

# Part III

# Applications of Computational Search



# 10

## Shell Structures

### 10.1 Concrete Parabola-based Bridge Benchmark

The first use of Genetic Search in this PhD thesis parametric a single-objective GA used for a concrete shell bridge\*. This first search process parametric developed as a Benchmark for the Genetic Algorithm, to test its ability to find optimal and sub-optimal individual solutions an a simple structural problem. In order to track the evolution and results of the GA a “fitness landscape” of the same structural problem is plotted and the GA evolution mapped on top of it.

#### 10.1.1 Parametric Model

The Parametric model for this example is the exact same model shown in chapter 9 described by equation 9.3 and shown in figure 9.1.

As it was previously explained in chapter 9, this parametric definition of the surface guarantees that, by varying the values of  $h_1$  and  $h_2$ , different configurations can be obtained: a completely flat surface 9.2a; positive double curvature surfaces 9.2c; negative double curvature surfaces 9.2d; and single curvature surfaces 9.2b.

---

\*This example was developed by the Author and Mario Sassone and is published in a Chapter of the Book “Shells for Architecture: Form finding and structural optimization” edited by Sigrid Adriaenssens, Philippe Block, Diederik Veenendaal and Chris Williams (Méndez Echenagucia et al. 2014- IN PRINT). The Chapter entitled Computational Morphogenesis was written by the Author, Mario Sassone and Alberto Pugnale.

Since these two parameters effectively control the overall shape of the bridge,  $h_1$  and  $h_2$  are chosen GA the parameters. For both we have established a domain spanning from -40 m to +40 m. This means that the search space of this problem is two-dimensional and can be represented as a grid of values from 40 to 40 in its X and Y axis. We can thus assume that a vector of genetic variables  $x$  is made up of two variables:  $x_1 = h_1$  and  $x_2 = h_2$ .

The parametric definition proposed above is then implemented in the geometry modeler. When the GA calls for a shape in terms of a set of  $x_1$  and  $x_2$ , the CAD modeler generates the corresponding surface following equation 9.3, providing the object to be evaluated. However, for the FEM analysis a discrete model has to be generated. The geometry is discretized into a mesh, composed of shell or beam elements, depending on the type of structure. For this exercise, the shell is simplified as a grid-shell of comparable mechanical properties.

### 10.1.2 Structural Fitness Function

The construction of an FEM model for structural optimization, presents some differences with the ones used in normal analysis. Complex models require time consuming calculations, which represent bottlenecks in the flow of operations. The first requirement of FEM models for optimization is to be simple, with a number of elements strictly necessary and with a mesh correctly defined to evaluate the pertinent aspect of structural behavior. Even with powerful hardware setup, the repetition of hundreds, or even thousands of analyses might transform the optimization process into an extremely long task, if the model is not efficient.

There are basically two possibilities: (i) the use of customized finite elements solvers, developed in the same environment and (ii) the use of external applications, like commercial software. Both the alternatives have advantages and disadvantages, for this benchmark we will employ the first alternative. In shell analysis an important issue is the choice of the elements to use: in fact, even for a simple non-layered elastic shell, different formulations and approaches can be adopted. In the proposed application, the shell is approximated by a mesh of one dimensional beam elements, which geometric properties are defined in order to reproduce the characteristics of a continuous shell. This allowed to use a custom Python FEM code developed by Mario Sassone and capable of interacting seamlessly with the parametric modeler.

Displacements, strains, stresses, and strain energy are basically the effect of a load condition on an elastic structure. A stiff structure will show small

displacements and strains, while a strong structure will result in relatively small stresses, and both will have small strain energy. Displacements are a vector field, stresses are a tensor field, locally defined, and the strain energy is a scalar value, computed as an integral over the whole structure. Such quantities can be adopted as a measure of the structural performance. However, their differences will drive the optimization process to search for different optimal solutions. For this exercise, the maximum displacement of the whole structure is chosen as the fitness function to be minimized by the GA. As opposed to the strain energy, nodal displacements can reveal local, as well as global, weaknesses. Our fitness for this benchmark will be:

$$Fit_{structure} = \max(\Delta_{Z_i}) \quad (10.1)$$

where  $\Delta_{Z_i}$  is the deformation in the  $Z$  axis of the  $i^{th}$  node.

### 10.1.3 Fitness Landscape

The structural problem proposed for this benchmark is defined by only two parameters. This is purposely done in order to track and further explain the work done by the GA using a graphical representation. We first map out the solution domain of  $x_1$  and  $x_2$  by taking a two-dimensional parameter grid with grid points  $P(x_1, x_2)$ . By assigning a  $z$  value to each point of the grid, we convert it into a three-dimensional surface in which the  $z$  value represents the fitness calculated for the shape (individual) corresponding to that grid point P. For example:

- $P_{(0,0)}$  represents the completely flat surface, it has a maximum displacement in the z-axis (our fitness value) of 285 mm;
- $P_{(26,1)}$  has a maximum displacement in z of 21 mm;
- $P_{(40,40)}$  has a maximum displacement in z of 43 mm.

By repeating this operation for a series of individuals obtained by the discretization of the search space, we end up with a complete surface. This kind of three-dimensional representation of our problem and solutions is called a fitness landscape. Figure 10.1 shows the fitness landscape obtained in our Benchmark, which has many local minima, two global minima and an area of global maxima. This means that the problem can be considered a multimodal problem. Multimodal problems are notoriously difficult for gradient based or non-stochastic search methods, but they should be well within the possibilities of GA.

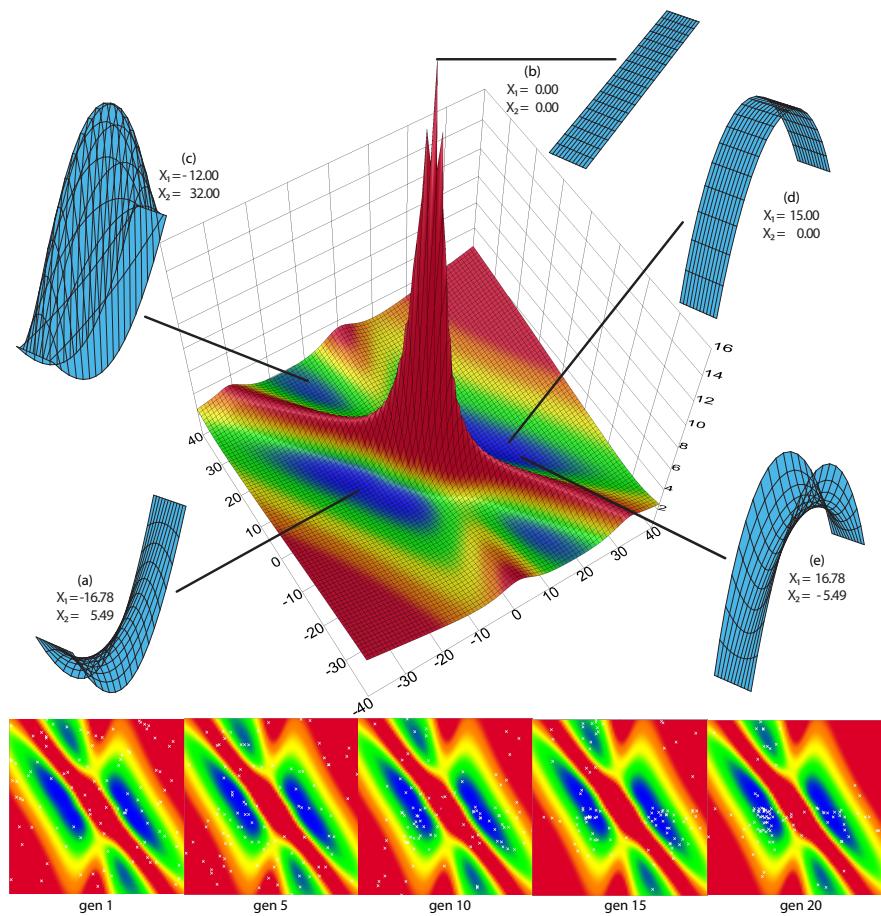


Figure 10.1: Fitness Lanscape for Parabola-based double curvature Benchmark, significant individuals and Genetic Algorithm evolution.

#### 10.1.4 Genetic algorithm inputs

The GA used in this benchmark employs a binary coding of design parameters with mutation and elitism operators. It terminates after 20 generations with a population size of 100 individuals. The two variables ( $x_1$  and  $x_2$ ) are here coded into eight digit binary numbers or genes. Such genes are combined into one chromosome with the  $x_1$  value positioned first and the  $x_2$  second as explained in chapter 6. The single point crossover operator is used. We can resume the GA inputs of this benchmark in the following table:

Population Size ( $N$ )	100	
Number of Variables	2	
Number of binary digits	8 for $x_1$	
Variable Domains	$x_1 \in [-40, 40]$	
Mutation Probability ( $p_m$ )	0.1	
End Condition	Number of Generations	
	20	

#### 10.1.5 Results

In Figure 10.1, we see a three-dimensional representation of the fitness landscape for this benchmark. Such a graphical tool allows us to study the exploration performed by the GA within the solution domain and to evaluate its efficiency by mapping generation by generation the search progress.

The plan views of the fitness landscape at the bottom of figure 10.1 show the evolution of the GA at generations 1, 5, 10, 15 and 20. The individuals being considered by the GA are represented by small white exes on top of the plan views of the fitness landscape. We can see how the GA gradually concentrates its individuals in areas of interests, most particularly global and local minima seen in blue.

The figure also shows different configurations of the bridge structure in relevant points of the fitness landscape. It is of particular interest to see how the shapes of the local minima differ from one another, even if they possess similar fitness values. The search process aims to find global minima, but we have seen with this exercise that even other sub-optimal candidate solutions might be worth considering:

- Individual (a) is the global minimum as found by the GA after 20 generations. It is a hypar with a maximum displacement of 169mm. Because of the symmetrical nature of the problem, we can say that

individual (e) is the symmetric opposite to individual (a) and becomes a global minimum as well.

- Individual (b) represents the global maximum, a flat surface with a maximum displacement of 285mm.
- Individual(c) is a local minimum with very tall parabolas forming an highly irregular hypar.
- Individual (d) is a near optimal single curvature configuration.

This benchmark showed that the single objective GA is able to quickly point out global and local minima in a simple structural problem. Already in generation 5, the GA had found the global minima, and by generation 20 it had explored a much wider area, point out other areas of interest in the fitness landscape, thus demonstrating that the GA is an appropriate tool for multimodal problems.

The fitness landscape is proven to be an interesting and effective tool in the study of performance related problems. Its main drawback is the fact that it is limited to a two-variable problem, otherwise there would not be enough dimensions to properly represent the problem and its results. Another important drawback of this method is the calculation times. Since, in order to have a detailed landscape, a big number of simulations have to be carried out, it can be considered a “Brute Force” or exhaustive search method, and therefore its efficiency is not very high. However, for simple problems with small domains it proved to be an interesting tool for the study of parametric models combined with performance simulations.

## 10.2 Case Study 1: Concrete free-form Roof

After testing the GAs ability to explore a wide search space, and a multimodal objective space to find local and global minima, it was time to take one step further, and develop a multi-objective structural problem. A multi-objective search problem with contrasting objectives for structural design is developed and solved using NSGA-II<sup>†</sup>. The problem consists in the design of a 24 × 24 meter roof supported at its corners with a fixed and continuous thickness, with the objective of making it as rigid and as light as possible.

---

<sup>†</sup>This Case study was published as a part of an article on the International conference on Structures and Architecture 2013 in Guimarães, Portugal (Méndez Echenagucia, Pugnale & Sassone 2013).

So when compared to the previous benchmark, this case study has the important difference of being a multi-objective problem, and additionally, this case study does not employ the same parabola based parametric study. It employs more complex geometric possibilities involving NURBS geometry.

### 10.2.1 Parametric Model

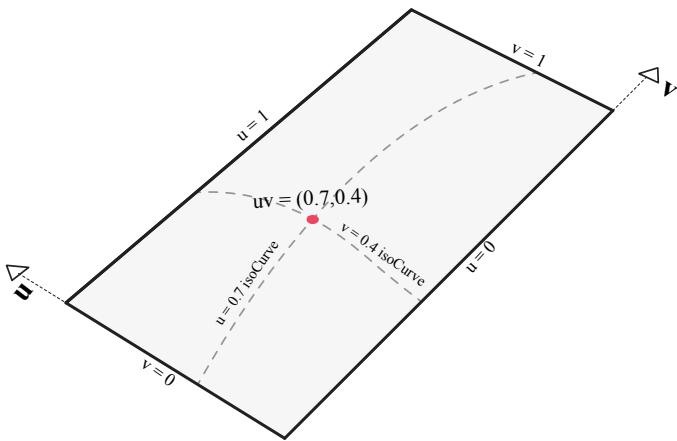


Figure 10.2: Non-Uniform Rational B-Splines -  $U$  and  $V$  parametrized surface.

In this case study the possibilities desired for the solution space went beyond what was possible with the previously studied parabola-based surfaces. Apart from single and double curvature surfaces, this case study parametric to include multiple curvature surfaces, often referred to as free-form geometry.

Non Uniform Rational B-Splines (NURBS) were introduced in the late 1970's mainly from the work of Pierre Bézier and Paul de Casteljau both working in the french automotive industry. They are the current standard for describing curves and surfaces in computer aided design (Rogers 2001). The representation of NURBS surfaces requires the use of two parameters commonly  $U$  and  $V$ , and so the  $X, Y, Z$  coordinates of any given point can be thought of as functions of  $U$  and  $V$ , e.g.  $x = x_{(U,V)}$ ;  $y = y_{(U,V)}$ ;  $z = z_{(U,V)}$ . So we can say that any point in this surface can be described by

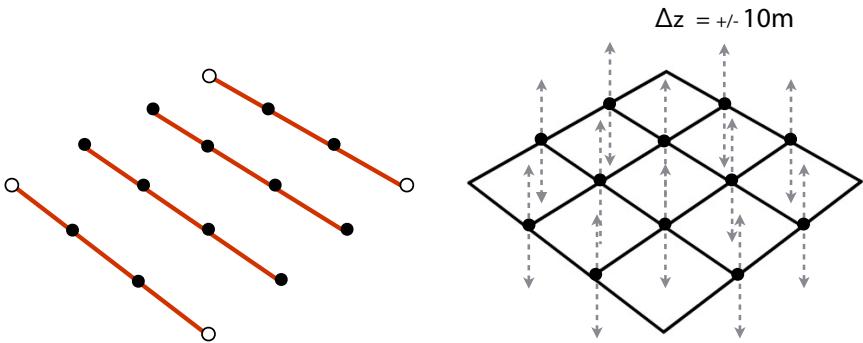


Figure 10.3: Parametric Model for the  $24 \times 24$  roof problem.

its bi-parametric coordinates  $U$  and  $V$  (figure 10.2). This is a very similar approach when compared to the one described for the parabola-based surface in equation 9.3. The most important difference being that the NURBS surfaces uses not simple parabolas, but B-splines, capable of describing just about any curve we could think of, with just a few control points.

The NURBS geometry used throughout this PhD thesis is generated with the aid of the Rhinoceros, a commercial CAD software that contains a large library of functions that generate, edit and analyze NURBS. Rhinoceros also offers its users the possibility to customize the use of its functions by writing scripts that can access functionality in various ways and with multiple programming languages. One of these languages is Python, the same language we have been employing to develop our search algorithms and FEM solvers. This means that we can seamlessly call geometric functions as easily as we can call search or FEM functions.

Figure 10.3 shows the parametric model for this case study. The surface for this case study is generated by means of four NURBS curves (shown in red). These curves in turn define the NURBS surface that is the subject of study. The NURBS is built starting from a set of four spatial curves. The two curves at the ends have its end points fixed (seen represented as white filled dots in figure 10.3, while the other two are NURBS curves laying in vertical planes. As it will be shown below, the fixed points correspond to the structure supports. The four curves act as four vertical sections of the surface to be generated. The surface is defined as a NURBS passing through the section curves (this is called a lofted surface), with assigned polynomial

degree. Each curve is defined by four interpolating points, whose vertical position, the  $Z$  coordinate, is variable. Each variable point can move within the confines of a vertical line 20 meters long, they can move 10 meters above or 10 meters below the starting point  $Z = 0$ . By modifying the coordinate of the interpolating point, the section curves change and so does the surface. In such a way, a set of 12 real numbers is used to completely define the surface shape. The other NURBS parameters of the surface, as the degree of interpolating functions or the number and position of control points, are set constants in the problem.

We have established that there are a total of 16 control point in the surface, but 4 of them are not variable. Also we established that only the  $Z$  coordinate of these points is variable, so we can say that the number of variables in this problems is equal to 12. This gives us a vector of variables  $X$  where:

$$X = z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9, x_{10}, z_{11}, z_{12} \quad (10.2)$$

We can also see this vector in a more general way:

$$X = x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12} \quad (10.3)$$

where in this case  $x_{i_{th}} = z_{i_{th}}$ .

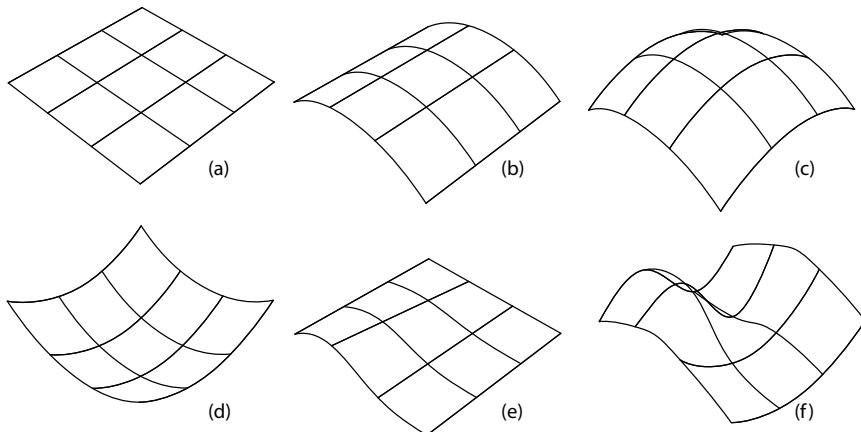


Figure 10.4: Possible individuals with the parametric model developed for the roof Problem.

The ranges of variability of each parameter define the set of potential solutions. As it has been already said, the length of the variable domain influences the search process and the possible outcomes. Figure 10.4 shows six of the possible surfaces contained within the above defined parametric model. Surface (a) is a flat surface, (b) is a single curvature surface, while (c) and (d) are positive double curvature surfaces in different directions, and (e) and(f) are surfaces with multiple curvatures, meaning they have both positive and negative curvatures within the same surface. Surface (e) has a complex curvature that is not very accentuated, while (f) has more pronounced curves.

### 10.2.2 Fitness functions

One of the important premisses for this Case Study is the use of multiple and contrasting objective functions. In particular for this case we want to search for rigid but light concrete structures. For every generated solution, two functions are used to calculate its fitness.

The first function is the same one used in the above benchmark, the maximum displacement in the z-axis. The FEM solver used in this example is also the same one used for the Benchmark, it is the FEM solver developed by Mario Sassone in the Python programming language. This enables a smooth communication between the parametric model, the FEM solver and the Search Algorithm. However, in this case, the geometry is not generated with the parabola function, but with the NURBS parametric model mentioned above, which leaves us with the problem of discretization of the otherwise continuous NURBS surface.

The FEM solvers used in this research are not capable of working with NURBS geometry. Geometry must be represented in small and flat shell elements. This means that we need to split up our continuous NURBS surface into small flat elements. Depending on the type of geometry, the only way to make sure that the resulting elements are flat is to generate triangular elements, not all of the resulting rectangular elements might be flat. Since we are working with free-form geometry, we discretized the NURBS surface into triangular elements using a set of Rhinoceros functions that were written for this exact purpose. The proper discretization of continuous surfaces can be a relatively simple task using this set of Rhinoceros functions, but as we will see later on this is not the case for surfaces containing windows or gaps.

The resulting discretized Mesh surface is loaded in its nodes with only a vertical load as shown in figure 10.5. The surface is only supported at its corners, that in this case are 24 meters apart in both directions. The support

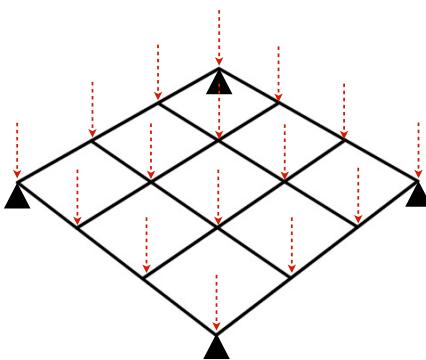


Figure 10.5: Loading and Node Constraint conditions for the  $24 \times 24$  roof problem.

points are assumed to have pin joints, so they will not offer resistance to rotation.

The second fitness function measures the surface's weight. This function could be calculated by taking the surface's area and multiplying it by a fixed thickness (thus obtaining volume) and consequently multiplying this volume by the proper cubic weight of its material, concrete. However, since both material weight and thickness is the same for all surfaces, this function is simplified and the only value taken into consideration is the surface's area, the only variable in the calculation explained above.

We can sum up this multi-objective problem in the following way:

$$\text{Case Study 1} \left\{ \begin{array}{l} \text{Minimize } f_1(x) = \max(\Delta_{Z_i}), \\ \text{Minimize } f_2(x) = S, \\ \text{subject to } -10 \leq x_i \leq 10. \end{array} \right. \quad (10.4)$$

where  $x$  is the vector of variables,  $\Delta_{Z_i}$  is the displacement in the Z-axis of the  $i_{th}$  node and  $S$  is the surface area.

### 10.2.3 Genetic algorithm inputs

This case study's genetic inputs were the following:

<b>Case Study 1 Run#</b>	1
Population Size ( $N$ )	10
Number of Variables	12
Number of binary digits	8
Variable Domains	$x_i \in [-10, 10]$
Mutation Probability ( $p_m$ )	0.1
End Condition	End after 500 generations

#### 10.2.4 Results

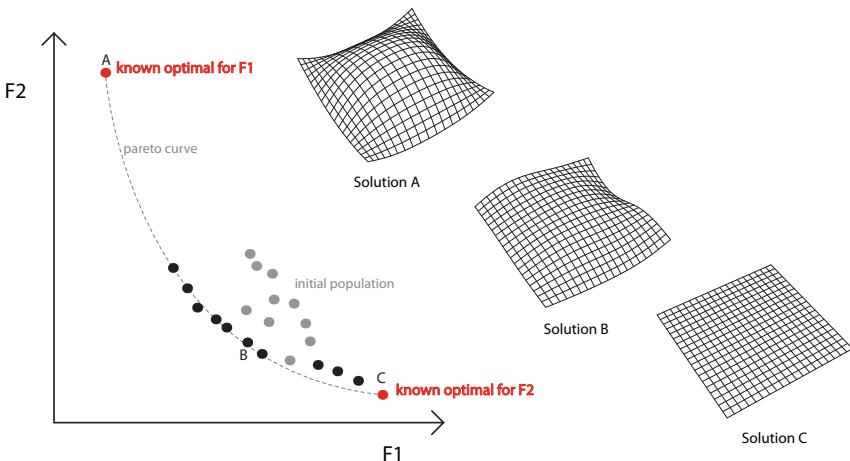


Figure 10.6: Objective space and Pareto front - 3 individual solutions for the roof structural multi-objective search problem.

Figure 10.6 shows the results of the search process for case study 1. The Pareto front, represented in the fitness space, contains the best solutions found by the algorithm: the lower branch of the curve contains solution that privilege lightness to stiffness, while in the left branch stiffer but heavier configurations can be found. In this benchmark, the best solution related to each fitness is known: the lightest shape is the flat shape, while the stiffer shape is a dome with the four central point at the top of the domain and the eight lateral points at the bottom. During the search process, the crowding distance algorithm tries to keep a good spacing between solution in

the front, but the two extremes (A and B) were not found. The knowledge of such extremes allows us to evaluate the efficiency of MOGA in terms of ratio between the size of the found front and the size of the actual front, including extremes.

The solutions spacing in the front is a good indicator of the variety of geometrical shapes and of the way different shapes answer to multi-objective requirements. A set of such shapes, related to the position on the Pareto front, is also shown in figure 10.6, together with the extreme cases. A direct representation of the position of shapes in the variable space is not suitable, due to the number of dimensions, but they can be compared one to another by the designer who is in charge to handle the produced material.

The fact that the extreme solutions (A and B) were not found could be due to an insufficient exploration during the search process (low number of individuals in the population or due to a small number of generations relative to the problem complexity) Since in this problem we do not know the exact shape of the true Pareto front, we cannot be 100% sure about the problem complexity, the true Pareto front could be almost discontinuous in the segments between the Front found and the A and B solutions. Meaning that there wold be no feasible solutions in those missing segments, and only one single solution in the extreme. This would make finding the extremes A and B highly difficult. What we can say for sure, is that the exploration in this case parametric insufficient for the extremes to be found, we cannot be sure about the problem complexity.

Solution (B) is an interesting compromise between rigidity and weight. It is a complex surface with multiple curvatures that assure its rigidity, but also it does not posses a high surface area, making it not very heavy, especially when compared with other more rigid solutions.

In order to better understand the relationship between problem complexity and number of calculations, the search algorithm is executed two more times with different GA inputs. The GA inputs for the second calculations were:

<b>Case Study 1 Run#</b>	2
Population Size ( $N$ )	30
Number of Variables	12
Number of binary digits	8
Variable Domains	$x_i \in [-10, 10]$
Mutation Probability ( $p_m$ )	0.1
End Condition	End after 160 generations

Meaning that there were close to 5.000 calculations (the same number

as the first run, but with a higher population size). And the third time the algorithm ran, the inputs were:

<b>Case Study 1 Run#</b>	3
Population Size ( $N$ )	100
Number of Variables	12
Number of binary digits	8
Variable Domains	$x_i \in [-10, 10]$
Mutation Probability ( $p_m$ )	0.1
End Condition	End after 100 generations

Meaning that there were 10,000 calculations, with a larger population size than the two previous search runs. Figures 10.7 and 10.8 show the Pareto Front and initial population for all three search runs. We can see that when compared to the first run, the second and the third have a slightly improved coverage of the Pareto front, since they are both closer to the two extremes (A and B). The most significant improvement is made in the direction of solution A, the improvement in the direction of A is not as noticeable. The Pareto front also seem to be slightly better, having solutions that are would dominate solutions in the first run. However, if we compare run 2 with run 3, the difference is hardly noticeable. We could even say that the second run has a better coverage of the Pareto Front.

Since there is a lot of randomness in NSGA-II, we cannot say for sure what the causes are for these differences in the search runs, but we can draw some observations. There is a small difference between the second and third runs in favor of the second, the third run having twice as many calculations as the second. This suggests that increasing the number of calculations is not a guarantee of improvement. Run 1 had higher number of generations than run 2, but an apparently insufficient number of individuals in the population. Run 3 had a larger population size than run 2, but a lower number of generations, and the data suggests that this run would have benefited form a few more generations.

There seems to be a significant relationship between population size and number of generations. Too many individuals in the populations seems to increase calculation times needlessly, while a small number of generations is also not appropriate. We can assume that these values are highly problem dependent, and that achieving an optimum number of generations and population size is not easy. This might also allude to the balance between exploration and exploitation described above. Higher individuals in population denote higher exploration, while higher number of generations are sings of higher exploitation.

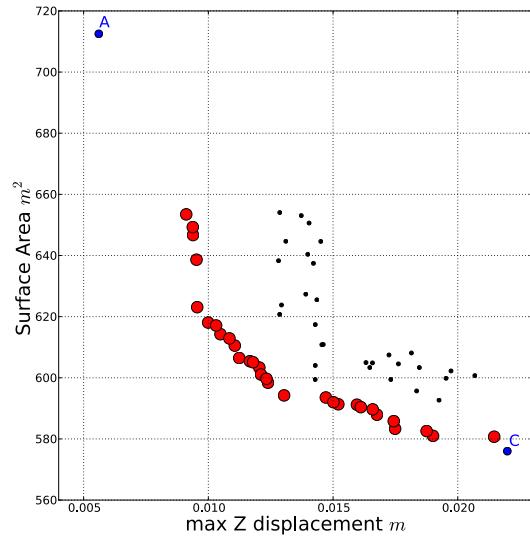
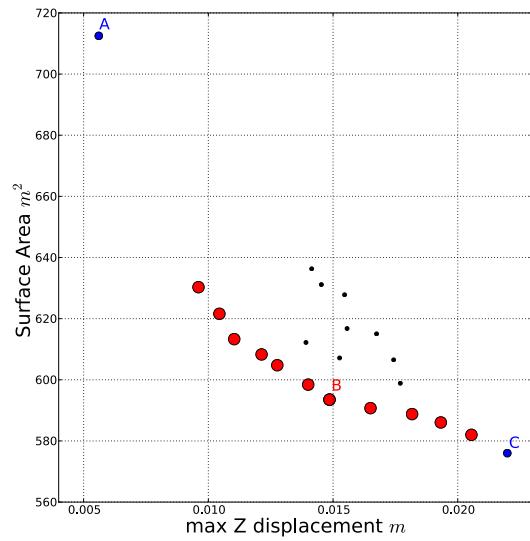


Figure 10.7: Objective spaces with Pareto fronts and initial population for Case study runs 1,2 and 3.

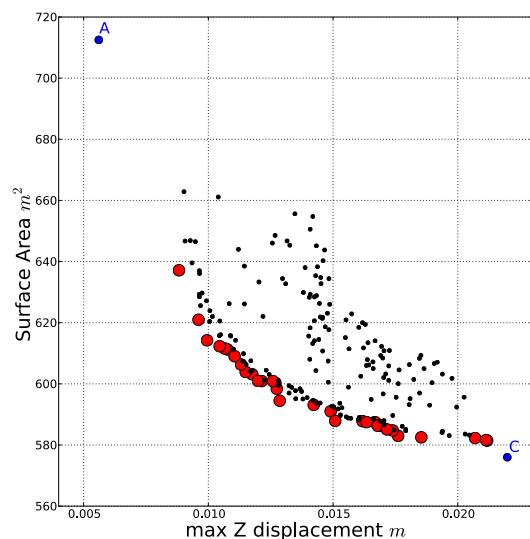


Figure 10.8: Objective spaces with Pareto fronts and initial population for Case study runs 1,2 and 3.

## 10.3 Case Study 2: Concrete free-form Bridge

The second case study shares many characteristics with the first one. It is also a multi-objective search problem with contrasting objectives for structural design and it is also studied using NSGA-II<sup>‡</sup>. The problem consists in the design of a  $24 \times 4$  meter bridge supported at its ends with a fixed and continuous thickness, with the objective of making it as rigid and as light as possible. It follows the exact same premise as the benchmark but it uses free-form geometry.

### 10.3.1 Parametric Model

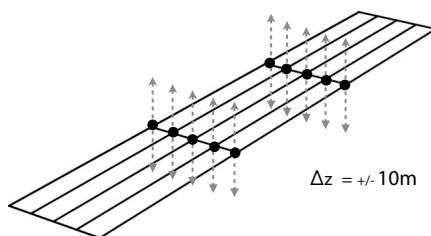


Figure 10.9: Case Study 2 Parametric Model -  $24 \times 4$  Bridge.

Figure 10.9 shows the parametric model for Case Study 2. It is generated in the same way as the model for Case Study 1, its based on four curves, but in this case, the two curves at the end are fixed as flat lines. Therefore it has only eight variables, they are the z-axis coordinates of the eight control points that define the two middle curves. Also in this case the range of motion for the control points is 10 meters upwards and 10 meters downwards, making the total range of 20 meters. Also in this case, the possible outcomes include the flat surface, single and double curvature surfaces either positive or negative, and multiple curvature surfaces as well.

The Variable vector for Case Study 2 is as follows:

$$X = x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8 \quad (10.5)$$

---

<sup>‡</sup>This Case study was published as a part of an article on the International conference on Structures and Architecture 2013 in Guimarães, Portugal (Méndez Echenagucia, Pugnale & Sassone 2013).

where  $x_{i_{th}} = z_{i_{th}}$ .

### 10.3.2 Fitness functions

The first fitness function in this problem is the maximum displacement in the z-axis  $\max(\Delta_z)$ . Figure 10.10 shows the FEM model for Case Study 2. The surface is discretized into flat shell elements, and it is loaded at every node. It is modeled as a continuous thickness concrete shell, constrained with pin joints at its ends. The python FEM software developed by Mario Sassone is also employed in this case study for the calculation of the structure's displacement.

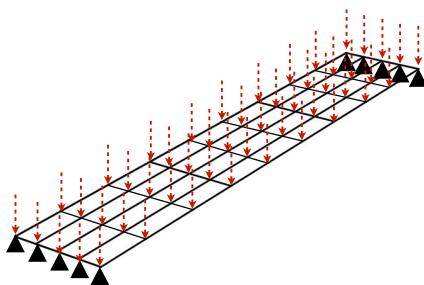


Figure 10.10: Loading and Node Constraint conditions for Case Study 2, the  $24 \times 4$  Bridge.

The second fitness function is the same weight function, simplified as the surface area  $S$ . The problem can be expressed in the following way:

$$\text{Case Study 2} \left\{ \begin{array}{ll} \text{Minimize} & f_1(x) = \max(\Delta_{Z_i}), \\ \text{Minimize} & f_2(x) = S, \\ \text{subject to} & -10 \leq x_i \leq 10. \end{array} \right. \quad (10.6)$$

In this case there is only one known extreme, and that is the flat surface, having the minimum surface area from all possible solutions  $S = 94m$ .

### 10.3.3 Genetic algorithm inputs

For Case Study 2 the GA inputs were slightly different than for case study 1:

### Case Study 2

Population Size ( $N$ )	50
Number of Variables	8
Number of binary digits	8
Variable Domains	$x_i \in [-10, 10]$
Mutation Probability ( $p_m$ )	0.1
End Condition	End after 100 generations

The population size is fixed at 50 individuals and the number of generations set to 100. This means that we have 5.000 calculations. Eight binary digits means that we are dividing our range of motion (20 meters) into 256 steps, leaving us with a domain discretization of 7.8 centimeters. Since we have 8 variables and 256 possible values for each, that means that we have a total of  $256^8$  possible solutions, that's  $1.84 \times 10^{19}$ , a very big number.

#### 10.3.4 Results

In figures 10.11 and 10.12 we can see the Pareto front for Case Study 2 in different levels of detail and at different stages of the search process. Even in this case the lightest shape is the flat shape (Individual A), but the stiffest does not have a theoretical significance. If the longitudinal section of the bridge were an arch with a shape perfectly corresponding to the pressure curve of the load, then the stiffest solution would have this shape and straight transverse section. It would be a barrel vault, or “flat arch”. Since the parametric model we created does not allow such a perfect shape (due to the fixed distance of the generating NURBS curves), the only way to increase stiffness is to add some bending stiffness, through a transverse waved section. The shell, in this case, becomes a kind of ribbed arch, in which ribs increase the arch stiffness. Besides the extreme case of the stiffest shape, this considerations are important for other Pareto front shapes. Ribs, in fact, increase the stiffness and the weight at the same time. Figure 10.12 also shows a few significant solutions generated during the Search process for Case Study 2. The set of shapes depicted in figure 10.12 include some solutions coming from previous search steps (generation 22), instead than from the last only: those shapes do not necessarily represent local minima, but simply steps of the search process. However, they can play a role in interactive design, because they can be chosen as starting points of new search processes, through a redefinition of constraints and of the domain, suggested by the designer evaluations.

The 100<sup>th</sup> generation contains the set of solutions that lie in the Pareto

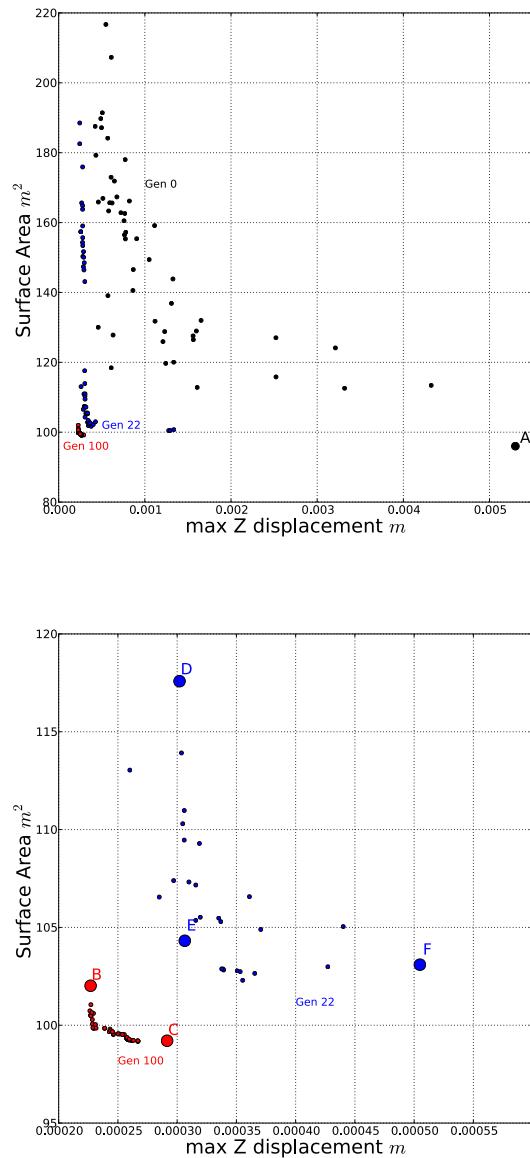


Figure 10.11: Objective space with Pareto fronts for the  $100^{th}$  and  $22^{nd}$  generations and the initial population for Case study 2.

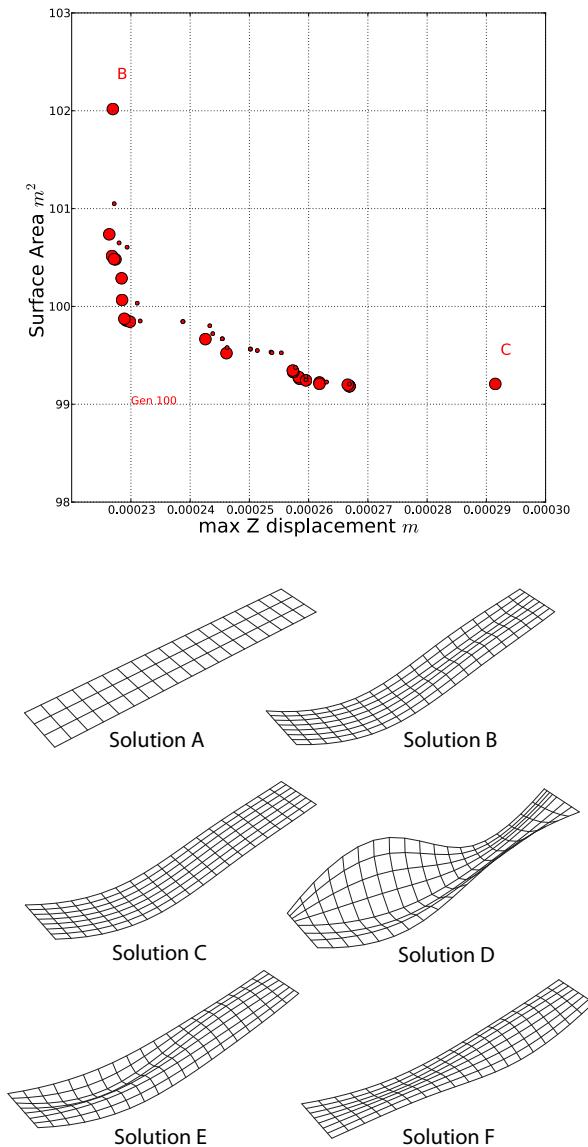


Figure 10.12: Objective space with Pareto fronts for the 100<sub>th</sub> and 22<sub>nd</sub> generations and the initial population for Case study 2 - Individual solutions A,B,C,D,E and F.

front, no other generation contained individuals that dominated any of these. If we take a look at the two Solutions presented in the  $100_{th}$  generation (B and C), we see that they are not very dissimilar from each other. Solution C contains no curvature in the transverse section, making it lighter than solution B. The latter has a curved transverse section, giving it some bending stiffness, in fact increasing its fitness, but making it heavier.

The  $22_{nd}$  generation is an interesting set of individuals to study. We can see that already after 22 generations the improvement from the initial population is significant. This set however, contains a much more diverse group of individuals when compared to the  $100_{th}$  generation. Solution D has a very pronounced cross-section, while E and F have less marked ribs, solution F has a more continuous curvature, while E is wave-formed.

Solution E seems to be in an almost orthogonal position from solutions D and F. In fact, if we look at the overall shape of the Objective space, and we trace lines from solution A (not found by the search process) and the individuals in the Pareto front (generation 100), we note a different pattern that we had not seen before. In section 7.6 we saw that the shape of the Pareto Front can be a clue as to the level of contrast of a given problem. But in this case we see an almost orthogonal pattern, as if one function has very little influence over the other. This denotes a low level of contrast between the two fitness functions.

The difference in fitness for  $f_1$  between solution A and B or C is gigantic, while the difference for  $f_2$  is negligible when compared to the other distances we can see in the objective space. For example, the difference in  $f_1$  between D and C is more than 6 times larger than the difference between A and C. These numbers tell us that we are gaining little weight from A to C, but on the other hand, we are reducing  $\max(\Delta z_i)$  by 96%.

This low contrast is true for this Case Study and in these sections of the objective space, however as we can see in the  $100_{th}$  generation, there is still contrast present in the problem. Other structural fitness functions could be investigated in order to see if the level of contrast changes, or if different measures give the same patterns.

These long orthogonal segments might also be a clue as to why NSGA-II is having problems in finding solution A or solutions between A and the Pareto Front in the  $100_{th}$  generation. If we a perfectly orthogonal pair of fitness functions, the Pareto front would be made up of only 3 solutions. This means that we would have empty segments between them, making it impossible for the MOGA to place solutions other than these 3 in the front. A pair of fitness functions that are almost orthogonal, but not exactly, should be a similar case. We can hypothesize that this is the behavior we have seen

in the last Case Study.

## 10.4 Masonry shells

Compression only structures, such as masonry and stone vaults, have a long and rich tradition, they are a fundamental part of the history of architecture and structural design. Since stone and masonry posses very little resistance to tension, their structural capacity relies a great deal on their geometry. They are designed to have compression only shapes, shapes that do not generate tensile stresses. The most iconic shape in this category is the arch, and more specifically the catenary arch. Shells, being three-dimensional objects can have more complex shapes than the arch. Traditional compression only shells were made in a vast array of shapes such as barrel vaults or sail vaults and they all have precise and well known geometry and proportions that make them compression only shapes.

Free-form compression only forms have also been studied and built in the past, perhaps more famously by Antony Gaudí. Recently the term structural form-finding has been given to the process of generating free-form compression only shapes by means of computation. As we saw in section 4, there are many ways of computing these shapes, and many of them do not involve digital computation. Several Digital form-finding processes exist, for example dynamic relaxation (Day 1965), force density method (Schec 1974), particle-spring systems (Kilian & Ochsendorf 2005) and Thrust Network Analysis (Block 2009). All of these methods tend to be very interactive, in that they need a user defined starting configuration, and many of them can be modified in such a way as to generate different compression only shapes almost in real time. They can be thought of as deterministic and gradient based methods.

The above methods and their applications demonstrate that compression only shapes can be very different from each other and from traditional masonry vaults. However, they are not completely free-form, they represent a sub-group of shapes that satisfy the compression only requirement. The above mentioned methods generate forms that reside within this sub-group, and the designer can interactively generate different shapes, exploring geometry that lies within this sub-group. From this point of view we can think of these form-finding processes as optimization methods rather than search methods. Genetic algorithms, while not being as fast and as efficient as the above mentioned methods in finding compression only forms, they are more powerful from a search point of view. They consider geometry groups or

sub-groups depending on the parametrization given by the user and they can consider these groups from other points of view, apart from the compression only shape. The selection of a GA or a form-finding method should be taken depending on the problem at hand and the user's needs.

In this section we will employ MOGAs to study masonry shell structures.

#### 10.4.1 Structural analysis of masonry vaults

This case study introduces the use of masonry, a material that differs from reinforced concrete in that it does not have a high resistance to tension forces. Therefore, important changes needed to be made in the fitness function that studies its structural capacity and the FEM model that analyses its behavior.

In finite element structural simulations, materials and their mechanical properties are introduced by means of series of values, most important of which is the Elastic or Young's Modulus ( $E$ ). The elastic modulus characterizes the stiffness of the material as the ratio of stress  $\sigma$  over deformation  $\epsilon$ :

$$E = \frac{\sigma}{\epsilon} = \frac{F/A_0}{\Delta L/L} \quad (10.7)$$

where  $F$  is the force applied to the material,  $A_0$  is the cross section area of the object to whom the force is applied,  $L$  is the length of the object and  $\Delta L$  is the variation in length of the object due to deformation.  $E$  is expressed in units of force over units of area, this means it is a pressure unit, it is usually expressed in  $Pa = N/m^2$ ,  $MPa = N/mm^2$  or  $GPa = kN/mm^2$

The elastic modulus for reinforced concrete and masonry depend on their specific materials and configurations, but they can be around  $30\text{ GPa}$  for concrete and around  $10\text{ GPa}$  for masonry. In this way FEM calculations take into account the difference in stiffness of the materials. But the fragility of masonry under tension forces is not taken into consideration. Linear FEM calculations assume the material given is a perfectly elastic and symmetrical material, meaning that it has the same resistance under tension and compression forces. This might be the case for concrete under normal loads, but it is not the case for masonry, therefore we need to somehow take this into account when we study masonry shells.

A brief parametric study of masonry shells is made with the purpose if understanding how structural FEM simulations could describe masonry vaults, and how to take into account the characteristic of masonry described

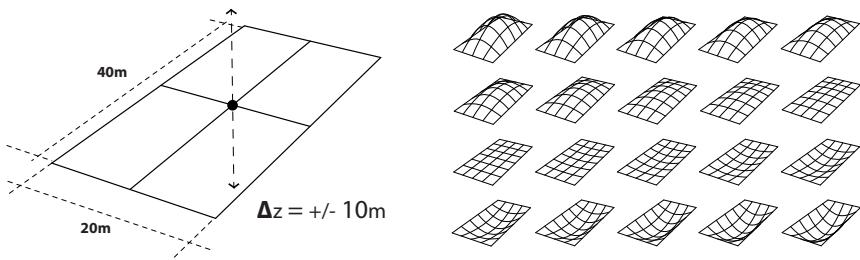


Figure 10.13: Parametric model for study of elastic potential energy and maximum tension of masonry shells - The 20 generated shells.

above. Figure 10.13 shows the parametric model made for this analysis, it contains 20 shells in total. The only variable in this analysis is the z-axis coordinate of the central control point of the surface. We can see that the generated surfaces are all positive double curvature surfaces, when  $z > 0$  we get concave surfaces and when  $z < 0$  we get convex surfaces. The only exception is the flat surface that occurs when  $z = 0$ .

Figure 10.14 shows the FEM model characteristics for this analysis, the shell is supported by pin joints along its edges, and each discrete shell element is loaded with its own weight. The surface is modeled with a fixed and continuous thickness of 5 centimeters. These FEM analysis are made by means of Oasys GSA, a commercial FEM software.

The purpose of studying concave and convex surfaces is to see the difference that the FEM analysis gives to the convex surfaces (that should be mostly in tension and therefore not viable for masonry vaults) and the concave surfaces that should be mostly in compression. The first analysis made is to plot the elastic potential energy  $U_e$  and the maximum tension stress  $\max(\tau^+)$  for all of the surfaces. We can call this a sensitivity analysis of the surface shape for these structural values.

Figure 10.15 shows the result of the analysis. We can see that the elastic potential energy  $U_e$  curve is symmetrical, it has the same values for concave and convex surfaces, for example  $z = -4$  and  $z = 4$ . This is to be expected since the FEM analysis considers the material to be perfectly elastic and symmetrical. The maximum tension  $\max(\tau^+)$  curve on the other hand is not symmetrical. We can see that the convex surfaces have higher tensions than their concave counterparts. Since the concave surfaces in this parametric model are not perfect compression only shapes, there are points in the

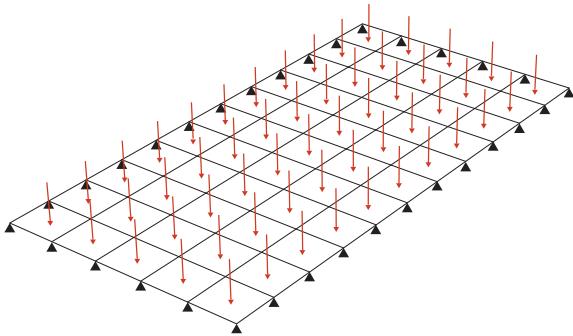


Figure 10.14: Loading and node constraint conditions for the parametric study, the  $40 \times 20$  masonry shell.

surface which are in tension, but their values are lower than the ones found in the convex surface. This gives us the opportunity to create a fitness function that can identify surfaces that have low tension and low overall stresses.

The elastic potential energy describes the general condition of the structure. The higher the elastic potential energy, the higher the structure is being deformed, indicating a lower stiffness in the structure. For this reason, it is often used in GAs as a fitness function, it is a single number indicator of the overall stiffness. But we can see in this case that  $U_e$  is not a good indicator for masonry structures, because it does not distinguish between structures under tension or compression. We need to somehow include the tension into a new fitness function in order to help the GA make this distinction.

But maximum tension on its own is also insufficient for a proper fitness function. For example, if we try to compare perfect compression only shapes by calculating their  $\max(\tau^+)$  values, we would see that they both have  $\max(\tau^+) = 0$ . This means that using only  $\max(\tau^+)$  as a fitness function would not allow us to distinguish between compression only shapes. Compression only shapes however, do have different compression stresses in them, and we can explore between them to find surfaces that minimize also compression stresses.

The following function parametric developed to incorporate both  $U_e$  and  $\max(\tau^+)$ :

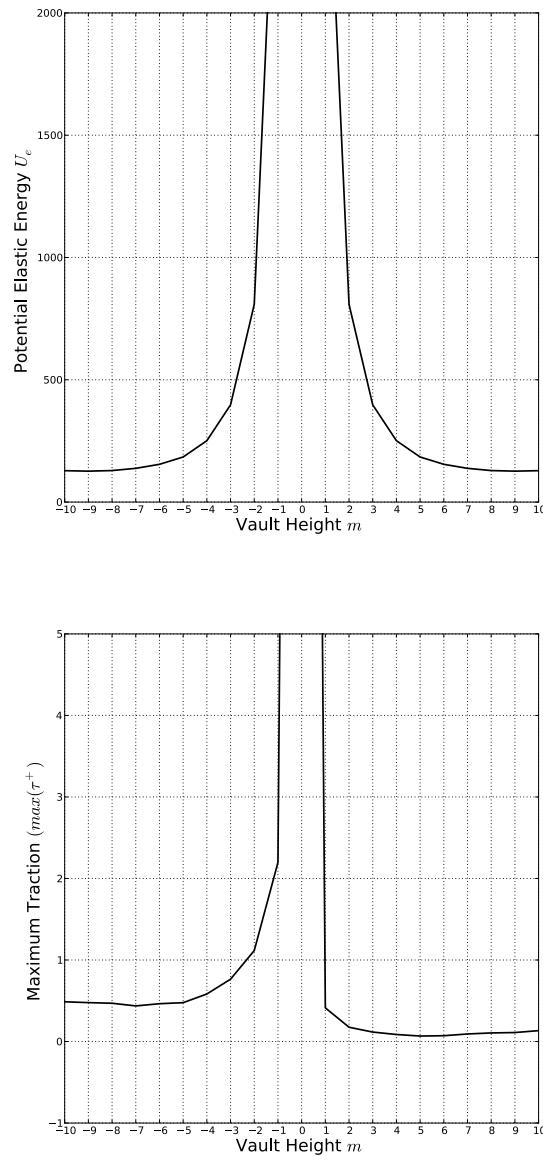


Figure 10.15: Potential elastic energy  $U_e$  for 20 masonry vaults - Maximum tension ( $\max(\tau^+)$ ) for the same vaults.

$$fit = \frac{U_e}{U_{e,0}} + \left( \frac{\max(\tau^+)^2}{\max(\tau_0^+)^2} \cdot w \right) \quad (10.8)$$

where  $U_{e,0}$  is the elastic potential energy for a reference shell,  $\max(\tau_0^+)$  is the maximum tension for that same reference shell and  $w$  is weight a coefficient. The function normalizes both  $U_e$  and  $\max(\tau^+)$  with a reference shell and then the tension is given a higher influence in the function by multiplying it by  $w$ . The reference shell should be a low performing shell, in this case the flat surface parametric selected.

Figure 10.16 shows the result of the parametric study for the fitness function ( $fit$ ) described above, for different values of  $w$ . We can see that when  $w \neq 0$  there is a good distinction between the convex and concave shell structures, but also there is a less noticeable but existing distinction between the different concave shells. This figure suggests that the developed fitness function is capable of studying masonry shells with the linear FEM analysis and with an appropriate distinction between tension and compression stresses.

## 10.5 Case Study 3: Free-form masonry roof

The first case study for masonry vaults is a shell roof for a rectangular building of  $40 \times 20$  meters. This Case study has several similarities with the previous two case studies, it is also a multi-objective search problem with contrasting objectives for structural design and it is also studied using NSGA-II. The main difference, apart from the shell dimensions, is the material, and the fact that in this case the shell is supported along all of its perimeter.

### 10.5.1 Parametric Model

Case study 3 also employs NURBS geometry for the generation of the surfaces. In this case only nine control points were used, and only five of these are variable. Figure 10.17 shows the parametric model setup for case study 3. It shows the location of the five variable control points and their range of movement, they are free to move -10 or +10 meters from the flat configuration shown in figure 10.17. In this parametrization scheme the four corners of the shell are kept fixed, but it allows the edges of the surface to move freely. These edges are completely structurally supported, this means that

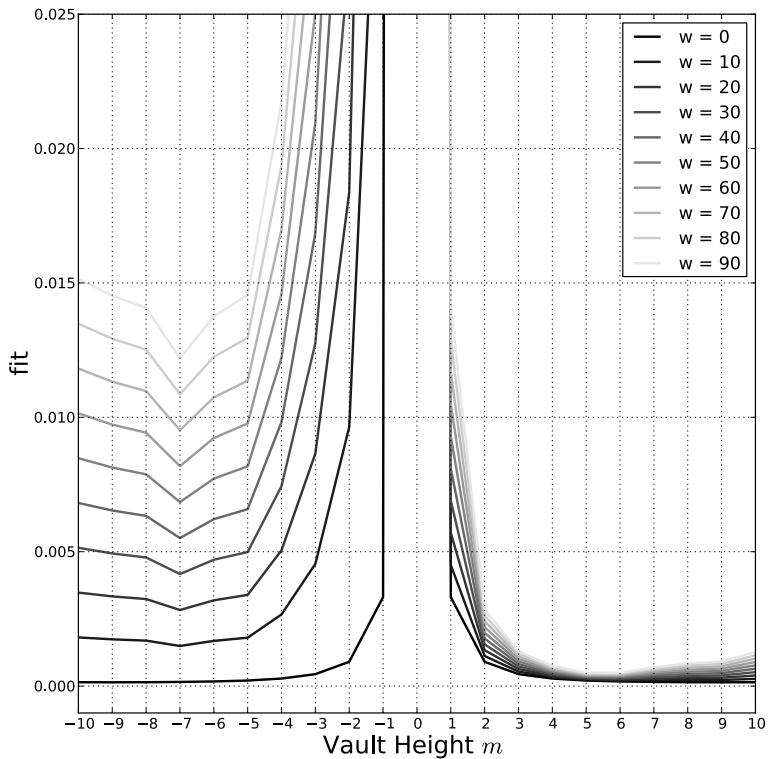


Figure 10.16: Parametric Study of the proposed Fitness function for masonry vaults with variable weight values  $w$ .

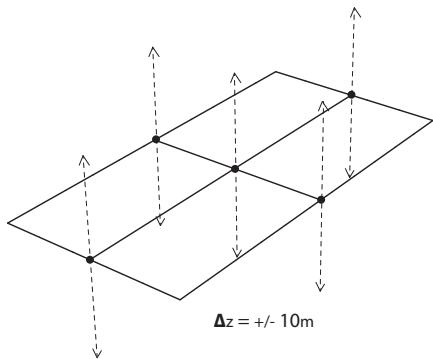


Figure 10.17: Case study 3 parametric model -  $40 \times 20$  masonry shell.

they can move in space, but independently of the shape the edges take, they will always be supported by pin joints.

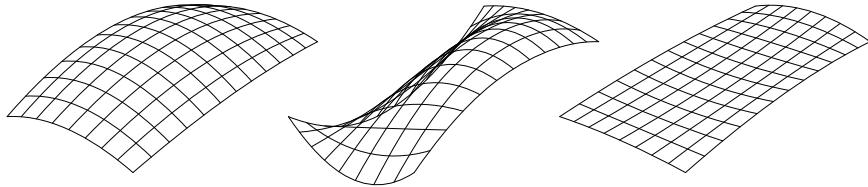


Figure 10.18: Possible individuals with the parametric model developed for case study 3.

In this parametric model all of the usual possible configurations are included: flat, simple and double curvature surfaces, both convex or concave as well as multiple curvature surfaces. Figure 10.18 shows a few possible configurations in this parametric scheme. We can see concave double curvature surfaces and complex multiple curvature surfaces, one with very noticeable curves and another that only slightly diverges from the flat configuration.

### 10.5.2 Fitness functions

This case study, as the two case studies before it, is intended to generate shells that are both light and structurally efficient. The first fitness function describes the masonry shell's structural capacity by means of the function described above in equation 10.8. The reference surface surface used to calculate  $U_{e,0}$  and  $\max(\tau_0^+)$  is the flat surface. The FEM model used for this case study is the same one used for the above mentioned analysis, seen in figure 10.14, where the shells are supported by pin joint in all of the edges, and loaded with their own weight. The thickness of the shells parametric fixed at 5 centimeters.

The shell's weight parametric used as the second fitness function, and as in previous case studies, it parametric simplified as the surface area since thickness and material were fixed. The fitness functions for this case study can be expressed as:

$$\text{Case Study 3} \left\{ \begin{array}{ll} \text{Minimize} & f_1(x) = \frac{U_e}{U_{e,0}} + \left( \frac{\max(\tau_0^+)^2}{\max(\tau_0^+)^2} \cdot w \right), \\ \text{Minimize} & f_2(x) = S, \\ \text{subject to} & -10 \leq x_i \leq 10. \end{array} \right. \quad (10.9)$$

### 10.5.3 Genetic algorithm inputs

NSGA-II parametric used to search for solutions to this problem. It explores 100 generations with 50 individuals in the population. The GA inputs were the following:

#### Case Study 3

Population Size ( $N$ )	50
Number of Variables	5
Number of binary digits	8
Variable Domains	$x_i \in [-10, 10]$
Mutation Probability ( $p_m$ )	0.2
End Condition	End after 100 generations

### 10.5.4 Results

After 100 generations NSGA-II produced the results shown in figure 10.19. The first objective space shows the complete figure with individuals from the first generation up to the last. We can see that the Pareto front has

a very orthogonal shape in this space, with solutions that are quite good at one function and not good at the other, as well as some solutions that are very good at both. This shows a low level of contrast between the two functions at this scale. The second objective space shows a detailed view of the solutions at the intersection of the orthogonal functions, solutions that are good at both functions.

We can also see some of shapes of the solutions in the Pareto front. With the exception of solution A, They are mostly concave and very symmetrical surfaces with different levels of height. Interestingly they all have arched edges. Solution C has a very low surface area while achieving a good amount of stiffness with its shallow concave shape. Solution A is a member of the Pareto front but mostly because of its very small area, its almost a flat surface and does not have a stiff shape for masonry construction. The stiffness difference between solutions C and D is not very big, but their difference in weight is more noticeable.

The stiffest solution in the entire set is solution B, it is the highest surface in the front, but not the heaviest in the set. We can expect the stiffest solution is not going to be the lightest, but the opposite is not true. Higher surface area does not mean a better shape is achieved, and also, since the surfaces are loaded by their own weight, the higher the weight the more load and perhaps a lower structural performance.

## 10.6 Case Study 4: Free-form masonry roof with variable thickness

The previous case study found optimal shapes for both stiffness and weight while having a fixed surface thickness. Case study 4 includes surface thickness as a problem variable, meaning that the GA can now search for stiffer solutions using the thickness, while also evaluating shell weight. Apart from this new variable, all of the other problem characteristics remain the same as in case study 3.

### 10.6.1 Parametric Model

Figure 10.20 shows the parametric model for case study 4. The variable control points make up the first five variables, and the shell's thickness is the sixth one. The shell's thickness can vary from a minimum of 5 centimeters to a maximum of 100 centimeters. The four corner control points of the

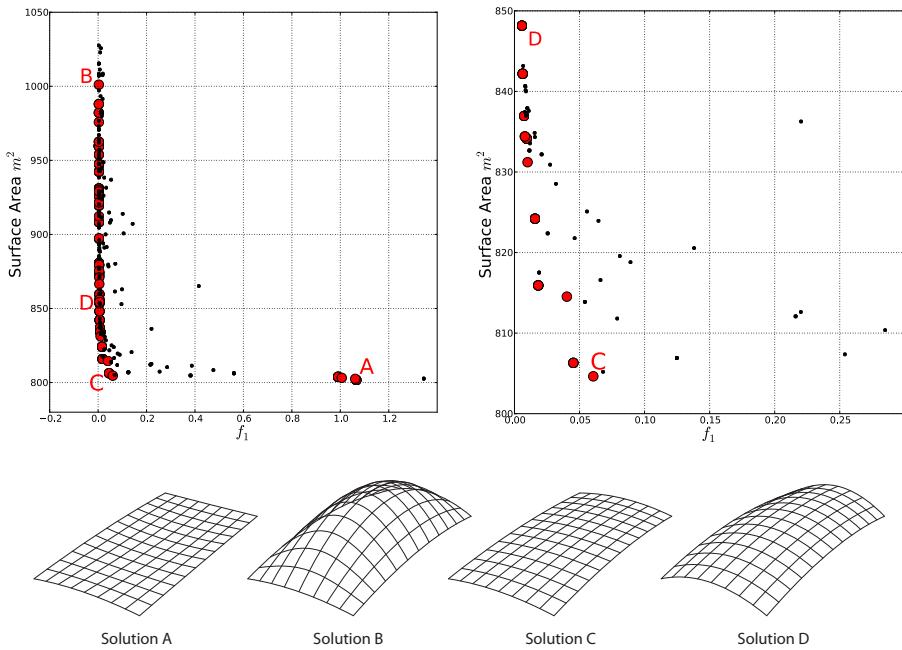


Figure 10.19: Objective space with Pareto fronts for all generations for case study 3 at different levels of detail - Individual solutions A,B,C and D.

NURBS surface remain fixed as was the case in the previous study. We can sum up the variables in this case study in the following variable vector  $X$ :

$$X = x_1, x_2, x_3, x_4, x_5, x_6 \quad (10.10)$$

where  $x_{1-5} = z_{1-5}$  and  $x_6 = \text{thickness}$ .

### 10.6.2 Fitness functions

The first fitness function in this case study is related to the shell's structural performance. The structural capacity of the shells are measured by the function explained above in equation 10.8. The reference surface for the calculation of  $U_{e,0}$  and  $\max(\tau_0^+)$  is the flat shell with the minimum thickness value (thickness = 5cm).

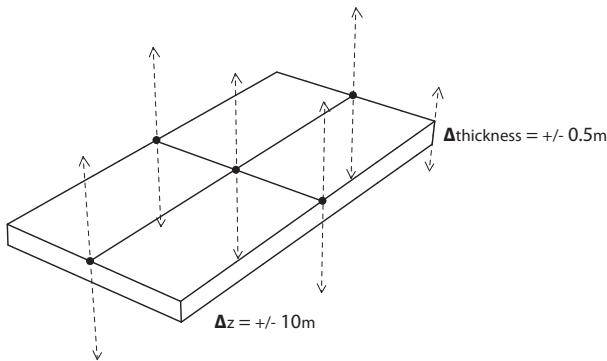


Figure 10.20: Case Study 4 parametric Model -  $40 \times 20$  masonry shell with variable thickness.

The second fitness function measures the weight of the surfaces, but since in this case the thickness is variable, the usual simplification of surface is not viable, in this case the shell's volume is used. The fitness functions for this case study can be expressed as:

$$\text{Case Study 4} \left\{ \begin{array}{l} \text{Minimize } f_1(x) = \frac{U_e}{U_{e,0}} + \left( \frac{\max(\tau^+)^2}{\max(\tau_0^+)^2} \cdot w \right), \\ \text{Minimize } f_2(x) = V, \\ \text{subject to } -10 \leq x_i \leq 10. \end{array} \right. \quad (10.11)$$

where  $V$  is the shell volume.

### 10.6.3 Genetic algorithm inputs

NSGA-II is employed for this case study as well. It explores 180 generation with a population size of 50 individuals. The GA inputs for this case study are detailed bellow:

#### Case Study 4

Population Size ( $N$ )	50
Number of Variables	5
Number of binary digits	8
Variable Domains	$x_{1-5} \in [-10, 10]$
Mutation Probability ( $p_m$ )	0.2
End Condition	End after 180 generations

#### 10.6.4 Results

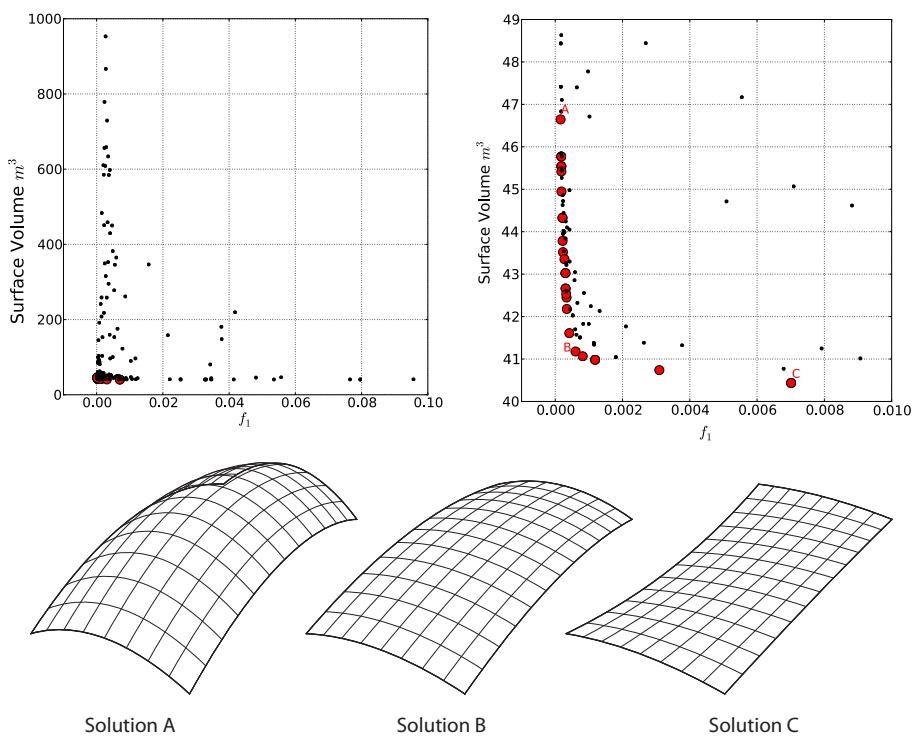


Figure 10.21: Objective space with Pareto fronts for all generations for case study 4 at different levels of detail - Individual solutions A,B and C.

Figure 10.21 shows the objective space with two levels of detail for case study 4. It shows an orthogonal configuration of this case study as well. The vast majority of the individuals in the Pareto front are quite far from the limits of the objective space, meaning that they are optimal when compared to the limits. The detailed view of the Pareto front shows very little contrast between the two functions. The best performing solution from a structural point of view (solution A) is only 15% heavier than the lightest one in the front (solution C).

Solutions A and B are very similar double curvature concave shells with different heights. Solution C is an almost flat surface, but it is a hypar surface. It has concave cross-sections and convex longitudinal sections.

Since they share their fitness functions and materials, results for this study can be compared to those in case study 3. We can see that the Pareto individuals for the structural function reside in a much narrower range, they have better fitness values. This can be explained by the fact that this case study ran for 80 generation longer than the previous one. If we compare the two cases from a structural point of view, we find that solution A from case 4 has a better performance than the best one from case 3 (solution 3-B). In fact solution 4-A has a performance value that is twice as good as 3-B, and it is also lighter.

An important result that we can point out is the fact that all shells in the Pareto front are 5 centimeters thick. In fact, after just a few generations the GA considered only solutions with very low thicknesses, excluding solutions with high thicknesses. After generation 80, the entire population is made up of solutions with 5 centimeters of thickness.