10_evolutive structures

topology optimization

"I don't want to undress architecture. I want to enrich it and add layers to it. Basically like in a Gothic cathedral, where the ornament and the structure form an alliance".

Cecil Balmond

Contemporary architecture practices have enabled a clear distinction between the role of the architect and the role of the engineer. In this paradigm, the architect is concerned with the buildings shape and functionality while the engineer is concerned with mathematical "firmitas." Technological developments and contemporaneous architectonic research have attempted to reintegrate the two fields by nesting shape and statics, as well as creativity and structural calculation.

The study of *optimization* integrates design and structural decisions, to create optimal solutions within set parameters. These solutions are often structural: and aim to reduce the amount material, span further, and leverage structural latencies. Optimization can also be applied to building program, function and form.

The two leading domains of optimization are: shape optimization and topological optimization. Both forms of optimization attempt to reach an optimal solution with respect to a set of parameters that define a fitness function.

Both approaches share the same procedural logic:

- An initial geometry is defined;
- Boundary conditions are defined that control the optimization;
- Definition of one or more fitness functions.

Even though the fundamental process is the same for both of these techniques, there are conceptual and algorithmic differences.

10.1 Shape Optimization

Shape optimization is the process of calculating a geometry that minimizes the assigned fitness function within the boundary conditions.
The elements of a geometric optimization are:

- Fitness function, that is commonly defined as a minimum research problem;
- Variable definition;
- Supports setting;
- Definition of a domain for the elaboration of a solution.

The first three elements are decisions that set the desired outcome path for the optimization problem. The optimization process can be summarized in 5 steps as illustrated below. Once the analysis is complete and the results have been evaluated, the algorithm seeks for an optimal solution. The result of this process will be a modified initial geometry that maintains the initial topology.

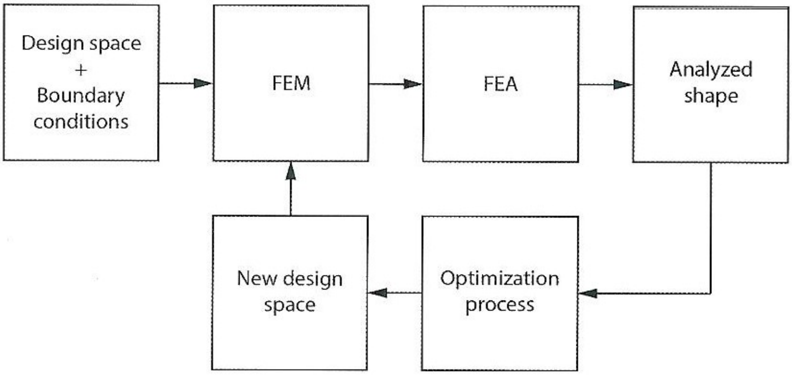


FIGURE 10.1
Shape optimization flowchart.

For example, this approach can be used to return an optimized cross section within the define domain of a cantilever.

The cantilever is composed by one fixed extreme named as A and one free extreme named as B. The boundary conditions are:

- Fixed connection at point A, no restraints at B;
- Point load at point B;
- Defined material properties and a list of possible cross sections: UNI5397-78 and UNI5398-78 standards.

The algorithm seeks to minimize the displacement of point (B). The calculations are performed by a genetic solver that analyzes all the possible solutions and discards solutions that do not efficiently meet the demands of the fitness function. Solvers will be further discussed in section 10.6.

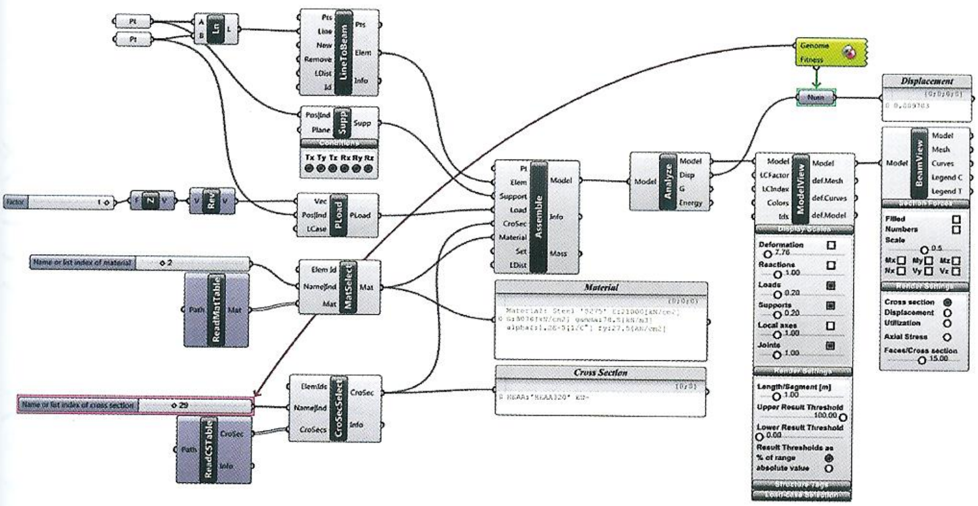


FIGURE 10.2

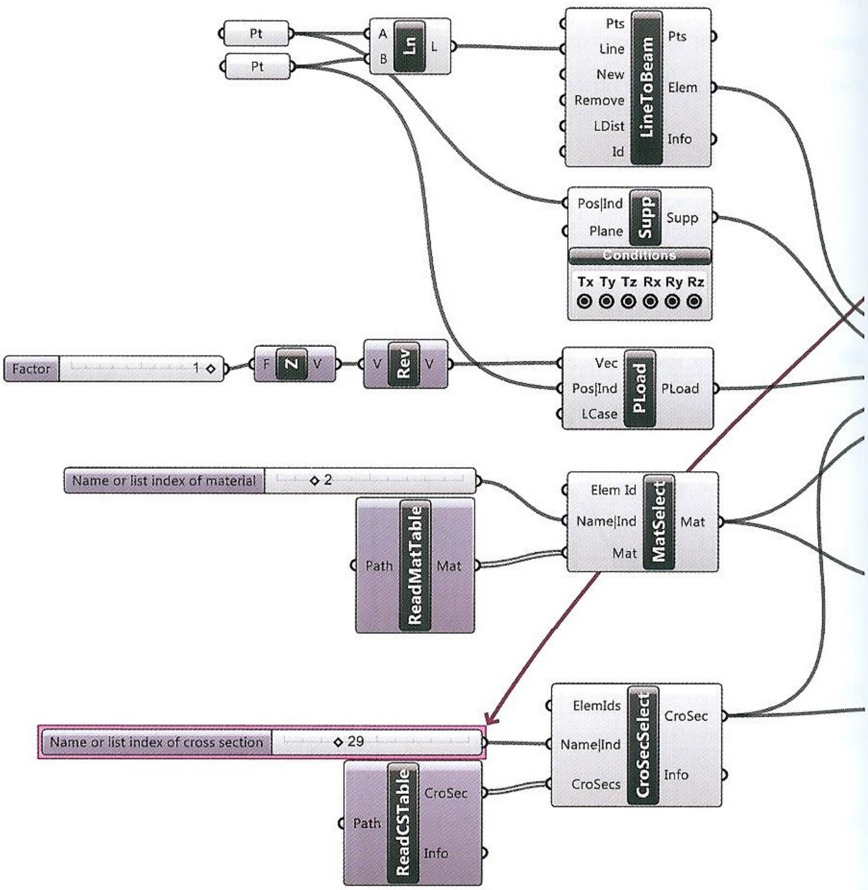
Algorithm definition of the shape optimization.

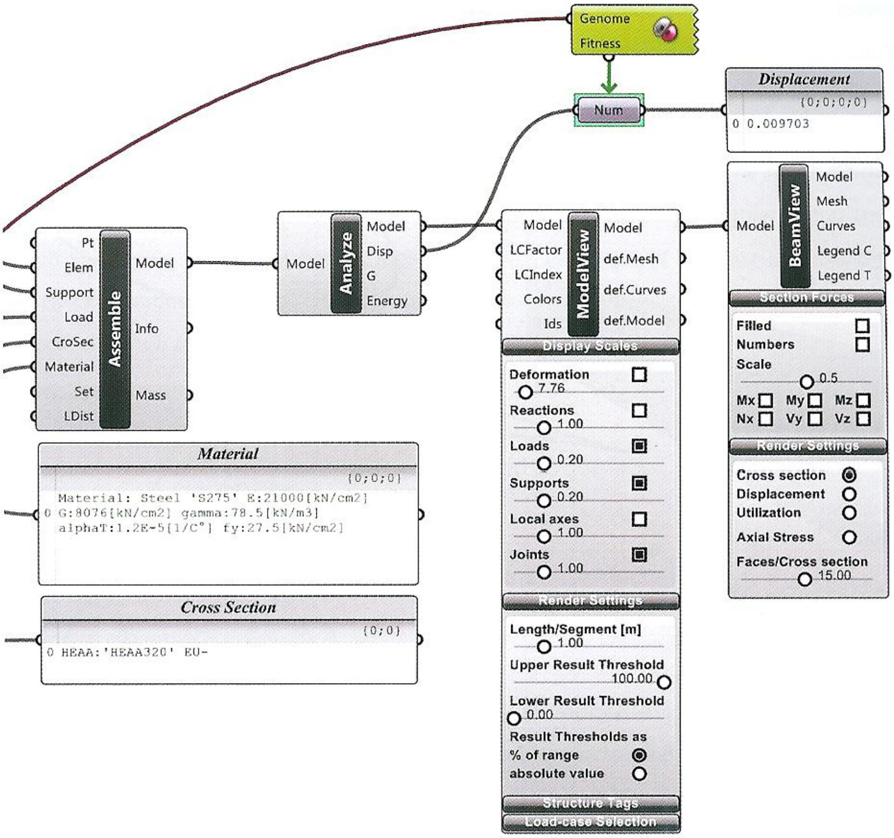
The definition illustrated in figure 10.2 can be read as a sequence of four steps:

1. Modeling of the static scheme and the definition of the boundary conditions;
2. Elaboration of the finite element model;
3. Model analysis;
4. Visualization and analysis of the results.

After the analysis is completed and the results are evaluated by the designer, the optimization process can start defining an additional step.

The first step in the structural optimization definition is constructed as follows. Two points set from Rhino are used to define a line in Grasshopper. The line defines the longitudinal axis of the beam on which the analysis is performed The end points (A) and (B) are respectively used to define the location of fixed supports and a concentrated load described by a vector in the Unit Z direction with a magnitude set by a slider. Finally, the material properties and the list of cross sections to analyze are set.





The material and cross section properties are set using the *Karamba*[28] material and cross section libraries respectively. In this instance S275 steel and IPE and HE sections are selected. The list of cross sections represents the population of solutions in which the genetic solver will look for the best one. Each cross section is analyzed under the described boundary conditions using finite element analysis; the section that allows the minimum displacement at point (B) is the optimal solution from the list. Once the process is completed it is possible to visualize all the solutions and the gradual selection of the optimum cross section.

NOTE 28
Karamba is a Grasshopper plug-in used to analyze structures such as: spatial trusses, frames or shell structures. It is developed by Clemens Presinger in cooperation with Bollinger-Grohmann-Schneider ZT GmbH Vienna. For further information visit www.karamba3d.com.

It is necessary to note that the solver selects the optimum section from a short list of possible solutions in a very well known problem; experience demonstrates that, dealing with complex problems, often the solution provided by the solver is not "the best one" but is closer to it.

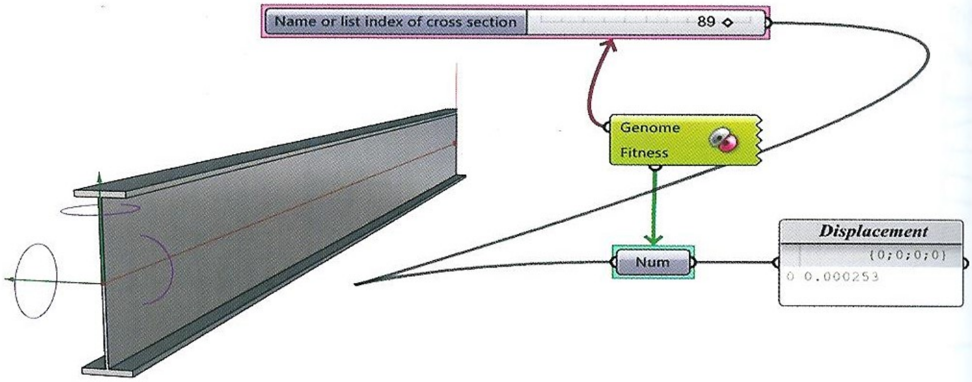


FIGURE 10.3
Optimized cross section.

The optimization calculations are performed by Galapagos (see 10.6), a genetic solver within Grasshopper. Research in the field of computational generative algorithms started in the 1960's with the publication "On the organization of the intellect"[29] published by Lawrence G. Fogel. The popularity of evolutionary calculation in the IT world came with the book "The Blind Watchmaker"[30] written by Richard Dawkins and the following applications have been made possible thanks to the improvement in computers.

NOTE 29
L.G. Fogel, "On the Organization of Intellect", PhD Thesis, UCLA, 1964.
NOTE 30
R. Dawkins, T*he Blind Watchmaker* (New York: W. W. Norton & Company,1986).

10.2 Topology

Topology is the study of the relationship between geometric parts undergoing deformation. Unlike plane geometry, topological analysis does not require metrical or angular measurements; instead the study of topology is based on the comparison of figures. The figure below shows a comparison between an equilateral triangle defined metrically and angularly, and the topology graph illustrating the "Konigsberg bridge problem"[31] as theorized by Euler in 1736. Even though the two illustrations are different the two graphs describe the same problem, since the position of objects and their dimensions do not affect the system topology.

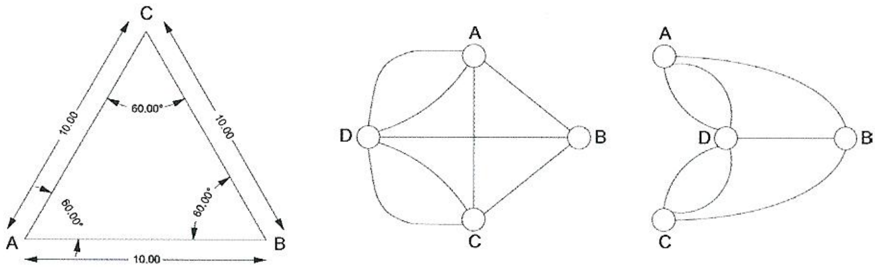


FIGURE 10.4
Comparison between the classical geometric definition of a shape and the topological approach to a problem using graphs.

The main points of topology are:

- The principle of continuity due to strain, even when losing its metrical and projective characteristics, there will be a biunique correspondence among the points of the original figure and those of the transformed one.
- The principle of homomorphism states that strains, causing penetrations or the overlapping of material are not admissible after the deformation.

Topology principles can be visualized by the transformation of a square. A homomorphism map can be created from the square to the circle; thus it is possible to carry out a transformation leading from one figure to the other without overlapping or cuts.
Homeomorphism between two figures is regulated by a parameter called *genus*. Genus is a topological

NOTE 31
L. Euler, "Solution problematis ad geometriam situs pertinentis", Commentarii academiae scientiarum Petropolitanae 8, 1741, pp. 128-140.

invariant that can be mathematically demonstrated which asserts that two homomophic figures must have the same genus value (g). The (g) value of a figure is equivalent to the number of holes within the geometry or, more exactly, is the greater number of non-intersetting simple closed curves drawable on a surface without splitting the surface itself in two separated parts. From a morphological perspective the presence of the same number of holes in two figures, or a total absence of them, leads to homomorphism. Thus, when the sphere is defined by a genus (g) with a value of 0 and when the torus is defined by a genus (g) with a value of 1, they are not homomorphic and it is impossible for one to become the other and vice versa. So the homomorphism settles the transformation possibilities among figures.

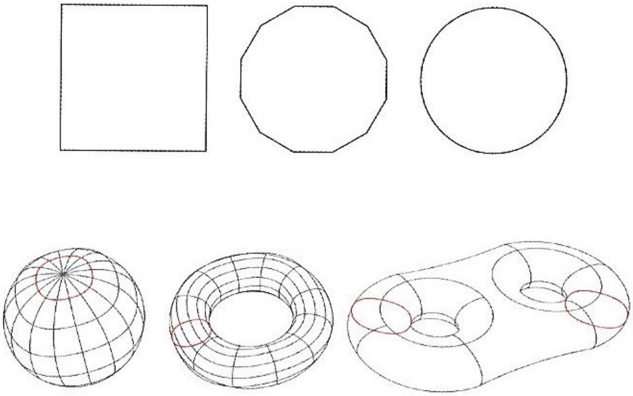


FIGURE 10.5
Representation of continuity and homomorphism principles.

10.3 Topology optimization

Mathematical topology principles are used to study the topological optimization of form. Research in the field of structural optimization has brought teams of engineers, mathematicians and later on, architects, to investigate topological optimization to more efficiently use material.
Topological optimization began conceptually in the 19th century, however, the results were not validated till the end of the 1980's with the improvement of digital calculation. The first functioning algorithm was presented in 1992 by Xie and Steven under the name of Evolutionary Structural Optimization (ESO).

The main characteristics of the algorithm are:

- Possibility to change the structural topology exceeding shape optimization;
- Evolution-based procedure for the development of an optimization process.

The process follows four main steps:

1. Definition of the initial data: geometry, supports, loads and constitutive material;
2. Definition of optimization parameters;
3. Finite Element Analysis (FEA);
4. Structural optimization.

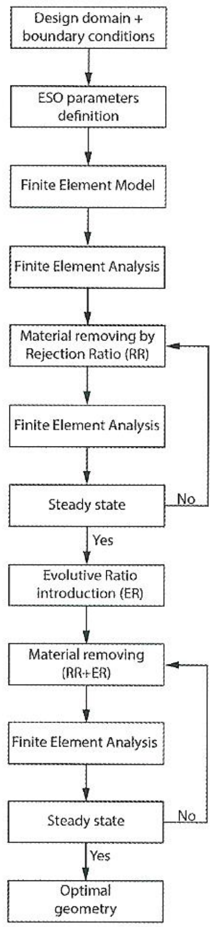


FIGURE 10.6

Evolutionary structural optimization flowchart.

The first step in an ESO process is to set the input data, then allow the ESO to run subtracting material. The initial geometry definition requires a larger design space compared to the final result of the optimization process. Contextually, the position and type of the supports, the loads and the material properties will be set so that the strain in each part of the structure can be calculated by the Finite Element Analysis.

The second step in the ESO process determines the parameters for the optimization process and the discretization of the initial design space into finite elements (FE). Once the structural FE analysis is concluded the distribution of the strains can be visualized within the cross sectional area clearly displaying areas of high and low stress.

These two steps are applied to a simple bi-dimensional example of a beam supported at each end and loaded with a concentrated load applied to the lower edge of the beam[32].

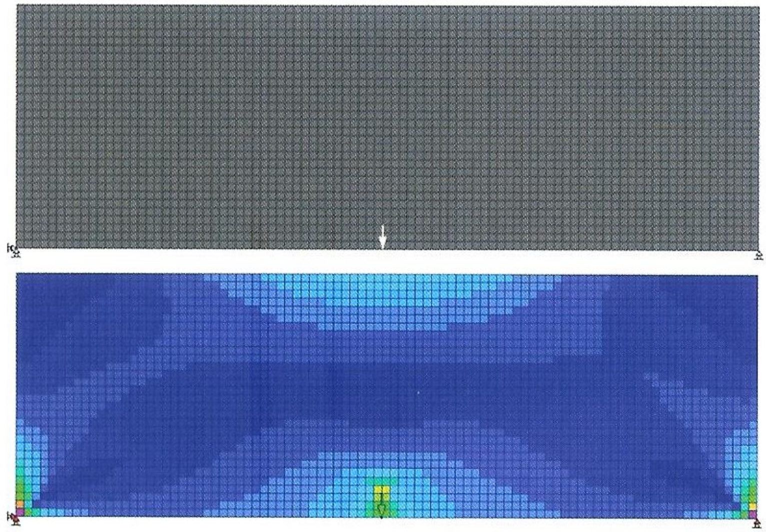


FIGURE 10.7

Finite element model and finite element analysis.

Once the calculation is complete it is then possible to discard inefficient material using Rejection Criterion (RC) defined by the Von Mises stresses, and the Rejection Ratio (RR).

NOTE 32

Example realized using BESO2D, software developed by Prof. Mike Xie and introduced in the book *Evolutionary Topology Optimization of Continuum Structures: Methods and Applications*, Huang, X. and Xie, Y.M. Chichester, England, John Wiley & Sons, Ltd, 2010.

The RR is a percentage value used by the algorithm to identify FE discretized cells were material must be rejected. If the RR is set up to 50%, we must reject all of the finite elements where the RC reaches values lower than the 50% of the maximum admissible value.

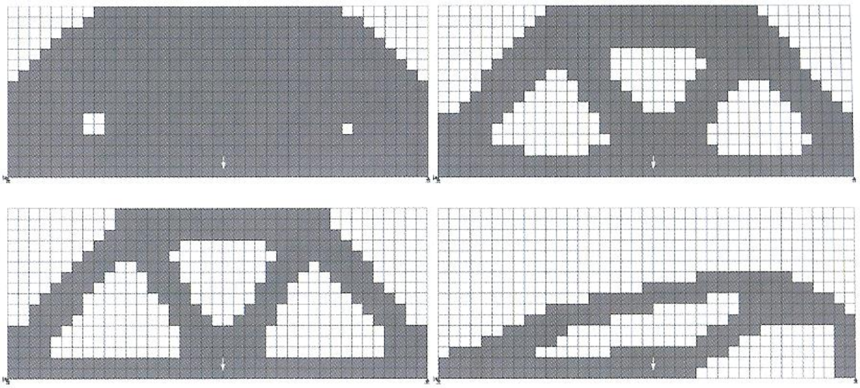


FIGURE 10.8
Different rejection ratios applied on a MBB beam optimization.

The correct calibration of the RR value is essential to the accuracy of the optimization. The above image shows the same optimization example, with different RR values applied.
By examining the results, high RR values (i.e 90%) lead to an extremely limited subtraction of material, Medium RR values (i.e. 60% and 50%) show results that are compatible with the expectations provided by experience, and Low RR values (i.e. 30%) lead to an excessive rejection of material; in this case a non valid static solution is produced.
Once the first step of material removal is concluded and a stable configuration for the beam is achieved the beam is further optimized using an Evolutionary Rate (ER), which is added to the RR.
The removal of inefficient material has changed the distribution of the strains inside the structure consequently the stress concentrations inside the material which survived the evolution process have increased. It is desirable to let every element work at the maximum of its mechanical potential by including a parameter to limit the possible tension inside a determined maximum value.

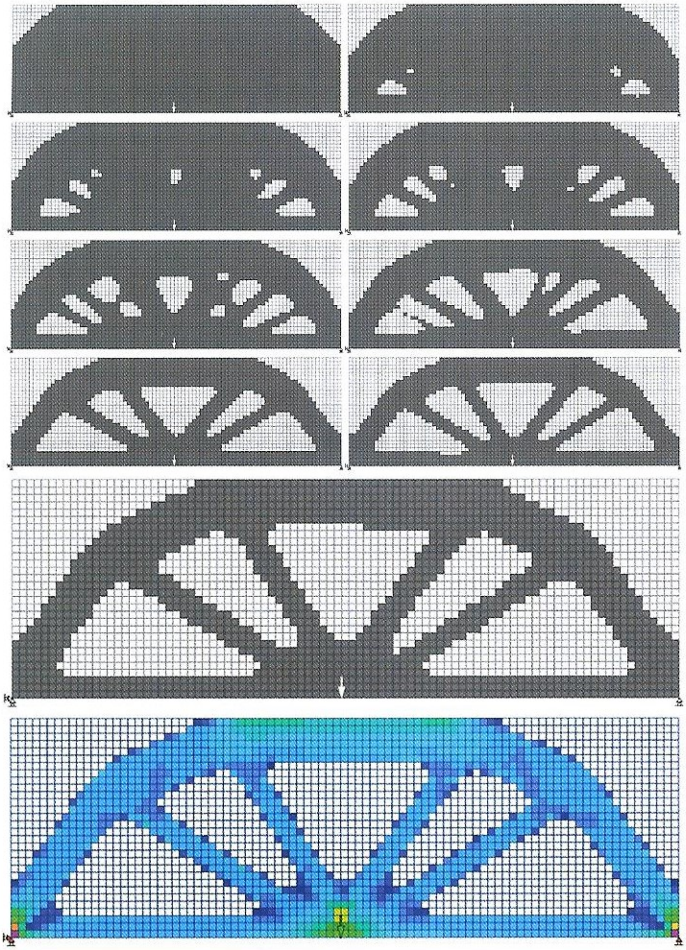


FIGURE 10.9
MBB beam optimization and Finite element analysis of the optimized beam.

Extended ESO (XESO), suggested by Sasaki[33], is based on the concept that organisms underwent environmental adaptations by rejecting the unnecessary and strengthening the necessary. The *Banyan Tree* epitomizes this approach. The tree has a set of solid aerial roots that by extending to the ground support the main trunk. This structure is only made of what is functionally necessary without wasting material.

NOTE 33
C. Cui, H. Ohmori and M. Sasaki, "Computational morphogenesis of 3d structures by extended eso method," *Journal of the international assocuation for shell and spatial structures* IASS, 4 (1), (2003): 51–61.

Starting from this example Sasaki expounded that ESO, in its original configuration, is limited by two factors: the mono-directionality of the evolutionary process; which allows the discarding of only inefficient material, and the lack of control of the optimization.

To fix the limits, two innovations were introduced:

- Control strategy of optimization using *contour lines*;
- Bi-directional evolution.

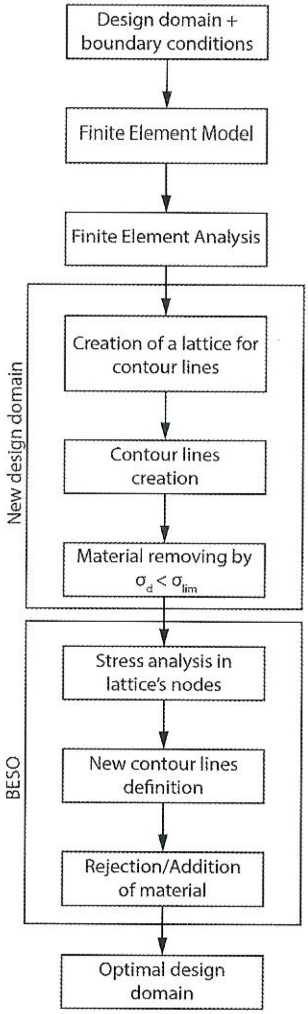


FIGURE 10.10

Extended evolutive structural optimization flowchart.

The XESO traces a set of lines describing the stress status of the object indicating portions of the surfaces with:

a) Lower stresses compared to the limit value;
b) The same stress compared to the limit value;
c) Higher stresses compared to the limit value.

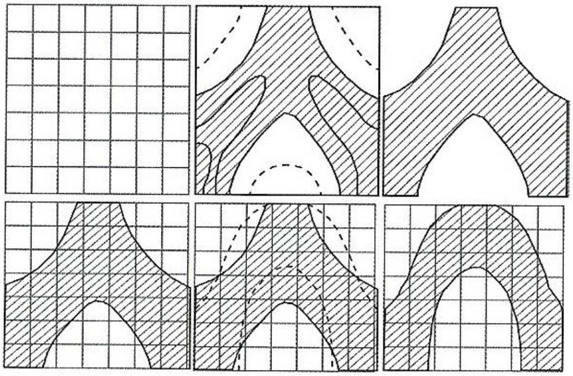


FIGURE 10.11
Contour lines application and material removal scheme.

The XESO approach speeds up the optimization initial steps because the material that exists satisfies the Von Mises condition $\sigma_d<\sigma_{lim}$; material that does not satisfy this condition is discarded. Once the initially optimized design space is set; material is added and removed material and subsequently the stresses are calculated at the overlap between the contour lines and the initial grid. Then, new contour lines are traced to depict where to remove or add material. Contour lines inside the material are equated to material to be removed, while contour lines outside of the material are equated to material to be added.

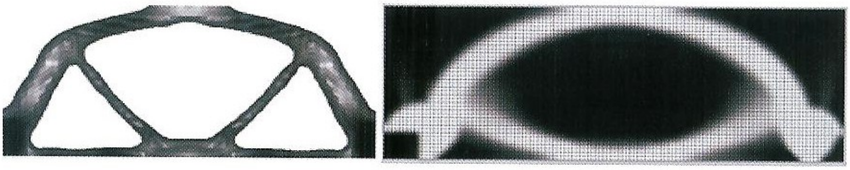


FIGURE 10.12
Comparison between MBB-beam optimized with ESO method and XESO method. Cui, C., Ohmori, H., & Sasaki, M. (2003). Computational morphogenesis of 3d structures by extended eso method. Journal of the international association for shell and spatial structures IASS, 4 (1), 51-61.

10.4 Works

10.4.1 Akutagawa office building

The Akutagawa office building was designed and built by Hiroshi Ohmori in 2005 using the XESO method. The building is within a larger commercial area close to Takatsaki station on a parcel square-shaped 6 x 10m. The XESO method was applied to the northern, southern and western facades, while the eastern facade and the attics were excluded from the optimization process. In the finite element model the specific weight of the structural elements in vertical direction, and the horizontal forces simulating a telluric action were applied. The topology of the three facades evolved as material was removed from the areas undergoing less stress and added in areas of higher stress.

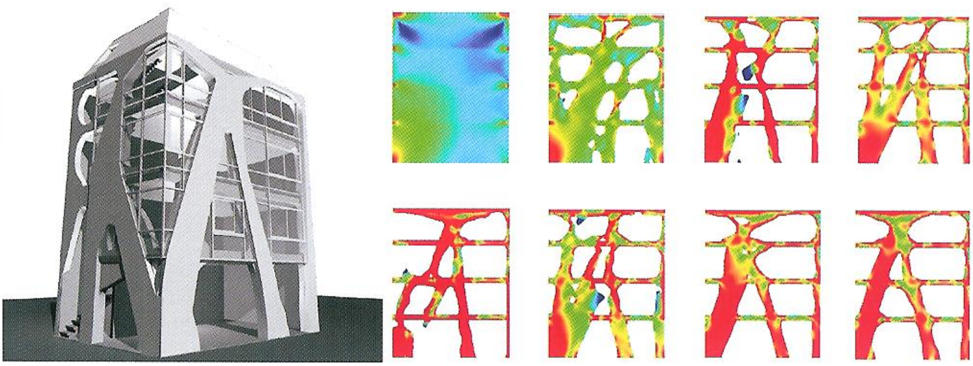


FIGURE 10.13
Akutagawa office building - model and finite element analysis of optimization steps. Ohmori, H. "Computational morphogenesis, Its current state and possibilities for the future" - IASS-IACM 2008 Cornell University, Ithaca, NY, USA 28-31 May 2008.

10.4.2 TAV station in Florence

Arata Isozaki and Mutsuro Sasaki TAV station proposal in Florence used the same algorithm that had been applied to the Akuttagawa building. The building: 400m long, 40m wide and 20 m high, houses the stations program within a single superstructure. During the optimization process, the structure was modified from the initial shape of a bridge with simple frames on the extremities to a more organic shape. The XESO process was illustrated by a set of schemes describing the evolution of the structure. During the development phase Sasaki's team was supported by Buro Happold - an international engineering firm located in London - to define the structural sequence. The structure is composed of steel trunks, pre-stressed beams and steel and concrete slabs. Isozaki, Sasaki and Buro

Happold used the same approach for several competitions until they realized the Qatar National Convention Center.

10.4.3 Qatar Education City

The XESO method used in the TAV station competition was later used to design the Qatar National Convention Center in Doha. In this case, the Japanese architect Arata Isozaki adopted the XESO method to design a structure 25 m long, 30 m wide and 20 m high volume with a roof plane. The main loads in the XESO calculations were imposed by the self weight of the structure.

The project is an archetype of maximum efficiency and the minimum use of material. The design process was a synthesis between architects, engineers and machines. Relying on the Florence TAV experience, the designers began with a more refined model to speed up the entire process. This choice allowed a considerable reduction in time-schedules concerning the analysis and optimization phase.

FIGURE 10.14
The Qatar National Convention Center. Image by UNCTAD.

10.4.4 Radiolaria

Equally imperative as the application of XESO optimization method is the necessity to define an evolutionary building technology.

The process of building optimized structures, have stretched the conventions of construction. For instance, to build the Akutagawa Office Building molds were taken from the shipbuilding industry. For the Qatar Education City, Buro Happold used an elaborate system of prefabricated modules that were built in situ nullifying Sasaki and Isazaki's flux structure idea.

Currently the evolutionary building technology 3D printing is being investigated. For example the work of Architect Andrea Morgante was realized by Engineer Enrico Dini using the D-Shape printer. The project was based on a micro-organism, the Radiolaria, and was printed monolithically without formwork directly from the static approved CAD model.

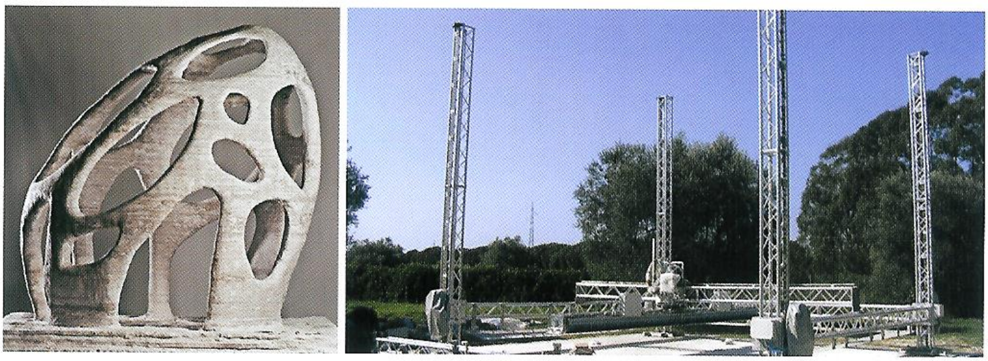


FIGURE 10.15

Radiolaria printed model and picture of the D-Shape3D printer. Image courtesy of D-Shape.

10.5 Examples

In 1904 Anthony Michell[34] economized structure by topological optimizing a beam to support itself and an applied load while minimizing weight and material. The Michell Truss consists of a planar beam designed to support a single load placed on one extremity having, on the opposite extremity, a circular support area.

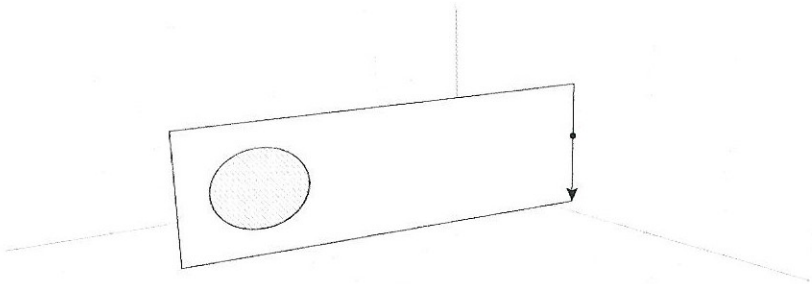


FIGURE 10.16
Michell truss 3D model.

Michell Truss problem can be calculated in Millipede[35] using the components for 2D structural optimization.

The workflow is composed of four steps:

1. Definition of the design space and of the boundary conditions: initial geometry, supports and loads;
2. Creation of the finite element model;
3. Topological optimization;
4. Visualization and analysis of the results.

NOTE 34
A.G.M Michell, "The limits of economy of material in frame-structures," *Philosophical Magazine*, Vol. 8 (47), (1904): 589–597.
NOTE 35
MIllipede is a Grasshopper plug-in developed by Kaijima Sawako and Michalatos Panagiotis that allows FEM analysis, form finding and topology optimization (www.sawapan.eu)

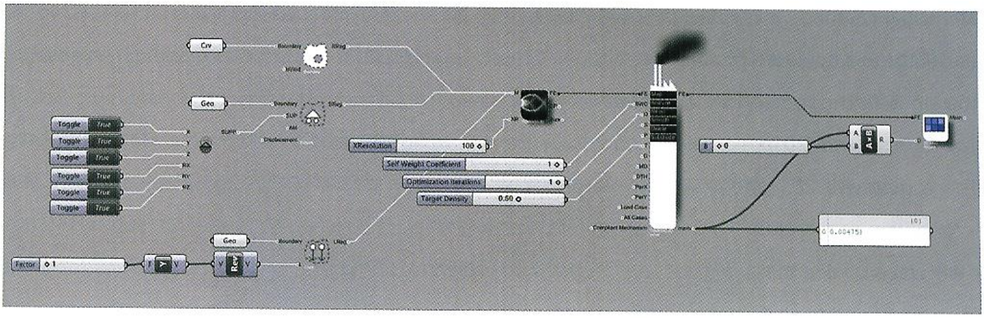
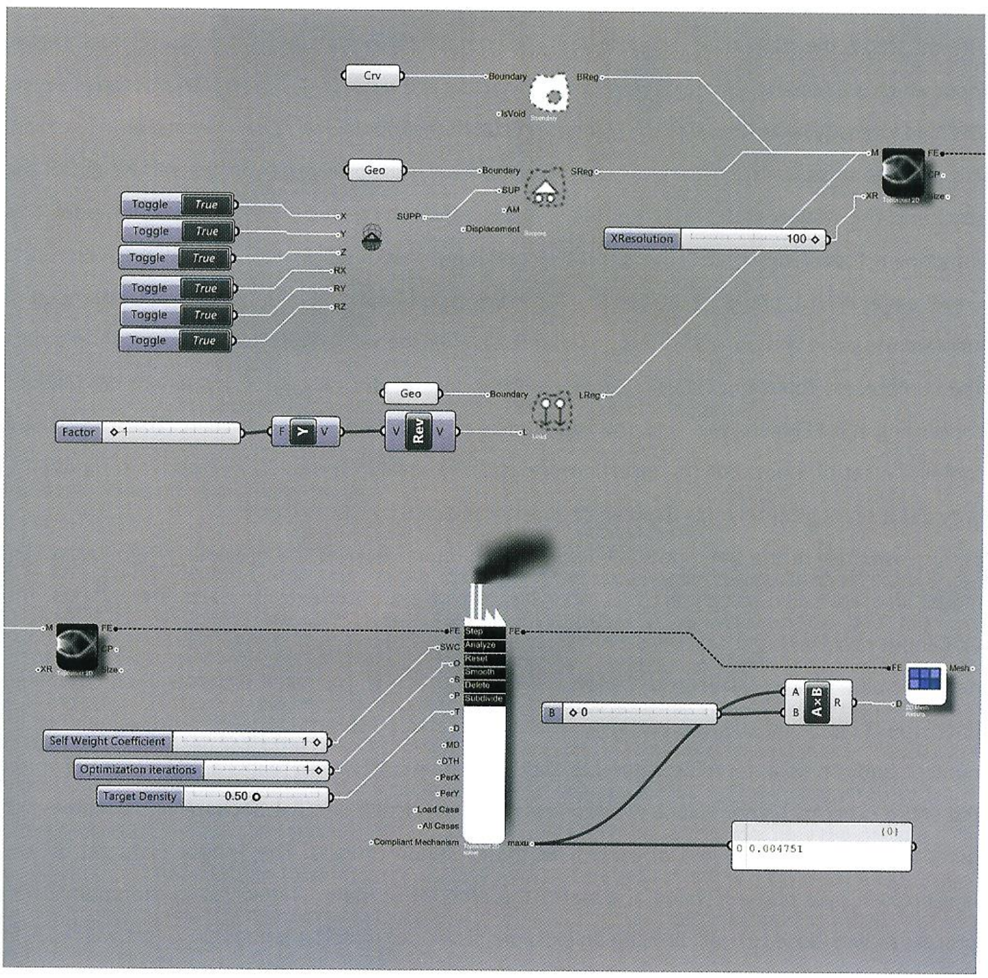


FIGURE 10.17
Algorithmic definition of the Michell truss.



The initial geometry is the design space where the optimal geometric configuration must be found, and in the examples is a rectangle realized in Grasshopper or set from Rhinoceros, which represents the input for the component *2DBoundaryRegion*.
The components *2DSupportRegion* and *Stock Support Type* are used to model the static supports. The component *2DSupportRegion* requires input geometry ideally representing the support and information concerning fixed displacements and rotations. The region that must be supported is a circle placed inside the beam at the center of the median axis of the rectangle. The type of support in this case is fixed so displacements and rotations are fixed in each direction. Restraints can be released or engaged by introducing a *Boolean toggle* to each of the *StockSupportType* component inputs, and toggling True and False respectively. Since this is a bi-dimensional problem it is unnecessary to support the displacements in the z axis and the rotations in the x and y axis.
The last step is the creation of a load region. The component *2DBoundaryLoad* requires a geometric input which is a rectangle placed at the extremity acting in the Unit Y direction with a negative sense. The load intensity has no relationship with the result of the topological optimization. For this reason, it is possible to set a fictitious load intensity of L=-1N/mq or to rely only on the self weight of the structure. The system is assembled by the component which creates the finite element model. This component Topostruct2DModel also has a second input that determines the model resolution. The preset value XR=12, proves to be too low for every type of application, and it is recommended to increase the value with the consideration that the higher the value the longer the evaluation time.
The component *Topostruct2DSolver* represents the core of the plug in and is where calculations concerning the FEM analysis and the topological optimization are performed. This component contains a set of inputs for the insertion of data, and unlike classical Grasshopper components, provides a set of internal tools that can be directly selected by the designer.
In this case, four of the ten inputs are used: the FE-input, receives the previously elaborated FEM model. It is advisable to connect this input last because it automatically starts the analysis. The SWC-input deals with the self weight of the structure and requires an integer numeric value; if the numeric value is 0 the structures weight will be none, if it is 1 the self weight will be computed once. The O-input sets the number of iterations calculated by solver; in this case the value must be numeric and can be entered by a number slider, practice suggests to follow a O=1 value to control the entire process, higher values will require a longer calculation time because of the number of iterations and may cause loss of control on the evolution process. The T-input or target density is expressed by a numeric value and states the amount of material involved in the optimization process (values close to 0 will provide an extreme discard of material, values close to 1 will result in a inadequate decrease of material).

Once the FE-input has been connected, the FE analysis will automatically start. The direction of the topological optimization process is controlled by the STEP tool embedded in the solver, which starts a single iteration. Further iterations will produce an optimal configuration.

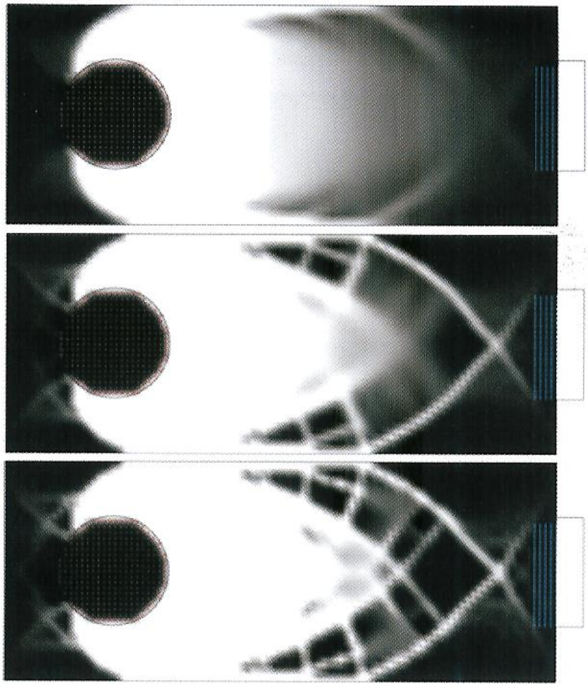


FIGURE 10.18
Iterations of the optimization process.

It is possible to visualize real time results by linking the solver with a visualization component *2DMeshResult* to graphically illustrate information about the stresses inside the member (VonMisesStresses), the flux of the main stresses (Principal tension) and displacements (Deflection). The *Topostruct2Dsolver* provides two outputs: the FE-output, used to visualize the results and the maxu-output that retrieves the maximum displacement value. The last one can be used to visualize the physical deformation of the object. Generally the maxu-value is so low, compared with the dimensions of the analyzed object, that requires an increasing factor to show how the deformation works. In order to boost the magnitude of this value is sufficient to multiply it by an integer number through a *Multiplication* component. Note that this enhancement affects the visualization of the deformed object and not the real displacement.

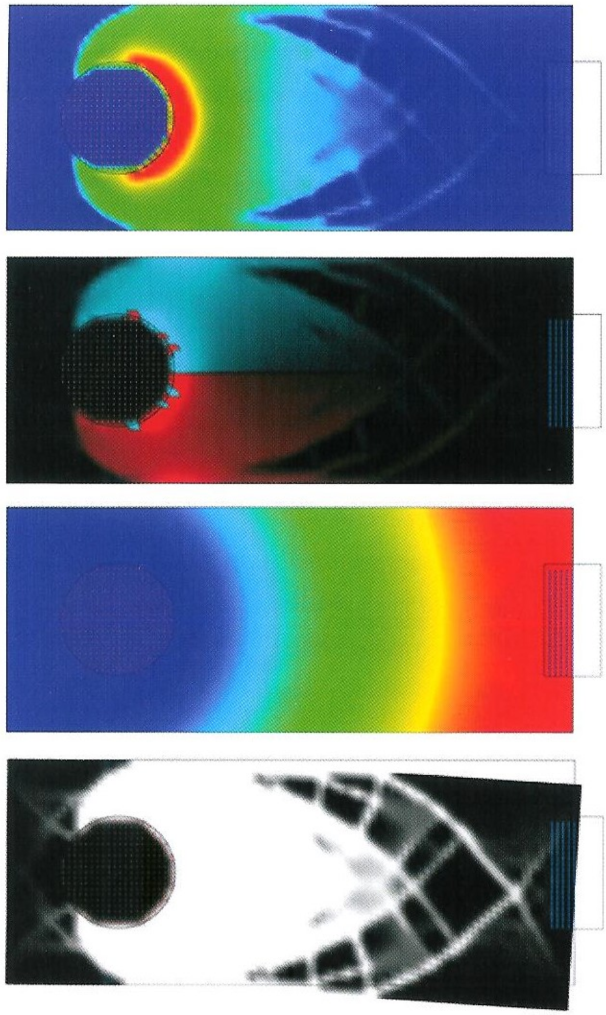


FIGURE 10.19
Michell truss's Von mises stress, principal strain and deflections.

Another common example of bi-dimensional topology optimization is the MBB-beam under self weight.

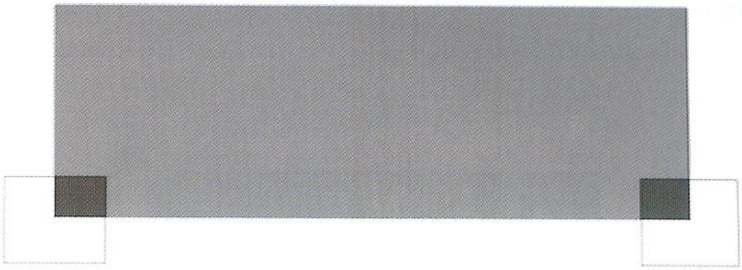


FIGURE 10.20
MBB beam scheme.

The initial model is elementary; a rectangular beam with two supports without any external applied loads.

The optimization strategy is the same as in the previous case except for two differences: the absence of agent loads, and the addition of material self weight from the reinforced concrete; during the creation of the FE model. The definition is as follows:

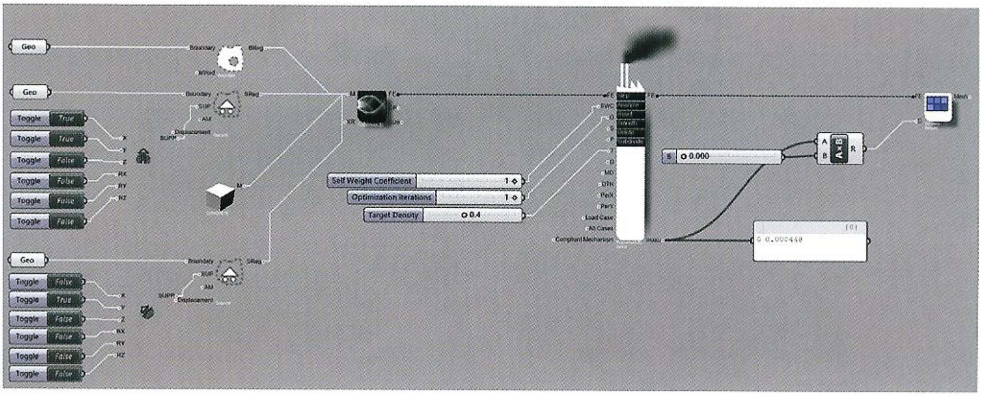


FIGURE 10.21
MBB beam algorithmic definition. The following QR code opens an enlarged image of the algorithm.

In this case there are two regions at each support. The first resists displacements in x and y direction while in the second resists displacements only in the y direction.

5

Material properties can be set from the *Stock* section in Millipede. In this case reinforced concrete has been chosen.

The remaining part of the definition is the same as the Michell Truss example. The solver calculates the maximum value of the structural displacement. This numeric value, multiplied to increase intensity, can be used to visualize the deformed configuration of the object.

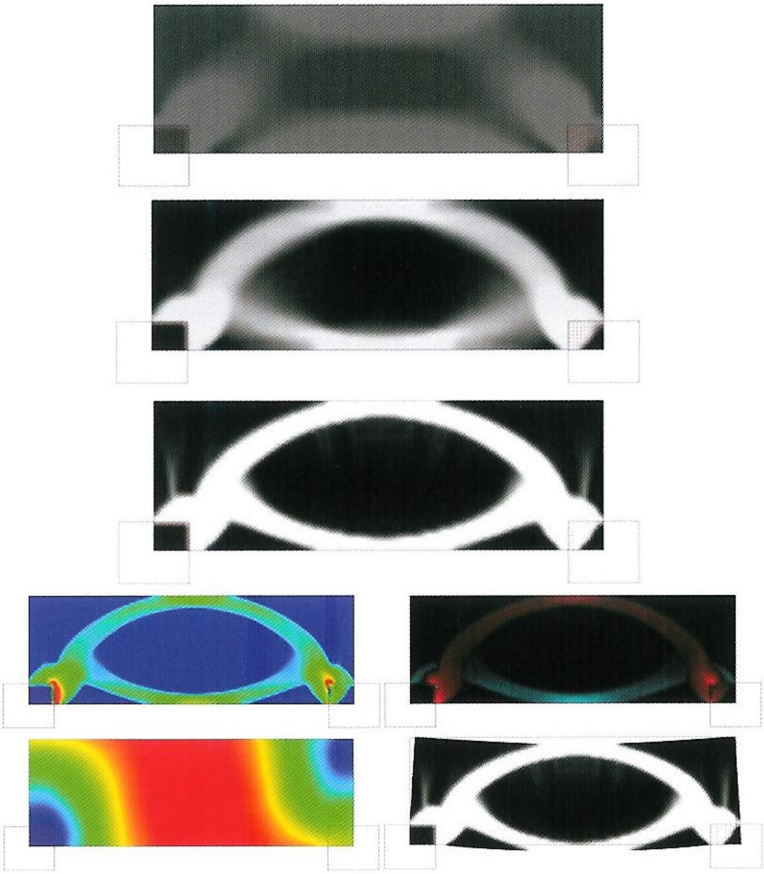


FIGURE 10.22
MBB beam optimization process and images of strain and deformation behavior of the structure.

The third application investigates topological optimization applied to a three- dimensional bridge structure. The schematic model is a tri-dimensional rectangular volume with two supports and without any external applied loads.

The optimization strategy is the same as in the previous case, there are no agent loads, and the material self weight from the reinforced concrete will be considered during the creation of the FE model. However in this instance, the design space and the supports are three-dimensional.

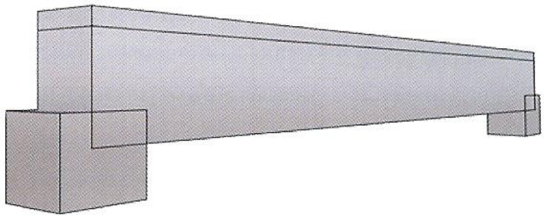


FIGURE 10.23
Bridge scheme.

Evolution optimization relies on the known data of a starting point to generate a variable output. In this case it is necessary to maintain the framework so a further geometric element is required to identify the volume which will not undergo the evolution process. The element that will not undergo optimization is a parallelepiped having the same length of the former beam and a arbitrary height. The parallelepiped will be inset within the design space framework. This new element will represent the input data (3DDensityRegion), a region were everything inside of it does not undergo optimization (non-design space).

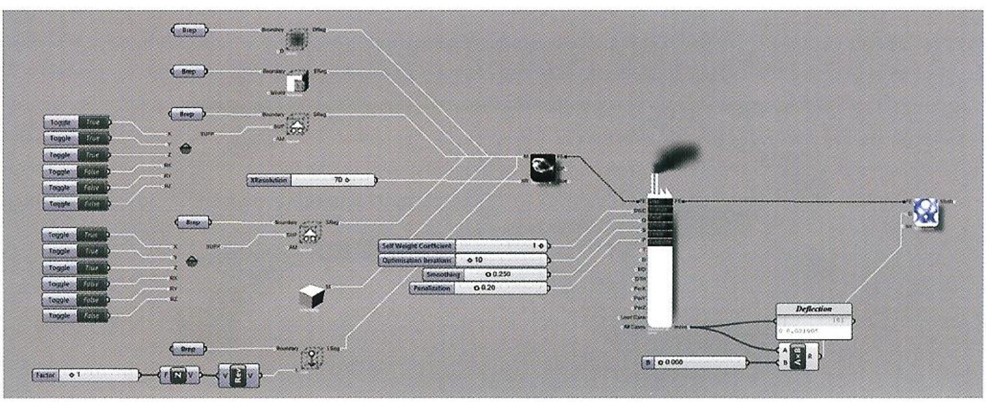


FIGURE 10.24
Algorithmic definition for the bridge optimization. The QR code opens an enlarged image of the algorithm.

6

All the components used in this definition are the exact three-dimensional transposition of those described in the two previous examples. The inputs that define the initial conditions, design space, non-design space, supports and constitutive material, are used to create the FEM and thus connected to the solver.

At the end of the analysis and at the end of the evolution optimization, the results can be visualized by using the component *3DisoMesh*, which: renders the geometry, displays the internal stresses and the maximum displacements of the structure.

By observing the results of the Extended Evolution Structural Optimization (XESO), on which Millipede is based, the behavior can be understood.

FIGURE 10.25
Optimization iterations and stress and deflections visualization.

Starting from a framework which remains unaltered during the optimization process, the XESO does not only remove the material with low stress levels, it also adds material where required. For instance material is added at the restraints, and at the connection between the extremity and the framework.

The examples analyzed represent a first approach to topological optimization. The simplicity by which it is possible to obtain valid architectonic results is not equated to the mathematical complexity required to calculate a solution. It is highly recommended to face the problem under a theoretical point of view and be able to manage the variables during the calculations phase.

The evolution algorithms can also lead to improvements in other applications. Since XESO is versatile it is possible to use the technique not only for research on structural optimization but also for solutions to other problems that can benefit from constant improvement algorithms.

This chapter is part of the Ph.D. thesis developed by Davide Lombardi at School of Advanced Studies "Gabriele d'Annunzio" (Pescara, Italy) under the supervision of prof. arch. Livio Sacchi, teacher at Università degli studi "Gabriele d'Annunzio" and prof. arch. Alberto Pugnale, lecturer at University of Melbourne (Australia) about evolutive optimization techniques in architecture.

Davide Lombardi (1985) architect, graduated in Architecture (Master of Sciences) with honors at Università degli Studi "Gabriele d'Annunzio" Chieti-Pescara. In 2012 started his PhD course at School of Advanced Studies "Gabriele d'Annunzio" and he is currently developing his thesis under the supervision of Alberto Pugnale, lecturer in architectural design at University of Melbourne. His interest in the field of computational design, focused on structural design and optimization/analysis, led him to start a personal training following high level workshops with international trainers as the AA Rome Visiting School. He taught at "Dynamic Morphologies" workshop organized in Rome by the Oxford Brookes University. He collaborated with architecture firms in Pescara and Rome working on international projects in Saudi Arabia and Ethiopia.

10.6 Optimization: finding solutions with Grasshopper

Optimization solvers enable users to solve real-world problems by finding the best solution among feasible alternatives. There are many optimization solvers available. Fundamentally, users must recognize the appropriate solver for a particular problem. Solvers can be grouped into two categories:

1. **Exact solvers**: find the *best* solution and yield the same result each time;
2. **Heuristic solvers**: find an approximate solution when no exact solution can be found. The resulting output can vary from one solution to the next.

The first category includes *linear solvers*, and the second category includes *evolutionary solvers*. Evolutionary solvers apply the principles of natural evolution to the problem of finding an optimal solution. This section will introduce two Grasshopper optimization solvers: *Galapagos* and *Goat*. *Galapagos* is a built-in Grasshopper solver developed by David Rutten and *Goat* is a plug-in for Grasshopper developed by Simon Flöry[36]. The following text will discuss how to use these solvers as a tool for optimization from a designer's point of view.

10.6.1 Optimization problems

A problem can be described by a model composed of two elements:

1. An *objective function* that we want to *minimize* or *maximize*;
2. A set of *variables* which affect the value of the objective function. Variables are usually defined by a range, with a start and end value.

For example, a basic problem is to find a point on a curve that is closest to a given external point A. In this case the *objective* is the *d(t)* function that describes all possible distances between the points on the curve and the external point A. The only *variable* is the parameter (t) ranging between 0 and 1. *The optimal solution* for this problem is the value of (t) that *minimizes* the d(t) function.

NOTE 36

Simon Flöry received his PhD in mathematics from Vienna University of Technology in 2010. For many years, he has been developing and consulting software solutions in geometry processing and architectural geometry. His ongoing research focuses on effectively processing and optimizing geometric data for geodesy, medical applications and facade engineering. http://www.rechenraum.com/en/

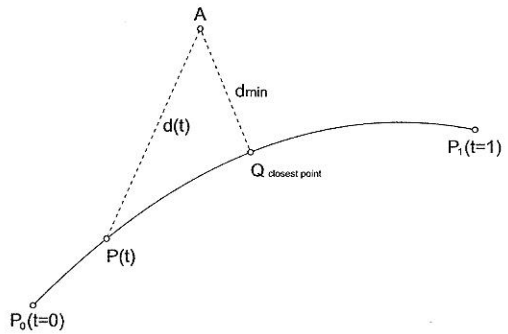
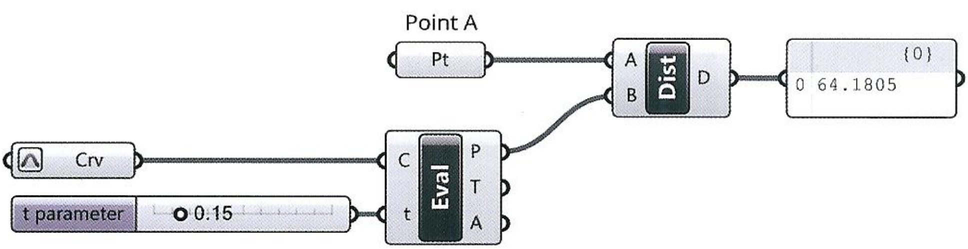


FIGURE 10.26
The objective function d(t) and the parameter (t) ranging between 0 and 1.

10.6.2 Exact solvers. Goat plug-in

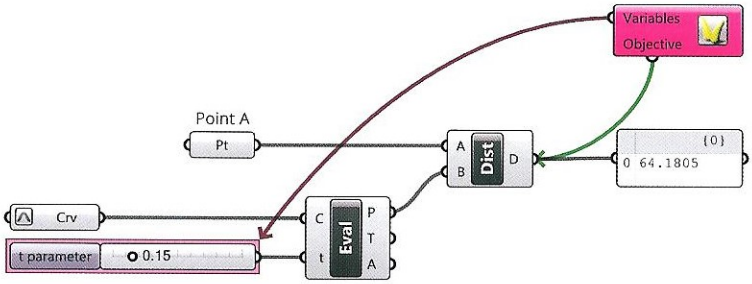
Optimization problems can be translated into Grasshopper definitions; the following sequence calculates the distance between point A and a generic point P on a reparameterized curve. The position of point P with respect to the curve is controlled by the input (t) of the component *Evaluate Curve*.



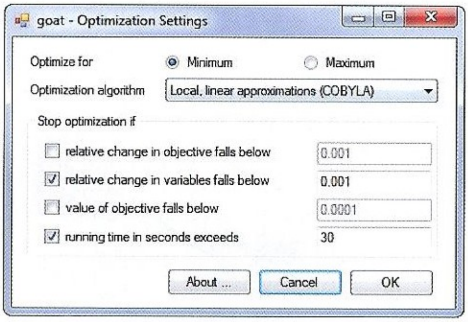
By moving the *slider* we can read the distance value on the *Panel* and we might "manually" try to find the *t* value related to the minimum distance *d*. Of course this method is inconvenient and will not generate an exact value without significant efforts. In order to find the optimal solution the *Goat* solver can be used. After installation, *Goat* can be found in *Special* > *Util*; the component has two slots for outgoing connections instead of the ingoing connections of standard components.

1. The *Variables* slot can maintain one or more *Number Sliders*. In Grasshopper the variables related to the optimization problem are **always Number Sliders**;

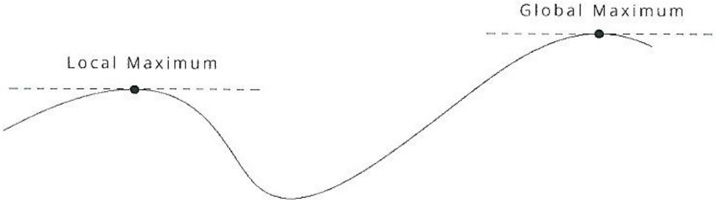
2. The *Objective* slot is always connected to a *numerical output* to minimize or maximize. The *Objective* input has **a single value** connected to the slot.



Double-clicking the *Goat* component opens the contextual settings window which can be used to set several parameters. The first option is used to minimize (*Minimum*) or maximize (*Maximum*) the *objective* function; in the example, the distance component output (D) is optimized for the minimum output.



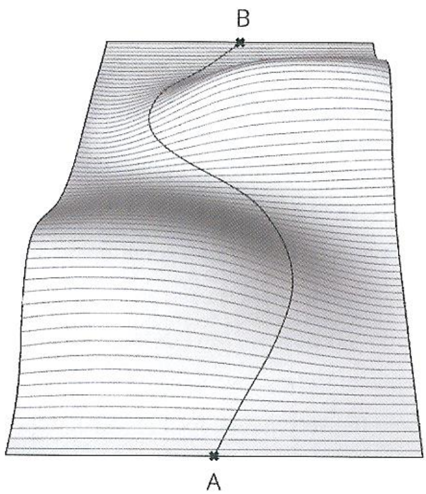
The second option allows to select among two type of optimization based on *Local* or *Global* algorithms. Roughly speaking, *Local* algorithms attempt to find exact solutions by inspecting values close to the starting value of the variable, while *Global* algorithms inspect all possible values. For very complex problems it is often useful to perform a *Global* algorithm and then a *Local* optimization.



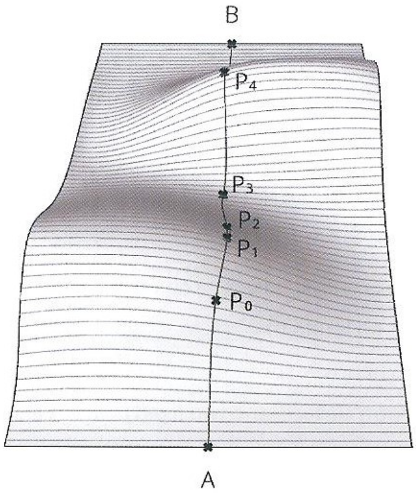
Lastly, clicking *OK* will initiate the optimization procedure. The slider will move guided by *Goat* until the optimum value (minimum distance) is reached. When the procedure ends two values can be read: the (t) parameter on the number slider, and the minimized distance output (D), stored in a panel.

10.6.3 Evolutionary solvers. Galapagos

Heuristic evolutionary solvers are used when optimization problems have a large number of variables and an optimal solution cannot be found through exact solvers. The evolutionary solver embedded in Grasshopper, *Galapagos*, is a heuristic solver that will be introduced in the following example. Given a reparameterized freeform surface, and two fixed points A and B, located on opposite edges of the surface, the *Galapagos* solver can be used to calculate the shortest path on the surface that connects the two points.

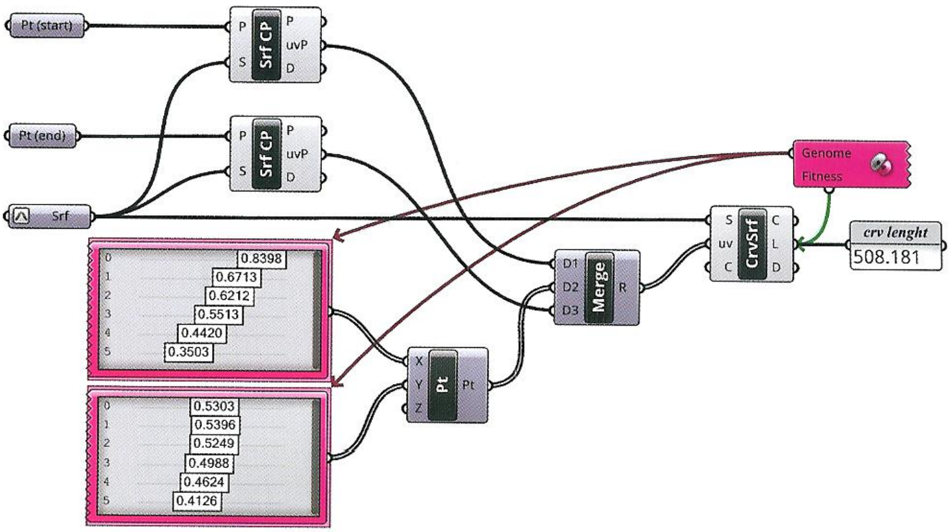


The first step in the shortest distance definition is to describe a *path curve*. The path curve is defined as a *flexible* interpolated curve that is coincident to the surface with fixed ends A and B. The curve geometry is controlled by the *uv* coordinates of a set of interpolation points P_i (P_0, P_1, P_n). By changing the coordinates of the interpolation points, different curve lengths will be output. The heuristic solver is used to find the interpolation points coordinates that correspond to the curve with the minimum length, among all feasible alternatives.



The *path curve* is created using the component *Curve on Surface* (Curve > Spline). The *Curve on Surface* component creates an interpolated curve on a surface input (S), through an arbitrary set of points input (uv) expressed in the LCS. In this instance, the *Curve on Surface* component requires three sets of *uv coordinates*, which are collected by a *Merge* component, whose inputs are:

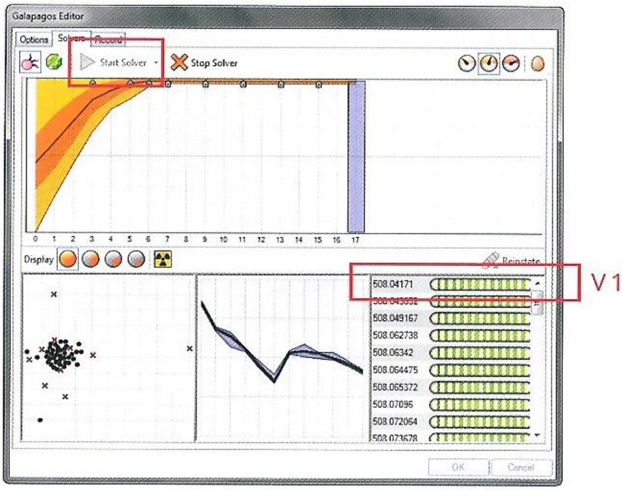
- D1: are the fixed coordinates of the start-point (A) converted to the Local Coordinate System through the *Surface CP* component (see 3.7.2).
- D2: are the (uv) coordinates of an arbitrary set of points (P_i) on the surface. To define the points the *Construct Point* component is used with coordinates set through sliders ranging between 0 and 1. For this purpose we can use the ***Gene Pool*** (Params > Util), a component that embeds a collection of sliders. Two *Gene Pool* components are used, the first one for the (u) coordinates and the second one for the (v) coordinates. The number of sliders or *Gene Count* as well as the sliders range and significant digits can be set by double-clicking the *Gene Pool* component. The larger the number of sliders, the higher the accuracy of the resulting curve.
- D3: are the fixed coordinates of the end-point (B) converted to the Local Coordinate System through the *Surface CP* component.



Since the *Curve on Surface* output (L) returns the curve's length value as the (uv) coordinates change position, a fixed element and a variable element exist respectively, that can be used to define an optimization procedure for an evolutionary solver. For this optimization *Galapagos* (Params > Util) is used. Similar to *Goat, Galapagos* has two ports for outgoing connections: *Genome* which corresponds to *Variables* in *Goat*, and *Fitness* which corresponds to *Objective* in *Goat*. The *Genome* port is satisfied by the arbitrary set of sliders embedded within the two *Gene Pool* components. The *Fitness* port is connected to the L-output of the *Curve on Surface* component.

Double-clicking the *Galapagos* component opens a contextual menu, where the *Fitness* can be set to be minimized or maximized.

Since, the shortest length is desired the *Fitness* option is set to *Minimize*. To perform the optimization the *Solvers* tab is opened and the *Start Solver* button is selected.

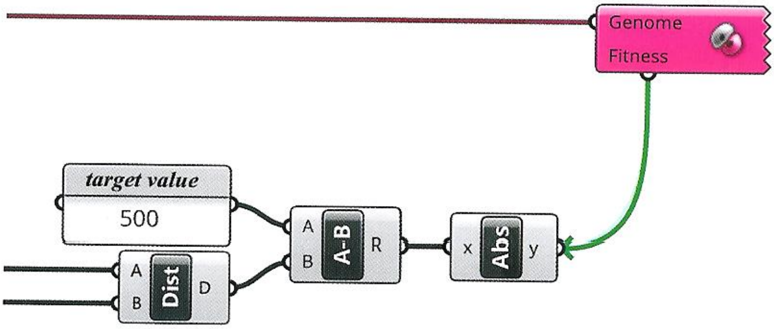


The optimization process runs until an optimum is found, which can take minutes or hours depending on the complexity of the solution. Galapagos can be stopped by selecting the button *Stop Solver* if the first value displayed on the bottom-right window (V1 in the previous image) becomes stable, tending to a specific number. In this example, the optimization returns a set of sliders representing (uv) coordinates of the interpolation points that correspond to the minimal path curve. The shortest curve is returned by the C-output of the *Curve on Surface* component.

Evolutionary solvers apply principles of evolution to problem solving, and are actually based on *population of candidate solutions*. The candidates attempt to reach better solutions with better genetic characteristics through *mutations*, *crossovers* and random changes. This is in contrast with exact solvers that target the single best solution. Roughly, evolutionary solvers perform a *natural selection* among possible solutions through a process in which only the best members of population (according to the specific *Fitness*) survive.

10.6.4 Target values

Both exact and evolutionary solvers operate by minimizing or maximizing a *Fitness* function. In some instances a target value for the *Fitness* function is desired; in these cases the absolute value difference between the *Fitness* function and the target value are minimized and maximized.



In the above definition the *Fitness* is defined as the distance between two points. In this instance, the target value is 500, the absolute value of the distance minus the target value is minimized using the *Fitness* function set to minimize.

Crassula "Buddha's Temple", photographed at the Huntington gardens, Los Angeles.