research lies in the in-depth analysis of the results, allowing us to understand the way GAs work on scissor structures and to provide new insights into the parameters which make the optimisation of barrel vault scissor structures efficient. The importance of the number of individuals and generations in the population is shown and the evolution of the constraints and design variables is discussed. The application and validation (using metrics) of genetic algorithms for the shape and size optimisation of scissor structures for two objectives are the novel contribution of this work.

*Literature Review.* Very little research is conducted on the structural optimisation of scissor structures. Kaveh et al. were one of the first to perform analyses and size optimisation using an Ant Colony Optimisation (ACO) algorithm [8, 9]. Gantes et al. introduced briefly genetic algorithms in the optimisation of deployable structures [10]. The material properties and cross-sectional dimensions of a bistable deployable structure were adapted to optimise the cost of the beam members, while satisfying the deflection and stress constraints. Kaveh and Shojaee also used the genetic algorithm [11] or the Recursive Quadratic Programming Method [12] to show the potential of optimising foldable structures. Thrall used Simulated Annealing (SA) for finding efficient forms and section profiles for deployable structures [13] and in a recent publication she used stochastic search and gradient-based algorithms for identifying minimum weight solutions [14]. Lastly, Alegria Mira also applied Simulated Annealing (SA) for the optimisation of a reconfigurable component for deployable scissor structures [15]. SA is however not an algorithm that allows working with multiple objectives. Furthermore, it is very time-consuming and it does not build up an overall view of the search space because it works only with one candidate solution. Finding optimal multiobjective

TABLE 1: Comparison of current research with previous researches on optimisation of scissors structures.

Authors	Algorithm	Objective(s)	Variables	Constraints
Kaveh et al. [12]	Recursive Quadratic Programming Method	Weight	Cross-sectional (CS) area	Stress & buckling
Kaveh and Shojaee [11]	Genetic algorithm	Weight	CS area	Stress & buckling
Thrall [13]	Stochastic search	Weight	Shape and CS area	Stress & buckling
Mira et al. [15]	Simulated Annealing	Weight	Shape and CS area	Stress & buckling
This paper	Genetic algorithm	Weight & compactness	Shape and CS area	Stress, buckling & deformation

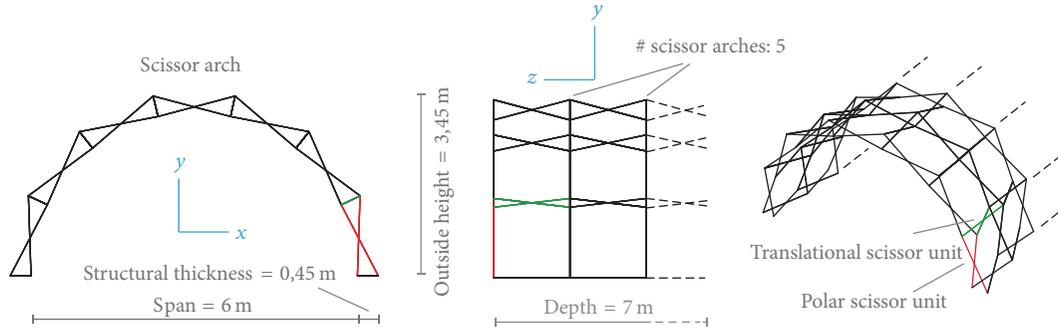


FIGURE 2: The arches of the barrel vault are made of polar units and, in-between, translational units are placed to hold the arches together. The number of arches depends on the number of unit (in this figure, there are 7 units).

solutions with ACO is difficult because numerous parameters must be fine-tuned [16, 17].

In [18], Alhanjouri and Belal demonstrated that genetic algorithms (GAs) perform better than ACO. GAs also have several other advantages [19]: first of all, they allow single or multiobjective optimisation; the latter is needed for this research as the aim is to minimise both the mass and the compactness. Secondly, instead of a single design point, GAs work with a population of design points in the space of design variables, producing a broad search of the objective space and thus lowering the chance of being stuck in a local minimum. At last, GAs can handle continuous, discrete, or mixed optimisation problems. In the case presented here, the number of units is, for example, a discrete variable and the continuous variables are the dimensions of the cross sections.

Table 1 provides a comparison of the current work with previous research on the optimisation of scissor structures.

## 2. Detailed Description of the Considered Case Study: The Barrel Vault Scissor Structure

This paragraph extensively investigates a case study of a barrel vault shaped scissor structure in order to study the algorithm parameters and to reflect on the results.

**2.1. Geometry and Loads.** The considered barrel vault is composed of polar scissor units for the arches and translational units between the arches with equal member's length and has an outside height of 3,45 m (Figure 2). The inside span of the structure is 6 m. Taking into account the structural thickness of 45 cm, the outside span is 6,9 m. As the depth

depends on the length of the scissor elements which in turn are dependent on the number of units, the depth is not fixed, but the total ground surface should be around 45 m<sup>2</sup> (determined by the head of the Médecin Sans Frontière Operation Centre Brussels Logistic Department [20]). This means that the depth will be approximately equal to 7 m. The barrel vault structure is hinged on each support with the ground.

For the scissor elements, unhardened 5059 aluminium ( $E = 70$  GPa,  $f_y = 160$  MPa, and  $\rho = 2780$  kg/m<sup>3</sup>) is used.

In Figure 3, the load cases are shown on the structure. The transversal wind is assumed to blow from the left to the right which causes pressure on one side (Zone A) and suction on the other sides (Zones B and C) and the longitudinal wind is causing pressure and suction in five different zones (D, E, G, H, and I); both wind loads are calculated using *Eurocode 1-4: Wind Actions* [21]. The two snow load cases (uniform and drifted snow case) are calculated using *Eurocode 1-3: Snow Loads* [21]. For the joint weight, a constant value of 1,5 kg is used. Lastly, the weight of a PVC-coated polyester membrane is assumed (700 g/m<sup>2</sup>); however, structural interference between the membrane and the arches is not considered in this optimisation exercise. With the appropriate load combination factors [22], a first preliminary structural analysis shows that two load combinations are detrimental for the analysis of the barrel vault. The load combination with the transversal wind load and drifted snow load (LC1) is determinative for the structural optimisation of the polar units in the arch and the load combination with the longitudinal wind load together with the drifted snow (LC2) is determinative for the structural optimisation of the translational units.

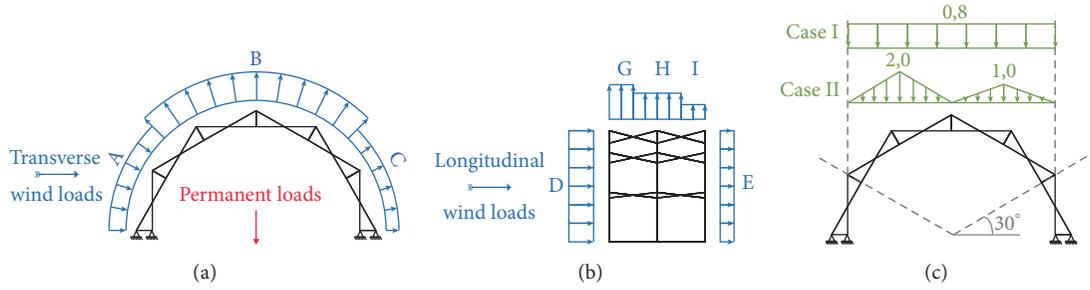


FIGURE 3: The transverse wind load causes pressure in one zone and suction in the two other zones (a). The longitudinal wind load causes a pressure in Zone D and suction in Zones G, H, I, and E (b). Two snow loads are considered, a uniform and a drifted one (c).

**2.2. Objectives, Constraints, and Design Variables.** The two objectives of the optimisation are the mass of the structure (calculated as a function of the volume of all structural elements and the permanent loads such as joints and membrane) and the compactness of the structure (calculated as the volume of the undeployed barrel vault).

The following constraints are considered, each time for both the polar and the translational units and for two load combinations:

- (i) The maximum stress is 160 MPa (unhardened 5059 aluminium).
- (ii) The maximum deflection is  $L/100$ . This constraint is less strict than for traditional structures and was set in agreement with our partner *Médecins Sans Frontières*.
- (iii) The maximum horizontal displacement is  $H/100$ . This constraint is less strict than for traditional structures and was set in agreement with our partner *Médecins Sans Frontières*.
- (iv) For the local buckling, an analytical calculation is implemented to check the member's local buckling by taking into account the normal force and the bending moments around the two axes following Eurocode 9 [23].
- (v) The global buckling calculation is based on the eigenvalue buckling analysis (linear perturbation procedure) of the model using Abaqus/CAE 6.12 [24].

Seven design variables are defined in this research: the number units of the barrel vault structure (typology optimisation) and the height, width, and thickness of the rectangular tube cross section for both the polar and the translational units (size optimisation). The boundary values for the design variables are shown in Table 2.

**2.3. Mathematical Formulation of the Problem.** One has

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_i(\mathbf{x}) \quad i = 1, 2, \\ \text{Subjected to:} \quad & g_j(\mathbf{x}) \leq 0 \quad j = 1, \dots, 16, \\ & \mathbf{x}_p \in \mathbf{X}_p \quad p = 1, \dots, 7, \end{aligned} \quad (1)$$

where  $\mathbf{x}$  is the vector of design variables (height, width, and thickness of both polar and translational units and the

TABLE 2: Boundary values for the seven design variables (height, width, and thickness of both the polar and the translational units).

	Lower limit	Upper limit
Number of scissor units	5	10
Height	40 mm	150 mm
Width	40 mm	150 mm
Thickness	2 mm	10 mm

number of units),  $\mathbf{X}_p$  is the set of  $\mathbf{x}_p$ ,  $f(\mathbf{x})$  is the objective functions that should be minimised (compactness and mass), and  $g_j(\mathbf{x})$  is the inequality constraints (stress, buckling, horizontal and vertical displacement, and global buckling for both the polar and the translational units and for two load combinations).

### 3. Optimisation Methodology

**3.1. The Genetic Optimisation Algorithm.** A genetic algorithm uses operators inspired from biological evolution to generate individuals, which represent in this case scissor structures with certain design parameters (e.g., 7 units with a hollow cross section of  $45 \times 45 \times 7$  mm). The quality of the individuals is then evaluated through a fitness function (e.g., a representation of the values of the objective functions). The set of solutions is then improved by applying biological operators [19]. GAs have three important operators to generate solutions with an increasing fitness level, that is, solutions that are more optimal: *selection*, *recombination*, and *mutation* [25]. The *selection* operator selects the individuals for the next generation. The *recombination* (e.g., single-point crossover) is a procedure where the good genes of each parent are combined together. At last, during the *mutation* phase, the characteristics of the individuals are changed (e.g., the values of the genes are changed from “1” to “0” or vice versa).

For this research, the Nondominated Sorting Genetic Algorithm II (NSGA-II) is used [26]. NSGA-II is a GA using the three operators previously mentioned but with some extra adjustments in the way the selection operator works which make it more efficient [27]. Before the selection part, the population is ranked into nondominated fronts. A solution is part of the nondominated front, also called a Pareto front,

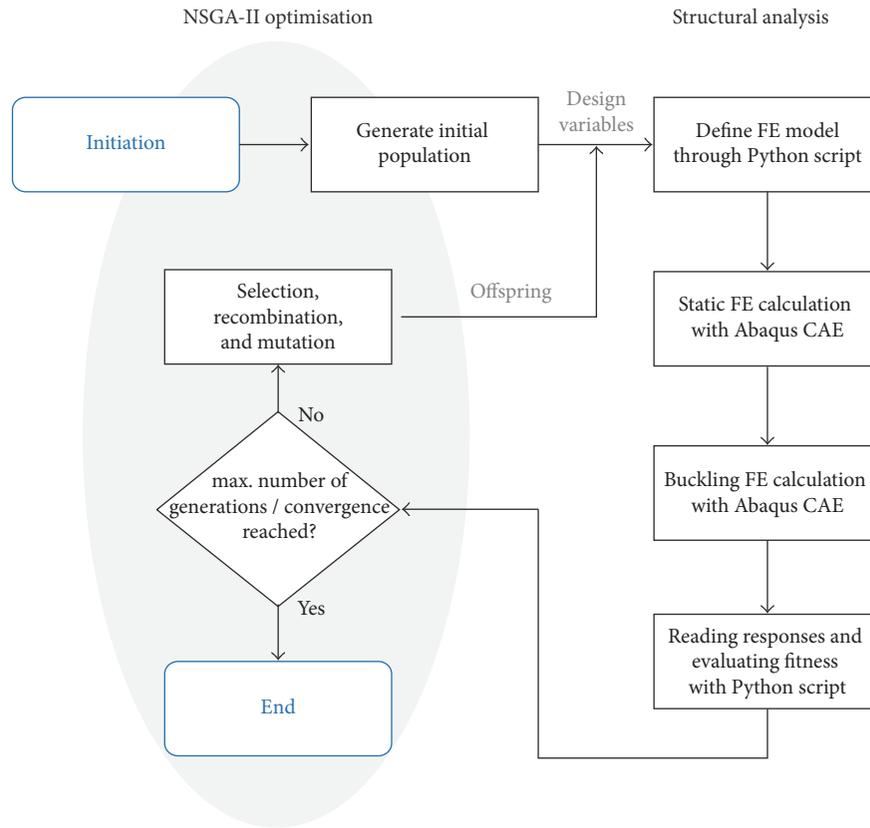


FIGURE 4: NSGA-II generates a first population that is sent to Abaqus for static and dynamic calculations. The output is sent back to Matlab to calculate the fitness function.

if no other feasible solution exists which would decrease some criterion without causing a simultaneous increase in at least one other criterion (for a minimisation problem) [25]. Once the nondominated front is found, it obtains a high fitness value and is temporarily ignored in order to sort the rest of the population into a second nondominated front. This process continues until the entire population is classified into several fronts. The new population is then formed by adding solutions from the first front till the size exceeds the population size  $N$ . For the last accepted front, the first  $N$  points are chosen by opting for the points that are located in a region with a smaller number of points, in order to maintain variety in the population (using the crowding distance principle) [19].

Instead of opting for the Pareto optimal method, several other approaches are possible to deal with a biobjective optimisation problem. Considering a linear combination of the objectives as done by Shahvari et al. in [28–30] could also be an option. However, within the larger scope of this research, it is essential that several optimal solutions are proposed by means of, for example, a Pareto front so that the designer can choose a posteriori the solution that fits best his needs. Indeed, in function of the preferences of the end user, a solution with less compactness or higher weight will be chosen. Furthermore, population-based search produces a broad search of the objective space and thus lowers the chance of being stuck in a local minimum. The current NSGA-II

algorithm used in this paper has been applied in different other case studies [31–33].

**3.2. The Structural Analysis Framework.** In order to couple the structural analysis of the barrel vault scissor structure with multicriteria optimisation in an automated way, a framework is set up combining Matlab R2014b, Abaqus 6.12, and the programming language Python. This process is illustrated in Figure 4: in a first step, the NSGA-II algorithm written in *Matlab* generates a first population of individuals with their own design variables based on Table 2. For each individual, a *Python* script is then launched which defines the barrel vault scissor structure geometry in Abaqus CAE without using the graphical user interface (GUI). A static and buckling calculation is then activated using the script. The output of the structural analysis is read in another script and the constraints are calculated. Those constraints are sent back to NSGA-II which evaluates the different objective functions and creates a new population as explained in Section 3.1. This process is repeated until the last individual of the last population is reached or until convergence is obtained (as explained in Section 4.5).

The strength of this open framework is that it can be used for any parametrised structure. It is flexible as it can optimise one or multiple objectives by varying multiple continuous and discrete design variables and by taking into account many constraints.

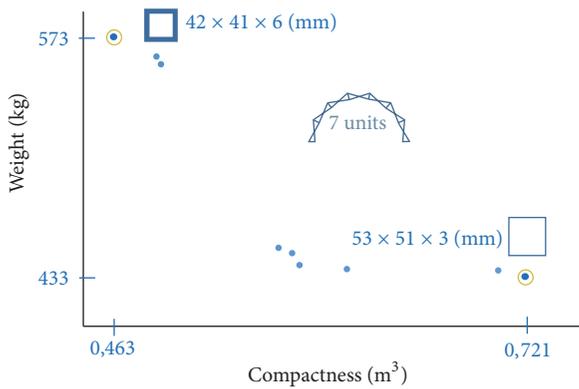


FIGURE 5: The Pareto front obtained for 60 individuals and 20 generations shows the conflicting behaviour of weight and compactness. A small cross section procures a compact solution, but, to meet constraints, the cross section must be thicker, which causes heavier solutions (and vice versa).

#### 4. Analysis and Discussion of Optimisation Results and Parameters

The structural optimisation consists of many design variables and constraints; fourteen constraints are considered (for both load combinations LC1 and LC2: horizontal displacement, deflection, stress of the polar units, stress of the translational units, local buckling test of the polar units, local buckling test of translational units, and global buckling) and seven design variables are optimised (number of scissor units and the three dimensions of a box cross section for both the polar and the translational units). Analysing the behaviour of each individual constraint and design variable is cumbersome and slow and does not significantly contribute to the aim of this study. Therefore, in a first instance, a simplified structural optimisation will be run in order to analyse the global behaviour of the objective space, the constraints, and the design variables. In this simplified simulation, the cross section of the polar and translational units will be identical and only one load combination will be analysed, LC1 (dominant transversal wind load and a drifted snow distribution). For this first optimisation, 60 individuals and 20 generations are used. More explanation about the number of individuals and generations and a convergence check will be provided in Sections 4.5 and 4.6.

**4.1. The Pareto Front and Pareto Set.** This subsection handles the general behaviour of the Pareto front. After running the optimisation routine, instead of finding one single optimal solution, a front is obtained with several optimal solutions. This is the consequence of working with multiple objectives. The set of Pareto solutions in the design space is called the *Pareto set* and the *Pareto front* is the image of the Pareto set in the objective space. The Pareto front, or the nondominated front as explained in the Introduction, of the optimisation of the barrel vault scissor structure is shown in Figure 5.

Because of the two conflicting objectives, mass and compactness, a Pareto front is obtained (the range of the design variables of the Pareto set is given in Table 3). This would

TABLE 3: Range of design variables of the Pareto set.

	Range
Number of scissor units	7
Height cross section	41 mm–51 mm
Width cross section	42 mm–69 mm
Thickness cross section	3 mm–6 mm

not have been the case if the objectives were not conflicting. In that case, only one optimum would have been found. The conflicting objectives can be explained as follows: on the one hand, a small cross section procures a compact solution, but, to meet the strength, stiffness, and stability criteria, the cross section must be thicker, which causes heavier solutions (the heaviest being the solution with a cross section of  $42 \times 41 \times 6$  mm, compactness of  $0.463 \text{ m}^3$ , and a weight of 573 kg). On the other hand, a bigger cross section has the opportunity to meet the criteria with a thinner cross section, but this affects the compactness of the whole structure (the lightest being the solution with a cross section of  $53 \times 51 \times 3$  mm, compactness of  $0.721 \text{ m}^3$ , and a weight of 433 kg).

Note that, in this simulation, the cross sections have a continuous variation in height, width, and thickness but, in reality, the cross sections will be chosen from a catalogue. This catalogue is not implemented in this simulation in order to have a better understanding of the relation between mass, cross section, and compactness. The interface is however open and the switch from continuous to discrete variables is therefore feasible.

**4.2. The Objective Space.** Figure 6 shows the projection of all calculated structures into the objective space. Three different kinds of markers can be distinguished in the figure: the green dots are scissor structures which fulfil all the constraints. The red crosses do not fulfil one or more of the four constraints and the blue dots are Pareto optimal solutions, that is, the optimal solutions of those that fulfil all the constraints. One can note that there is no clean separation between the red and the green dots. Some red points are above the Pareto front and are still failing, while neighbouring dots do fulfil the constraints. This behaviour can be explained by recalling that Figure 6 shows the objective space and not the design space. This means that two adjacent individuals could have totally different design variables but almost the same objective values. These outlier solutions were generated in one of the first generations and represent structures with a very small number of units which causes their elements to fail by local buckling. The behaviour of the design variables is explained in Section 4.4.

**4.3. The Constraints.** The aim of this subsection is to determine which one(s) of the constraints are the most dominant in the structural optimisation of the barrel vault scissor structure. The evolution of the four constraints (stress, deflection, local buckling, and global buckling) was plotted using the *boxplot* function of Matlab (Figure 7) [34]. Important to note is that the Y-value in the figure is the normalised difference

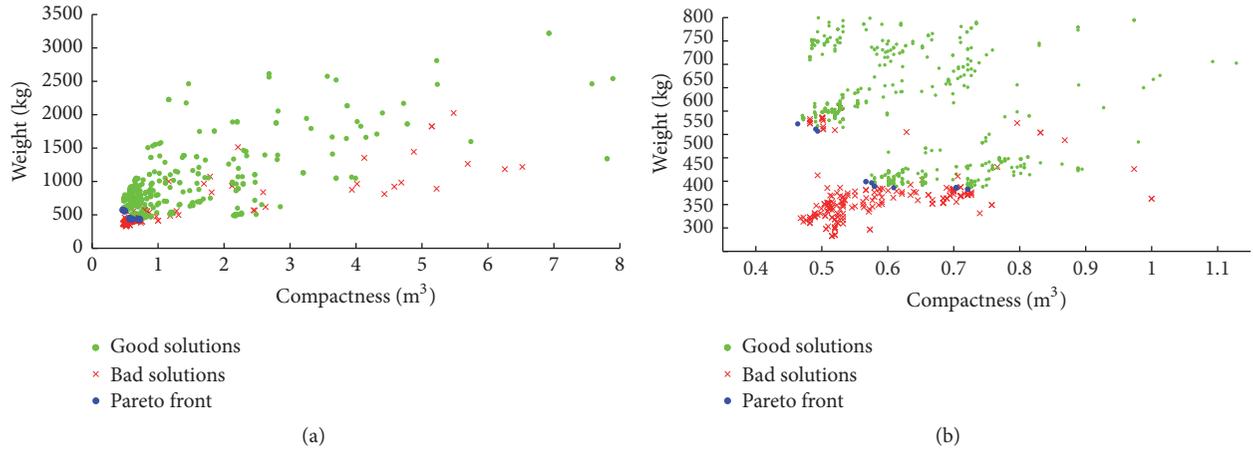


FIGURE 6: (a) The objective space for 60 individuals and 20 generations and (b) zoom on the Pareto front. The red crosses above the Pareto front have a low number of scissor units, which is the reason why they fail.

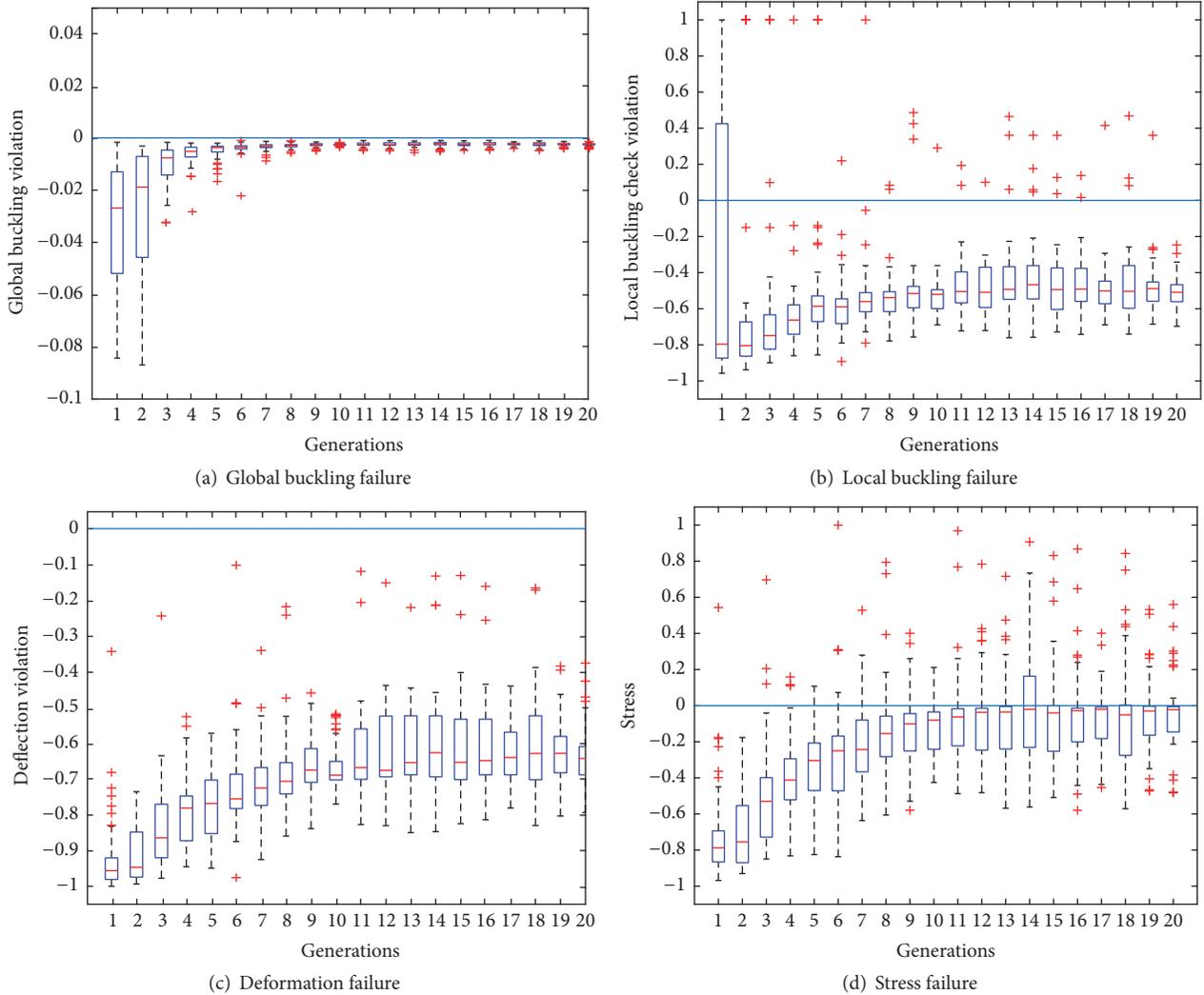


FIGURE 7: These figures show that global buckling is not an important constraint while the stress constraint is the most important one. The optimisation tries therefore to stay on the edge of the stress constraint.

between the actual and the limit value of the constraint. This means that all the individuals of the same constraint were divided with the highest value so that the results lie between  $-1$  and  $1$  and that each individual with a positive constraint value fails. For each generation  $x$ , the edges of the box are the 25th and 75th percentiles of the population for generation  $x$ , the central horizontal mark is the median, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually (red plus signs).

**4.3.1. Global Buckling.** The first step is to analyse the global buckling behaviour of the construction. As shown in Figure 7(a), there are no solutions that fail in global buckling. Note that the boundaries of the  $y$ -axis are between  $-0.1$  and  $0.05$  which means that the outliers are quite prominent. Indeed, as all individuals were normalised by dividing by the highest value (as explained in the introduction of Section 4.3), if most of the  $y$ -axis values are small, this means that the maximum which was used to perform normalisation is very high. This is not the case for the other constraints.

Global buckling is thus not the most important constraint; other constraints are more dominant. This is an important conclusion because it takes an important part of the computation time to make the global buckling calculation. By knowing that the global buckling calculation is not detrimental, this check can be removed from the optimisation loop (Figure 4) and only be checked a posteriori for valid solutions. For the next optimisation runs that are made with different numbers of individuals and generations (see Section 4.6), the buckling calculation is thus not included in the loop anymore. This causes a gain of 30% in computational time.

**4.3.2. Local Buckling.** The calculation of the local buckling is an analytical calculation based on the normal force and bending moments in the most loaded beam [23]. Figure 7(b) shows that besides for the first generation 75% of the population meets the constraint. The reason why the first generation has so many solutions failing in local buckling is the small number of units (5 as shown in the evolution of the variables). Until the last generation however, some individuals are still failing in local buckling. We can also observe that there is evolution towards solutions that are increasingly prone to local buckling (meaning that the cross section is changing to thinner/smaller sections). This local buckling check is thus necessary to consider but is not the most important one.

**4.3.3. Deformation.** The deformation failure has evolution that is quite identical to the stress evolution, with positive evolution of the deformation towards the limit but, from generation 10 on, the deformation stays quite constant and far under the limit. It could thus be stated that no individuals deform excessively. Hence, the serviceability limit state (SLS) of deformation is not the criterion on which the barrel vault is designed.

**4.3.4. Stress.** The stress constraint is the most important constraint as shown in Figure 7(d). It is clear that the design variables are optimised mainly in function of the stress criterion. During the first generations, too many solutions are satisfying the constraints. On the one hand, this means that feasible solutions are found, which is the aim of the optimisation. On the other hand, when compared to the other constraints, it seems that all of them are under the limit. The solutions are therefore feasible but not optimal. The optimal solutions are the ones that are just on the limit between failing and satisfying the constraints. Therefore, the algorithm tries to evolve to better solutions and this evolution is apparent in Figure 7(d), where the box is approaching the limit of 0 as the number of generations rises. Figure 7 shows thus clearly that the barrel vault is dimensioned based on the material strength.

**4.4. The Evolution of the Design Variables.** In this subsection, the evolution of the design variables is analysed. The evolution of the number of scissor units and the three parameters of the box cross section of the scissor units (height, width, and thickness) are shown in Figure 8. These figures give an overview of the way the design field is explored by the optimisation routine.

The variation in number of scissor units is visible in Figure 8(a). In a first generation, individuals varying from 5 to 10 scissor units are created with a median of 8. From generation 2 on, individuals with 5 units are no longer considered because, as already explained, the optimisation framework generates elements with a large length that fail in buckling. From then on, the median shifts from 8 to 7 (from generation number 5). This optimal number of scissor units of 7 can also be seen in the Pareto front in Figure 5. The evolution of the three parameters of the box cross section is shown in Figures 8(b)–8(d). As can be seen, the boxes are more shifted to the bottom side, especially for the height and the width. In the beginning, the height and the width are covering the whole range but as shown in the constraints this leads to too stiff and strong structures. This is the reason why these parameters are then reduced in order to have more optimal and light structures.

**4.5. Metrics for Comparing Nondominated Sets: Convergence Criterion.** In the first simulation, we opted for 20 generations. In this part, the behaviour of the optimisation routine will be analysed through some tools such as the generational distance, the hypervolume, and the rate of feasible solutions [35]. A convergence criterion will also be defined.

**4.5.1. Generational Distance (GD).** The GD represents how far in average the current set  $PF_{\text{current}}$  is from  $PF_{\text{true}}$  [36]. In Figure 9(a), a fluctuation of the general distance can be observed trending to a drop in GD. This value of GD will be used as a criterion for convergence of the Pareto sets for the next simulations that will be run. The criterion will be the following: if four consecutive Pareto fronts have a GD value lower than 0.01, this means that convergence has been obtained and that the optimisation loop can be stopped.

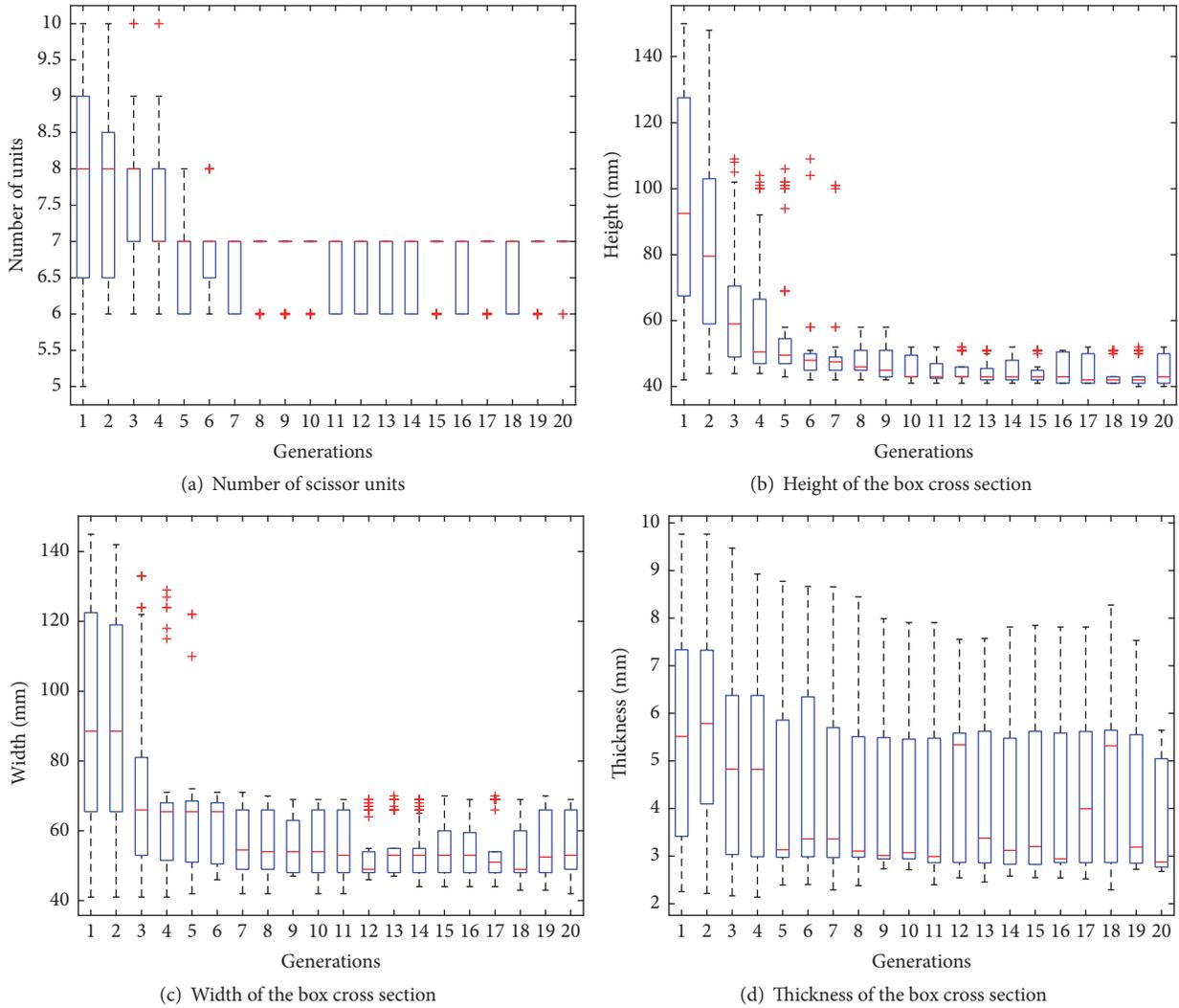


FIGURE 8: (a) shows that the number of units is optimal at 7. The three other design variables are evolving to the bottom side showing that the simulation is reducing the cross-sectional dimensions in order to obtain better solutions in the Pareto front.

4.5.2. *Hypervolume HR.* In biobjective problems, the hypervolume consists in calculating the area dominated by a set of solutions in the objective space and bounded by a reference point [37, 38]. Figure 9(b) shows the evolution of the hypervolume. As expected, the hypervolume increases with the generations to a value of 1.6. This value is obtained at generation 17. The hypervolume of generation 3 compared to generation 16 is illustrated in Figure 9(d).

4.5.3. *Rate of Feasible (RF) Solutions.* The rate of feasible (RF) solutions gives the percentage of individuals which satisfies all the constraints (Figure 9(c)). Generation 1 has fewer feasible solutions than the second generation, but, from then on, the rate of feasible solutions decreases. This is a desirable behaviour because it is important to have solutions which are failing. The latter means that the simulation is trying to find solutions of the Pareto front, on the edge between solutions that satisfy or fail the constraints.

4.6. *Influence of the Number of Individuals and Generations.* NSGA-II is a genetic algorithm and works with populations of individuals. The first step in the optimisation process is to generate solutions which cover the whole objective space, in order to have an idea of which individuals fail and which individuals meet the constraints. As the number of generations progresses, the individuals are clustered around clouds of good solutions, which converge to the Pareto front.

It may be expected that if the number of generations increases, the Pareto front will contain more points, thus less gaps, and that it will converge to the “theoretical” actual Pareto front. Secondly, it might seem better to have more generations than more individuals, because generations are combining the genes of good solutions to obtain better children. Drawing conclusions on this matter is, however, not so straightforward.

The influence of the number of individuals on the number of needed generations was therefore studied. Taking into account the convergence criterion (general distance

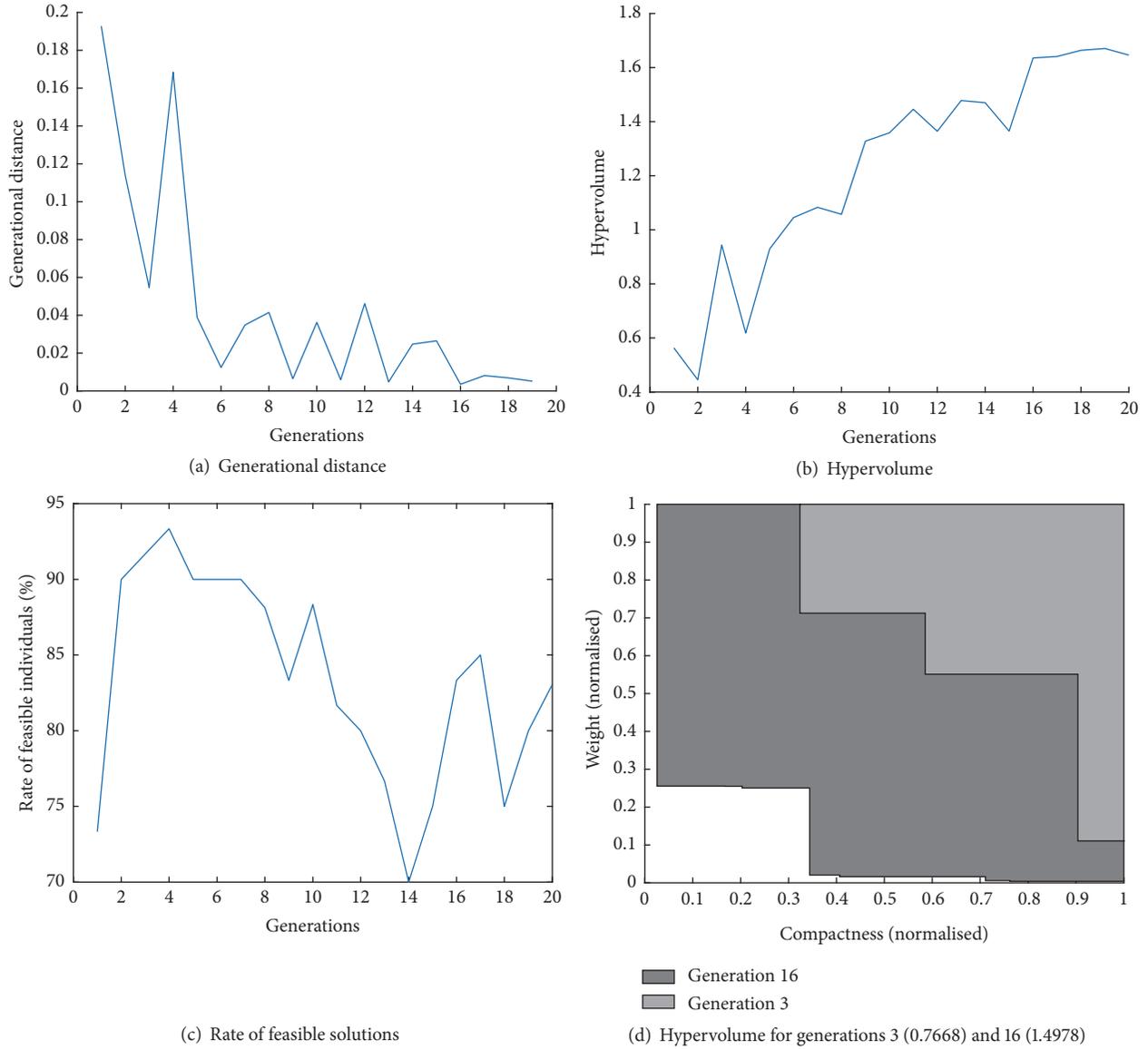


FIGURE 9: While the GD is dropping with the generations, the HR is rising and the RF is fluctuating. This means that as the generations progress, the Pareto fronts are coming closer to each other and their extent is becoming bigger while procuring enough solutions that are failing.

$\leq 0.01$  for four consecutive Pareto fronts as explained in Section 4.5), 8 simulations were run with a number of individuals varying from 20 to 160 in steps of 20 and the hypervolume of all the Pareto fronts was calculated (Table 4).

In general, it can be seen that the more the individuals, the smaller the number of needed generations until convergence but the higher the total number of iterations. It is therefore more time-consuming to opt for more individuals as the total number of calculated solutions will be higher (1440 for 160 individuals versus 420 for 20 individuals). Secondly, there is no correlation between the number of individuals, generations, or iterations and the hypervolume. For example, the hypervolume of 140 individuals is much higher than

TABLE 4: This overview shows that, in general, the more the individuals, the smaller the number of generations needed until convergence but the higher the number of total iterations.

# ind.	# gen.	# iterations	HR
20	21	420	93
40	16	640	78
60	12	360	287
80	13	1040	139
100	10	1000	171
120	13	1560	88
140	9	1260	441
160	9	1440	167

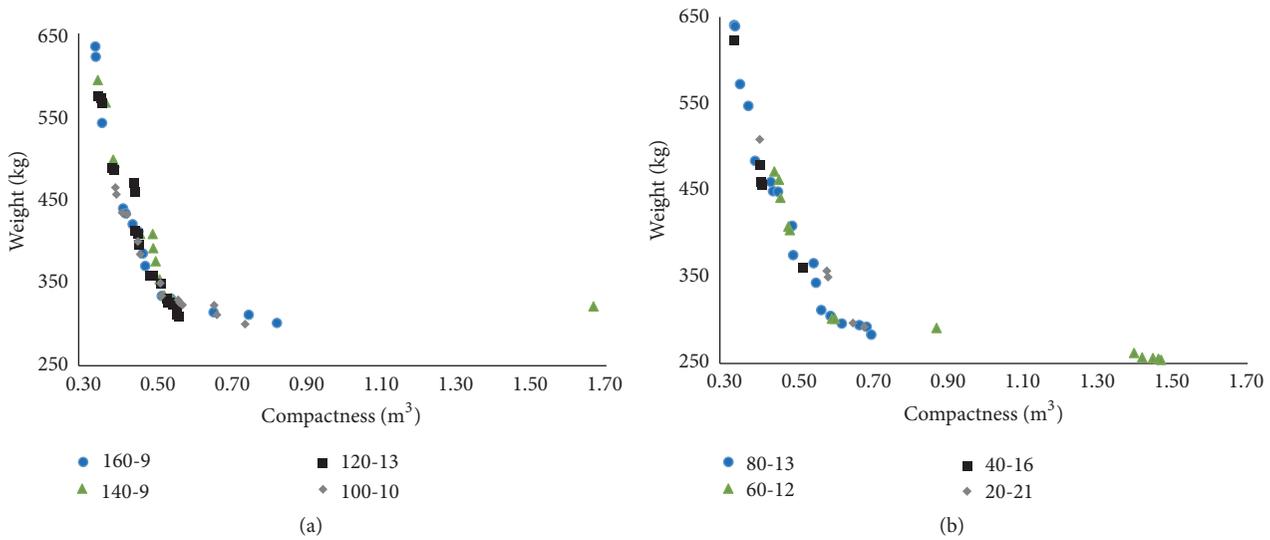


FIGURE 10: The Pareto fronts of the simulations with different numbers of individuals per generation show that the high value of the hypervolume of simulations 140-6 and 60-12 is related to the extent of the Pareto front and not to the quality of the latter.

the one of 160 individuals. In order to understand this phenomenon, the Pareto fronts of the different simulations have been plotted in Figure 10. The two largest hypervolumes (140-9 and 60-12) have a higher HR value because of the extent of their objective space (the green triangles go way further than the other Pareto solutions). However, it can be noted that in both cases the Pareto solutions lie globally higher than the Pareto solutions of the other cases (so a higher weight for the same compactness). It can thus be concluded that reaching for a broader objective space causes the solutions to be less “optimal” but it procures solutions with a great variety of design values. Comparing the hypervolumes of different simulations is therefore not the clear way of comparing simulations as the hypervolume is in fact a metric to show the expansion of the objective space and not the quality of the Pareto front.

In general, it can be said that the more the individuals, the smaller the gaps between Pareto solutions. The Pareto fronts are therefore denser, contain more optimal solutions, and have solutions that are closer to the theoretical Pareto front (e.g., note the difference between case 80-13 and case 20-21). It takes however more time to calculate the optimal solutions. Thus, the following important design conclusion can be formulated: if the aim is to have a couple of Pareto solutions inside a wide objective space in a quick way, the designer should opt for a small number of individuals with a minimum (in this case) of 60 individuals. Below 60 individuals, the Pareto front has too few solutions. On the other hand, if the aim is to have a dense and accurate Pareto front and if enough calculation time is provided, then it is best to opt for a higher number of individuals (e.g., 160).

### 5. Results of the Detailed Structural Optimisation Procedure

In Section 4, a simplified structural optimisation with fewer design variables and constraints (4 design variables and 7

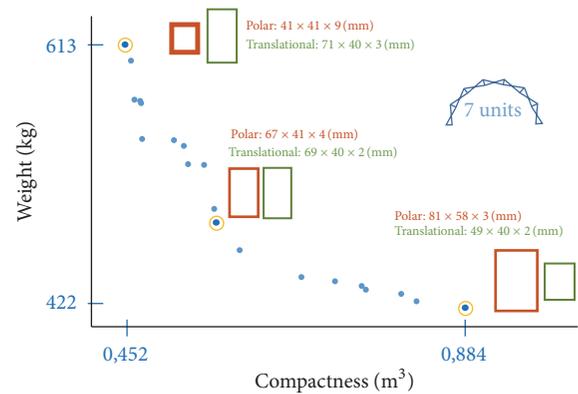


FIGURE 11: This Pareto front obtained with 160 individuals per generation reached convergence after 23 generations. It shows the design variables of the most extreme optima and of the solution that will be used for further research.

constraints) was run and discussed. For the sake of completeness of this study, the results of the full structural optimisation (with 7 design variables and 14 constraints) are shown in Figure 11. 160 individuals per generation were used and the optimisation needed 23 generations until convergence. Three Pareto solutions are shown, the two most extreme optima and the solution that will be used for further research on transformable barrel vault construction. The latter has been chosen because of the similarity between the outside cross-sectional dimensions of the polar and translational units (67 versus 69 mm and 41 versus 40 mm) which can greatly simplify the design of the needed joints.

### 6. Conclusion

This research paper has presented a general, open, and flexible methodology using Abaqus and Matlab to obtain a set of

optimal barrel vault scissor structure solutions in a fully automated framework without interference from the user. For the structural optimisation interface, several Python scripts are linked with Matlab (which contains the optimisation algorithm) and Abaqus (for the structural analysis). The fact that genetic algorithms are used gives, besides a wide search in the objective space, also the possibility of optimising other shapes of scissor structures, using discrete or continuous design variables. The objectives and the constraints can also be changed and the same methodology can be used even if the aim is the optimisation to one objective.

This paper shows the feasibility and capacity of this methodology by optimising a 6 m span barrel vault to weight and compactness and by obtaining a Pareto set of optimal solutions in an efficient way. The in-depth analysis of the influence of the optimisation variables on the results yields new insight which can help in making judicious choices with regard to the design variables, the constraints, and the number of individuals and generations in order to obtain efficiently a trade-off of optimal solutions.

As shown by the analysis of the design variables evolution with the indicators generational distance, hypervolume, and rate of feasible solutions, each solution of the Pareto set has an optimal number of scissor units of 7.

By mapping the constraints violations in function of the number of generations, the influence of the constraints was analysed; the conclusion is that stress is more dominant than global and local buckling and deformation. Because global buckling is not preponderant, the global buckling analysis was removed from the optimisation loop and only verified a posteriori for the optimal scissor structures. This means a significant gain in computation time (30%).

Studying the influence of the number of individuals and needed generations to obtain convergence (using the generation distance metric), it was found that the main clusters are mostly formed during the first generations, and, from then on, the new individuals are clustered into those clouds of good solutions. The more the individuals per generation, the smaller the number of needed generations until convergence but the higher the number of iterations, causing a higher computational time. Yet, a denser and smoother Pareto front is then obtained. In conclusion, the following design rule was formulated: if time is scarce and the aim is to have some Pareto solutions inside a wide objective space, the designer should opt for a small number of individuals per generation (minimum of 60 individuals). However, if the aim is to have a dense and accurate Pareto front, a higher number of individuals per generation are necessary. The final Pareto front of the full structural optimisation has been found with 160 individuals and 23 generations.

## Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

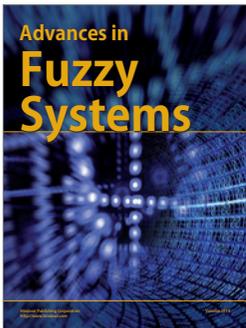
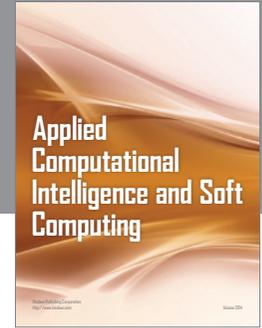
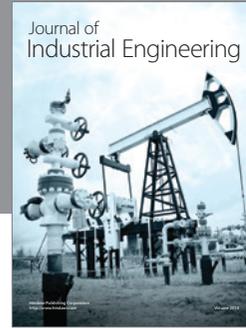
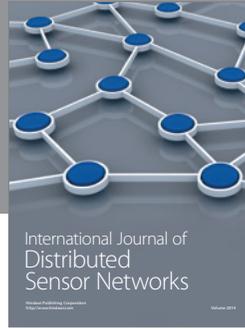
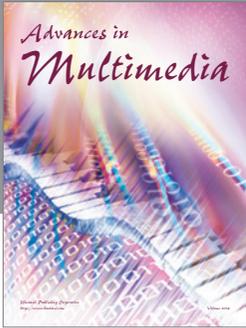
## Acknowledgments

The authors would like to thank the Red Cross and Médecins Sans Frontières for providing them with useful information about their expertise in the field of sheltering.

## References

- [1] T. Corsellis and A. Vitale, *Transitional Settlement Displaced Population*, Oxfam, Oxford, UK, 2005.
- [2] E. P. Pinero, *Project for a Mobile Theatre Architectural Design*, vol. 12, 1961.
- [3] C. Gantes, *Deployable Structures: Analysis and Design*, WIT Press, 2001.
- [4] Y. Akgün, *A novel transformation model for deployable scissoring structures [Ph.D. dissertation]*, University of Stuttgart, 2010.
- [5] N. De Temmerman, *Design and analysis of deployable bar structures for mobile architectural applications [Ph.D. dissertation]*, Vrije Universiteit Brussel, Brussels, Belgium, 2007.
- [6] F. V. Jensen, *Concepts for retractable roof structures [Ph.D. thesis]*, University of Cambridge, Cambridge, UK, 2004.
- [7] A. Koumar, T. Tysmans, N. De Temmerman, R. F. Coelho, and L. A. Mira, "Multi-criteria optimisation of a barrel vault structure for emergency relief," in *Proceedings of the Symposium: Shells, Membranes and Spatial Structures: Footprints (IASS-SLTE '14)*, Brasilia, Brazil, September 2014.
- [8] A. Kaveh and A. Davaran, "Analysis of pantograph foldable structures," *Computers and Structures*, vol. 59, no. 1, pp. 131–140, 1996.
- [9] A. Kaveh and S. Shojaei, "Optimal design of scissor-link foldable structures using Ant Colony Optimization algorithm," *Computer-Aided Civil and Infrastructure Engineering*, vol. 22, no. 1, pp. 56–64, 2007.
- [10] C. J. Gantes, P. G. Georgiou, and V. K. Koumousis, "Optimum design of deployable structures using genetic algorithms," in *Proceedings of the InSmart Structures Conference*, pp. 255–264, 1998.
- [11] A. Kaveh and S. Shojaei, "Discrete-sizing optimal design of scissor-link foldable structures using genetic algorithm," *Asian Journal of Civil Engineering*, vol. 4, no. 2–4, pp. 115–133, 2003.
- [12] A. Kaveh, A. Jafarvand, and M. A. Barkhordari, "Optimal design of pantograph foldable structures," *International Journal of Space Structures*, vol. 14, no. 4, pp. 295–302, 1999.
- [13] A. P. Thrall, "Hangai prize papers for 2011: shape-finding of a deployable structure using simulated annealing," *Journal of the International Association for Shell and Spatial Structures*, vol. 52, pp. 241–224, 2011.
- [14] A. P. Thrall, M. Zhu, J. K. Guest, I. Paya-Zaforteza, and S. Adriaenssens, "Structural optimization of deploying structures composed of linkages," *Journal of Computing in Civil Engineering*, vol. 28, no. 3, 2014.
- [15] L. A. Mira, A. P. Thrall, and N. De Temmerman, "The universal scissor component: optimization of a reconfigurable component for deployable scissor structures," *Engineering Optimization*, vol. 48, no. 2, pp. 317–333, 2016.
- [16] Y. R. Elhaddad, "Combined simulated annealing and genetic algorithm to solve optimization problems," *World Academy of Science, Engineering and Technology*, vol. 68, pp. 1508–1510, 2012.

- [17] S. N. Kumbharana and G. M. Pandey, "A comparative study of ACO, GA and SA for solving travelling salesman problem," *Societal Applications of Computer Science*, vol. 2, no. 2, pp. 224–228, 2013.
- [18] M. Alhanjouri and A. Belal, "Ant colony versus genetic algorithm based on travelling salesman problem," *International Journal of Computer Technology and Applications*, vol. 2, no. 3, pp. 570–578, 1930.
- [19] N. D. Lagaros, M. Papadrakakis, and G. Kokossalakis, "Structural optimization using evolutionary algorithms," *Computers and Structures*, vol. 80, no. 7-8, pp. 571–589, 2002.
- [20] Yannick Garbusinski, Quote, 2013.
- [21] European Committee for Standardisation, *Eurocode 1: Actions on Structures*, CEN, 2001.
- [22] European Committee for Standardisation, *Eurocode 0: Basis of structural design*, CEN, 2002.
- [23] European Committee for Standardisation, *Eurocode 9: Design of aluminium structures*, CEN, 2007.
- [24] Abaqus, *ABAQUS Documentation*, Dassault Systmes, Providence, RI, USA, 2012.
- [25] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Genetic and Evolutionary Computation Series, Springer, New York, NY, USA, 2nd edition, 2007.
- [26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [27] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.
- [28] O. Shahvari, N. Salmasi, R. Logendran, and B. Abbasi, "An efficient tabu search algorithm for flexible flow shop sequence-dependent group scheduling problems," *International Journal of Production Research*, vol. 50, no. 15, pp. 4237–4254, 2012.
- [29] O. Shahvari and R. Logendran, "Hybrid flow shop batching and scheduling with a bi-criteria objective," *International Journal of Production Economics*, vol. 179, pp. 239–258, 2016.
- [30] O. Shahvari and R. Logendran, "An enhanced tabu search algorithm to minimize a bi-criteria objective in batching and scheduling problems on unrelated-parallel machines with desired lower bounds on batch sizes," *Computers & Operations Research*, vol. 77, pp. 154–176, 2017.
- [31] E. M. Segal, L. Rhode-Barbarigos, S. Adriaenssens, and R. D. F. Coelho, "Multi-objective optimization of polyester-rope and steel-rope suspended footbridges," *Engineering Structures*, vol. 99, pp. 559–567, 2015.
- [32] E. M. Segal, L. Rhode-Barbarigos, R. D. F. Coelho, and S. Adriaenssens, "An automated robust design methodology for suspended structures," *Journal of the International Association for Shell and Spatial Structures*, vol. 56, no. 4, pp. 221–229, 2015.
- [33] M. De Munck, S. De Sutter, S. Verbruggen, T. Tysmans, and R. F. Coelho, "Multi-objective weight and cost optimization of hybrid composite-concrete beams," *Composite Structures*, vol. 134, pp. 369–377, 2015.
- [34] MATLAB, *Version 8.4.0.150421 (R2014b)*, The Math Works Inc, Natick, Mass, USA, 2014.
- [35] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.
- [36] D. A. Van Veldhuizen and G. B. Lamon, "Evolutionary computation and convergence to a pareto front," in *Proceedings of the Late Breaking Papers at the Genetic Programming Conference*, 1998.
- [37] J. D. Knowles, *Local-search and hybrid evolutionary algorithms for pareto optimization [Ph.D. thesis]*, Department of Computer Science, University of Reading, Reading, UK, 2002.
- [38] R. Filomeno Coelho, "Bi-objective hypervolume-based Pareto optimization," *Optimization Letters*, vol. 9, no. 6, pp. 1091–1103, 2015.



**Hindawi**

Submit your manuscripts at  
<https://www.hindawi.com>

