

UNIVERSITÀ DEGLI STUDI DI CASSINO
E DEL LAZIO MERIDIONALE



Dottorato di Ricerca in
Metodi, modelli e tecnologie per l'ingegneria
Ciclo XXXII

Computing in the Arts, a computational approach to Sculpture design.

Supervisors and co-supervisors

Prof. Maura Imbimbo

Prof. Ernesto Grande

PhD candidate

Alfonso Oliva

INDEX

CHAPTER 1 THE HISTORICAL USE OF COMPUTATIONAL METHODS

- 1.1 INTRODUCTION
- 1.2 BACKGORUND
- 1.3 STATE OF THE ART
- 1.4 THE RESEARCH INTENT
 - 1.4.a Pre-design phase / feasibility study
 - 1.4.b Schematic design (sd)
 - 1.4.c Design development phase (dd)
 - 1.4.d Construction documents (cd)
 - 1.4.e Bidding
 - 1.4.f Construction administration (ca)
- 1.5 THE NEW APPROACH TO DESIGN

CHAPTER 2 ALGORITHMS AS A MEANS TO DESIGN

- 2.1 DESCRIPTION OF THE ACTIVITY
 - 2.1.a Background
 - 2.1.b Research activity
 - 2.1.c Introduction to concepts of optimization
- 2.2 ALGORITHMS OF OPTIMIZATION
 - 2.2.a Choice criteria
 - 2.2.b Description
 - 2.2.b.i Simulated annealing

- 2.2.b.ii Tabu Search
- 2.2.b.iii Genetic Algorithm
- 2.2.c Choice of the algorithm
 - 2.2.c.i Applying GA to the Cantilever problem

CHAPTER 3 SOFTWARE SPECIFICATIONS

- 3.1 RHINO
- 3.2 GRASSHOPPER
- 3.3 REVIT
- 3.4 DYNAMO
- 3.5 KANGAROO
- 3.6 KARAMBA
- 3.7 GALAPAGOS
- 3.8 CSI PRODUCTS
- 3.9 INTEROPERABILITY
 - 3.9.i The big picture
 - 3.9.ii An algorithmic approach to documentation

CHAPTER 4 CASE STUDIES

- 4.1 FIRST CASE STUDY – ED CARPENTER
 - 4.1.a Design and optimization workflow
 - 4.1.a.i Loading conditions
 - 4.1.a.ii Material
 - 4.1.a.iii Connections
 - 4.1.a.iv Observations
 - 4.1.a.v Optimization steps

4.1.a.v.1 Improve performance through geometry manipulation

4.1.a.v.2 Improve performance through monitored material quantities

4.1.a.v.3 Improve performance through controlled stance width

4.1.a.vi Connection design

4.1.b Results

4.1.c Conclusions

4.2 SECOND CASE STUDY – ALYSON SHOTZ

CHAPTER 5 FROM RESEARCH TO PRODUCT

5.1 THE CONCEPT

5.2 PRELIMINARY DESIGN

5.3 MATERIALS FOR 3D PRINTING

5.4 3D MODELING FOR FINITE ELEMENT ANALYSIS

5.5 FINITE ELEMENT ANALYSIS SET UP

5.6 SETTING UP THE OPTIMIZATION PROCESS

5.7 MULTI OBJECTIVE OPTIMIZATION

5.7.a Background

5.7.b Basics of MOO

5.7.c Pareto method

5.7.d Octopus plug-in for Grasshopper

5.8 SETTING UP MOO OPTIMIZATION

5.9 FABRICATION

CHAPTER 6
FURTHER DEVELOPMENT AND POTENTIAL
APPLICATIONS

BIBLIOGRAPHY

CHAPTER 1

THE HISTORICAL USE OF COMPUTATIONAL METHODS

1.1 Introduction

This research project aims on redefining design techniques and process for large scale sculptural works.

There has been a lot of innovation during the years in the Artistic field and particularly in the sculptural one. With the implementation during the last decade of advanced design techniques, like computational design (Fig.1), we find ourselves in an era where the concept of sculpture design can be completely redefined.



Fig. 1 Works from ICD Institute for Computational Design and Construction

Techniques like software development can now be paired with contemporary technologies such as algorithmic driven design, virtual reality, augmented reality and robotics, creating the strong basis for the innovation process in the Art field.

The proposed research project will explore some of the above mentioned techniques and combine those to prove how the combination of non-related fields can lead to a more efficient engineering design process.

1.2 BACKGROUND

Looking back at the history and evolution of sculpture design, there are few Artists that is worth mentioning for living a mark in the field of sculpture design.

To name a few and in temporary order, we find Willem de Kooning [1904], Kenneth Snelson [1927], Richard Serra [1938] and Alexander Calder [1898] amongst many other leading visionaries that in many different ways have helped shape and advance this field.

For the sake of the research, rather than focusing on their life, let's focus directly on the typology of structures that they envisioned and built.

Starting in fact, with de Kooning highly abstracted figurative sculpture (Fig. 2) very reminiscent of his figurative paintings we move to the tensegrity sculptures of Kenneth Snelson delicate in appearance but very strong and depending only on the tension between rigid pipes and flexible cables (Fig. 3). From that, we look at the monumental corten steel large pieces (Fig. 4) from Richard Serra who became a pioneer of large-scale site-specific sculpture contrary to Alexander Calder's medium scale kinetic sculptures (Fig. 5). Calder, known as the originator of the mobile, has in fact, pioneered a type of moving sculpture made with delicately balanced or suspended shapes that move in response to touch or air currents.



Fig. 2 Williem de Kooning



Fig. 3 Kenneth Snelson



Fig. 4 Richard Serra



Fig. 4 Alexander Calder

What is interesting though is what all these different minds have in common.

If we look more in depth at their bio, most of them have an artist background and that could certainly be a similarity but that's not what made them unique. What made these sculptors pioneers in the field was the use of engineering techniques, whether it was mechanical, structural or electrical in some cases.

These names have strongly influenced my engineering and artistic side, helped them coexist, and have given me the inspiration to start this research project (Fig. 5).

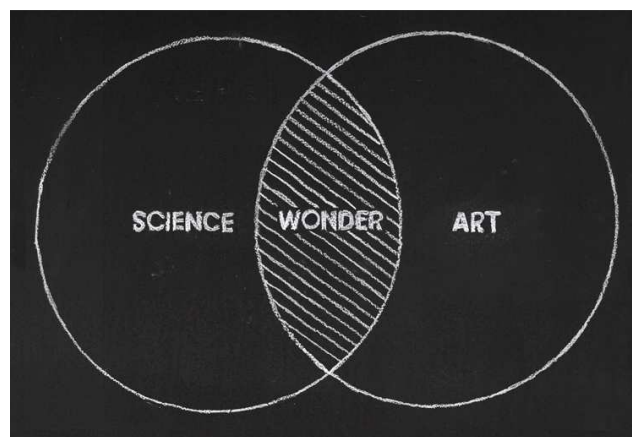


Fig. 5 The Art of Science

Further motivation behind this work comes from the need of innovation to solve contemporary design problematics that are encountered daily in a typical design process for an Art piece. This point will be introduced and discussed in the next chapter.

1.3 State of the art

We, as an industry are advancing at a very slow pace. Who is there to blame? What do we have today that we didn't have yesterday? and what else do we need? These are the most common questions that we hear in the EC (Engineering and Construction) industry.

First problem being the nonintegrated design process and in particular the lack of conversation between Artist (Client) and Engineer (Consultant) from the early stage of the design. This, in fact, is the first obstacle to the implementation of new technologies to optimize the design and overall workflow.

Let's explore that gap by looking at how the design workflow currently looks.

The current design process consists in a series of back and forth communications between the Artist and the Engineer that most of the times ends up being inefficient and counterproductive. What we just defined as "inefficiency" though, is nothing else that the use of two different languages, the Artistic language and the Engineering language. Establishing a good dialogue is key.

To further stress out the importance of the dialogue from the early stage of design, based on the experience accumulated over the years, it is safe to state, that even a sculpture conceptualized with the use of the latest software and technologies is, in most cases, not fully taking advantage from the technology itself because of the lack of

coordination amongst the different practices. The gap we are referring to though, is most of the times intangible.

This, is one of the problematic that this research project aims to address through the creation of what I define as a “sandbox” where both Engineers and Artist can meet and play together and speak the same language. That language in this case is called: “computational design”.

Before diving into the various opportunity that computational design offers, let’s look at other factors that have a big weight in the design process. Another more practical factor, is certainly costs. Costs are, in fact, often times the main driver and a brake on the implementation of new design process and/or technology.

That is only partially true though. It is in fact true only in the current design process where the above mentioned inefficiency of dialogue ends up consuming big part of the budget (up to 40% in some cases). The approach proposed in this research aims on bringing that percentage all the way down to 5% in some cases, to leave space for implementation of technology and advance the industry as a whole.

Now that we understand that the main inefficiencies in the design process come down to something simple, yet complicated to get rid of, how do we tackle this factor that is so critical to guarantee a good outcome for each project?

Through example of successful case studies. The case studies below will show how the use of computational design techniques has brought back to life and to construction projects that were declared out of budget and dead.

To put it simple, we need to optimize. Computational Design techniques can help us do that, not strictly/only in the engineering process but rather in the design process a whole. An example of the

potential disruption that computational design can provide to the design process is presented by Ramboll UK and is shown in Fig. 6.

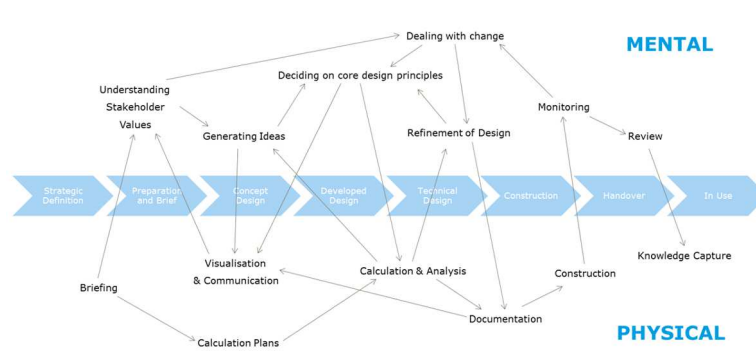


Fig. 6 Scheme of how Computational Design can disrupt the design process

Other side of the coin is that, as we aim on disrupting a design process that has been established for many years, we'll have to face all the practical challenges that go with it.

Biggest challenge to face is to make the individuals collaborate and not challenge each other.

Computational Design and its related processes of optimization, in fact, is nowadays a practice and an actual form of art and for that reason needs to have enough space in the design process to be expressed. This often times does not convey with the idea or ego of the Artist that could take a more conventional or comfortable approach to his/her design.

Innovation comes in fact with a price. This last point, more aimed on the management of the design process is equally fundamental to achieve a successful outcome of the project.

1.4 The research intent

All in all the aim of the research is to find where computational processes fits and in which form within the design workflow to improve efficiency.

To understand and define that, we need to take a more in depth look at the current established design process.

For a large scale sculpture the same design phases of an Architectural project apply. Here is a summary of all phases of design shown in Fig. 7.

There are six phases of design:

Pre-design or feasibility study

Schematic Design

Design Development

Construction Documents

Bidding

Construction Administration

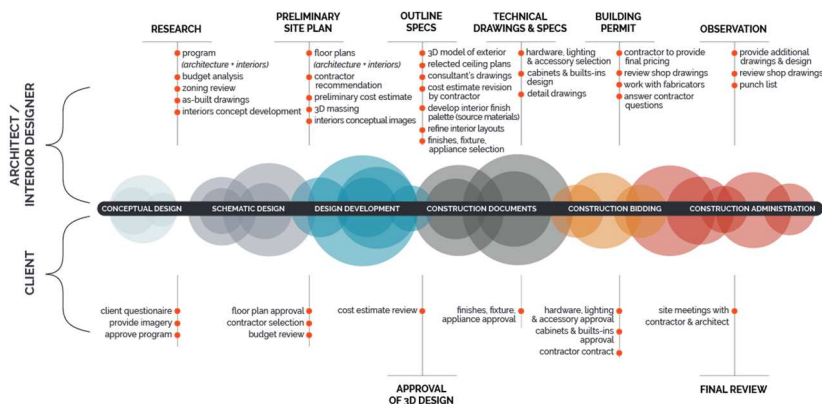


Fig. 7 The typical design process

1.4.a Pre-design phase / feasibility study

Pre-Design is a general term to describe the process prior to any form of actual design. This will include preliminary research on the site and the surrounding areas, scheduling and, if applies, any code restriction for the sculpture.

Essentially pre-design will be determining the information we need to begin design.

During this phase, the Artists also starts the so-called preliminary sketches, a mix of design and notes form the research. An example of a preliminary design sketch from Richard Serra is shown in Fig. 8.

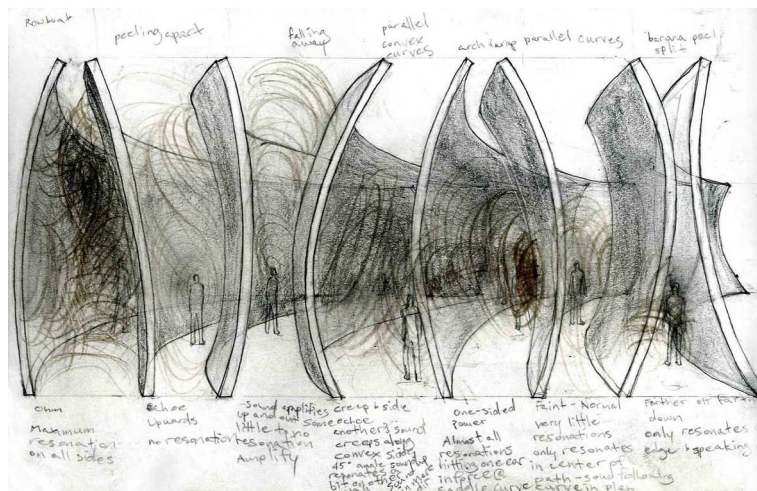


Fig. 8 Richard Serra preliminary design sketch

1.4.b Schematic Design (sd)

The goal of schematic design is to start developing the shape and size of the piece with some basic integrated design thoughts.

During this phase, overall plan views and 3D views are developed. Usually, most of the focus is centered on the final look of the piece

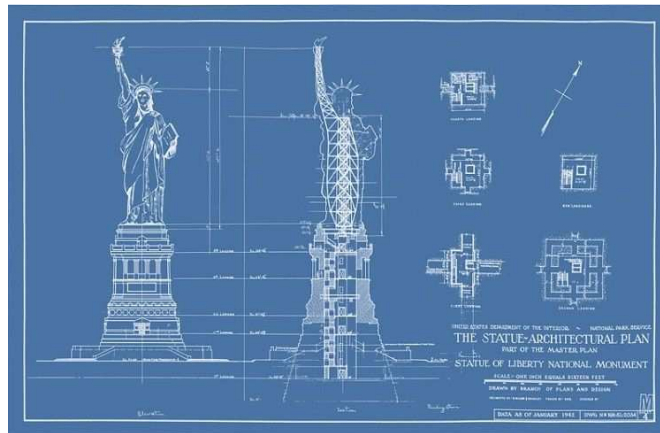


Fig. 10 Statue of Liberty blueprint

1.4.d Construction Documents (cd)

In the construction document phase all the technical design and engineering including structural engineering and detailing are finalized, and if needed, heating air conditioning and ventilation systems, plumbing, electrical, energy calculations, and all products and materials are selected and scheduled.

A drawing set including a filing set (required only for large pieces) for approval from the Department Of Buildings and a separate set of Construction Drawings is prepared. Fig. 11 shows an example of a construction drawing for a glass sculpture.

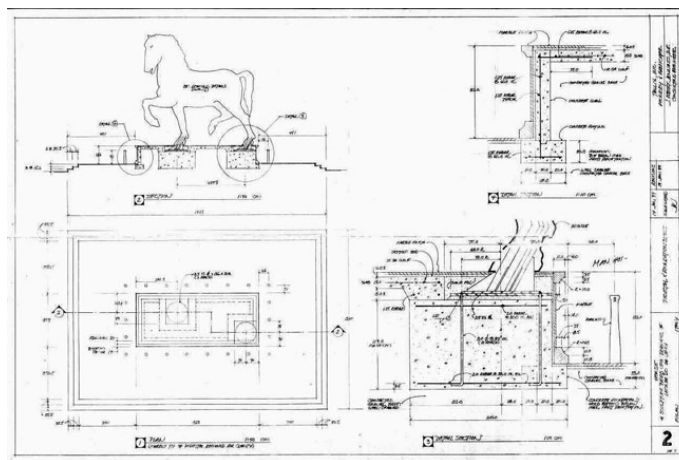


Fig. 11 Construction Drawing for a glass sculpture

1.4.e Bidding

Bidding should be self-explanatory. This is the time to select a fabricator for the job to then proceed with construction. Multiple contractors submit bids on the job or the client can directly hire a contractor without getting competitive bids. In some cases, the engineer here will assist the client.

1.4.f Construction Administration (ca)

The Construction Administration is the final phase of the design process. While this phase is the longest, it does not usually comprise the majority of the work. On a typical project the engineer supervise construction. That means that the engineer will periodically visit the job site to check on progresses and ensure the contractor is following the plans.

This process has been historically adopted for the design of most sculptures as well as buildings.

1.5 The new approach to design

With time, and particularly in the last ten years, we have learned a lot from the above mentioned design process. To help this disruption, an interesting factor has helped re-shape and accelerate the design process, that factor is Computational Design.

Computational Design has in fact individually tapped into the different design phases and helped streamline the overall workflow as a whole.

Early adopter have certainly been fabricators. This means that the last phase of design (i.e. CA phase) it has actually been the first

phase of design to benefit from Computational Design processes and techniques.

Why is that? The answer is pretty straightforward: “Automation”.

In fact, with the development of advanced design techniques more and more space has been given to automation tasks on the fabrication side. Two axis machine all the way up to 6 axis machine (aka robotic arms. See Fig. 12) have been implemented in the workflow to optimize the efficiency and precision of the fabrication process.



Fig. 12 Two robotic arms working on a sculptural piece

This is not news anymore though. Robotic arms, in fact, have been implemented for daily tasks by the industry.

The downstream seems to be covered in terms of implementation and even if there is more to do is well launched towards innovation. Fabrication is in fact considered as the downstream of the design process since is the step that precedes the delivery of the Art piece.

With this research work we are proposing to enter from the other extreme of the design process: Schematic Design phase (SD phase).

As defined previously, SD is currently a phase where very minimal (if any) engineering is conducted.

This factor, has in my opinion and experience, historically lead many project over-budget. The reason behind this is the fact that the SD phase is all about developing the vision without any of the engineering thoughts that are instead introduced in the next phase. Computational Design processes can help us solve that and integrate engineering as early as the SD phase without compromising the Artistic vision.

Let's then look at how we can make that happen. Specifically, we aim to understand how computational design can bridge the art and engineering field during SD and how that affects the subsequent phases of design.

To explain that, two case studies and one research project will be presented in the next chapters.

On a more practical level, Computational Design applied to the case studies presented in this research has allowed, through algorithm aided design to inform a large number of design intent in a relatively short amount of time. Without going in depth yet into the different algorithms that can be developed to optimize a design we can state, and later on prove with case studies, that the implementation of this technique in the design process is the key ingredient for a successful integrated design process.

CHAPTER 2

ALGORITHMS AS A MEANS TO DESIGN

2.1 Description of the activity

2.1a Background

Technology is moving fast and is finally breaking into our industry. Technology ties directly into computational design, hence, engineering processes mainly given to their common base, algorithmic processes. This intersection allows us today to think about different workflows for the design of structures. To reiterate, although this thesis research aims on exploring different processes for sculpture design, this workflow can be applied to any kind of structure.

This chapter is an introduction on new workflows encompassing the same established engineering techniques we have used so far in our industry paired with optimization techniques to ensure the delivery of designs that preserve the Artist vision while being on budget.

2.1b Research activity

One other question we aim to answer in the following chapters is: Which are these established engineering techniques and how do they get combined with more contemporary workflows?

Established engineering techniques refer to the design of a structural element whether we are referring to a beam, a column or a façade panel. The way it gets combined with the current workflows is by attaching the variables contained in the design of the elements to

custom algorithms of optimization. The advantage behind that is that a large number of iterations can be performed in less time than a human could do with or without a conventional software.

2.1c Introduction to concepts of optimization

To introduce concepts of optimization explored, we'll take as an example a cantilevered beam with a uniform distributed load [fig.13]:



Fig. 13 Cantilevered beam with distributed load

Our aim is to calculate the deflection in B:

$$\delta_B = F a^3 / (3 E I) \quad (2d)$$

where

$$\delta_B = \text{maximum deflection in B (m, mm, in)}$$

where:

F - Load (N, lb)

a - Length of beam between A and B (m, mm, in)

b - Length of beam between B and C (m, mm, in)

I - Moment of Inertia (m^4 , mm^4 , in^4)

E - Modulus of Elasticity (N/m^2 , N/mm^2 , psi)

The aim is to optimize the design of the cantilever. The next question to be asked is, what do we want to optimize the beam for?

These are 3 examples of factors we could optimize for:

- Deflection (Comfort)
- Material quantities (\$)
- Material usage (Weight of the structure)

The above mentioned factors could later on be combined to conduct a so called “multi-objective optimization”.

For this example we’ll focus on deflection only.

The first step is to pick the algorithm of optimization that we think is most useful for this task. This is in fact probably the most important part of the design process.

Every algorithm works in a different way and follows a different path. Some of them are affected by the initial configuration of the variables, some are affected by the previous iteration, some give a relative max or minimum, some give an absolute max or minimum.

2.2 Algorithms of optimization

2.a Choice criteria

The answer is not a straightforward one and most likely different people would pick different algorithms even when working on the same project. The key behind this kind of optimizations is in fact to be able to have a good understanding of the way the algorithm operates to be able to set up a solid algorithmic definition.

The three main algorithms we explored in this Research are listed below.

2.b Description

2.b.i Simulated annealing

This algorithm [Fig. 14] became very well-known after proving his benefits with outstanding performances when applied to optimization of a one-dimensional objective function [Fig. 15] but mostly when it was first applied to the “traveling sales man problem”, [Fig.16] problem where a salesman must visit some large number of cities while minimizing the total mileage traveled. If the salesman starts with a random itinerary, he can then pairwise trade the order of visits to cities, hoping to reduce the mileage with each exchange. The difficulty with this approach is that while it rapidly finds a local minimum, it cannot get from there to the global minimum. Simulated annealing improves this strategy through the introduction of two tricks. The first is the so-called "Metropolis algorithm" (Metropolis et al. 1953), in which some trades that do not lower the mileage are accepted when they serve to allow the solver to "explore" more of the possible space of solutions. Such "bad" trades are allowed using the criterion that:

$$e^{-\Delta D/T} > R(0, 1),$$

where ΔD is the change of distance implied by the trade (negative for a "good" trade; positive for a "bad" trade), T is a "synthetic temperature," and $R(0, 1)$ is a random number in the interval $[0, 1]$. D is called a "cost function," and corresponds to the free energy in the case of annealing a metal (in which case the temperature parameter would actually be the kT , where k is Boltzmann's Constant and T is the physical temperature, in the Kelvin absolute temperature scale). If T is large, many "bad" trades are accepted, and a large part of solution space is accessed. Objects to be traded are

generally chosen randomly, though more sophisticated techniques can be used.

The second trick is, again by analogy with annealing of a metal, to lower the "temperature." After making many trades and observing that the cost function declines only slowly, one lowers the temperature, and thus limits the size of allowed "bad" trades. After lowering the temperature several times to a low value, one may then "quench" the process by accepting only "good" trades in order to find the local minimum of the cost function. There are various "annealing schedules" for lowering the temperature, but the results are generally not very sensitive to the details.

There is another faster strategy called threshold acceptance (Dueck and Scheuer 1990). In this strategy, all good trades are accepted, as are any bad trades that raise the cost function by less than a fixed threshold. The threshold is then periodically lowered, just as the temperature is lowered in annealing. This eliminates exponentiation and random number generation in the Boltzmann criterion. As a result, this approach can be faster in computer simulations.

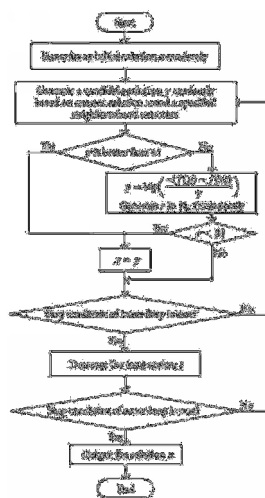


Fig. 14 Flowchart of simulated annealing algorithm

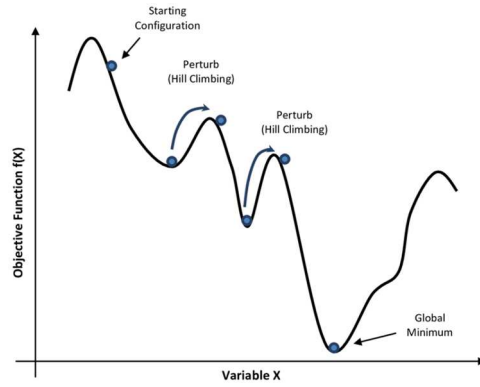


Fig. 16 Simulated Annealing optimization of a one-dimensional objective function

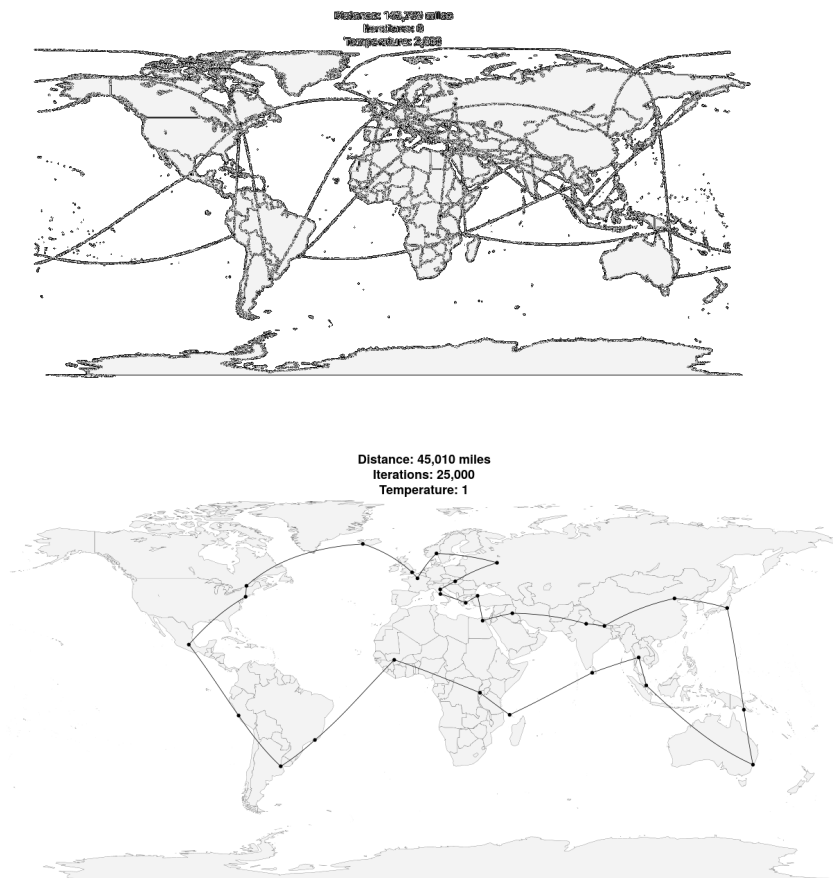


Fig. 15 Traveling Salesman without and with Simulated annealing

2.b.ii Tabu Search

Tabu Search [Fig. 16] is a Global Optimization algorithm and a Metaheuristic or Meta-strategy for controlling an embedded heuristic technique. Tabu Search is a parent for a large family of derivative approaches that introduce memory structures in Metaheuristics, such as Reactive Tabu Search and Parallel Tabu Search.

The objective for the Tabu Search algorithm is to constrain an embedded heuristic from returning to recently visited areas of the search space, referred to as cycling. The strategy of the approach is to maintain a short-term memory of the specific changes of recent moves within the search space and preventing future moves from undoing those changes. Additional intermediate-term memory structures may be introduced to bias moves toward promising areas of the search space, as well as longer-term memory structures that promote a general diversity in the search across the search space.

- Tabu search was designed to manage an embedded hill climbing heuristic, although may be adapted to manage any neighborhood exploration heuristic.
- Tabu search was designed for, and has predominately been applied to discrete domains such as combinatorial optimization problems.
- Candidates for neighboring moves can be generated deterministically for the entire neighborhood or the neighborhood can be stochastically sampled to a fixed size, trading off efficiency for accuracy.
- Intermediate-term memory structures can be introduced (complementing the short-term memory) to focus the search on

promising areas of the search space (intensification), called aspiration criteria.

- Long-term memory structures can be introduced (complementing the short-term memory) to encourage useful exploration of the broader search space, called diversification. Strategies may include generating solutions with rarely used components and biasing the generation away from the most commonly used solution components.

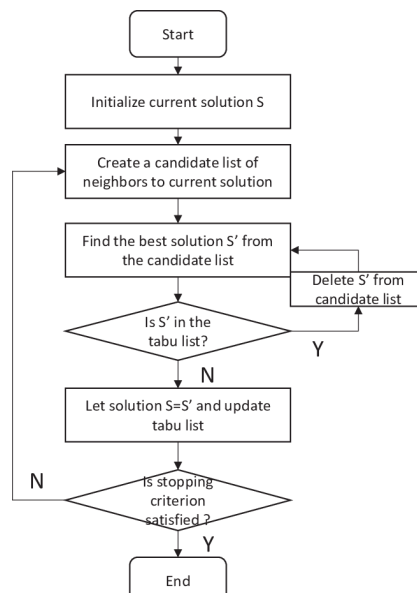


Fig. 16 Tabu flowchart

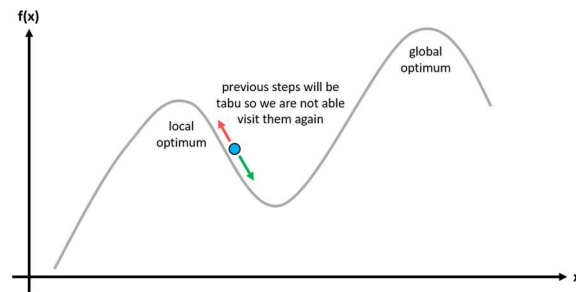


Fig. 17 Tabu Search Algorithm

Taboo Search Java implementation

```
public State solve(State initialSolution) {  
  
    State bestState = initialSolution;  
    State currentState = initialSolution;  
  
    int iterationCounter = 0;  
  
    //we make a predefined number of iterations  
    while(iterationCounter<Constants.NUM_ITERATIONS) {  
  
        //get all the available (reachable) states  
        in the neighborhood  
        List candidateNeighbors =  
        currentState.getNeighbors();  
        //get the tabu list  
        List solutionsTabu =  
        tabuList.getTabuItems();  
  
        //get the best neighbor (lowest f(x) value)  
        AND make sure it is not in the tabu list  
        State bestNeighborFound =  
        neighborSolutionHandler.getBestNeighbor(states,  
        candidateNeighbors, solutionsTabu);  
  
        //we are looking for a minimum in this case  
        if  
        (bestNeighborFound.getZ()<bestState.getZ()) {  
            bestState = bestNeighborFound;  
        }  
  
        //we add it to the tabu list because we  
        considered this item  
        tabuList.add(currentState);  
  
        //hop to the next state  
        currentState = bestNeighborFound;  
  
        iterationCounter++;  
    }  
  
    //solution of the algorithm  
    return bestState;  
}
```

2.b.iii Genetic Algorithm

The genetic algorithm, (GA) is a method for solving both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution. The algorithm repeatedly modifies a population of individual solutions.

In terms of analytical steps, there are two main operation in this algorithm, the selection and the crossover.

The selection process goes first, where two parent chromosomes gets selected from a population according to their fitness (the better fitness, the bigger chance to be selected) and then the crossover [fig. 18] of the parents takes place to form a new offspring (children).

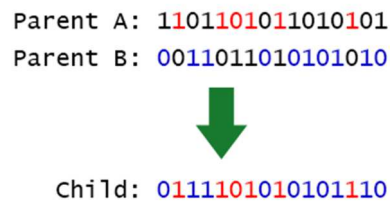


Fig. 18 Example of crossover operation in Genetic Algorithm

2.c Choice of the algorithm

The three above mentioned algorithms have been implemented in the early stage of design for this case study. By monitoring speed and behaviors we were able to make a decision on which one to adopt for the whole optimization workflow.

The GA (Genetic Algorithm) was chosen for two main reasons: first one being speed. In fact, genetic algorithms seemed to be faster than

the other compared algorithms. Second, the fact that genetic algorithms is affected by the initial configuration of the “parents” (see explanation below) giving in a way more and more space to consider different solutions for the same design. Last but not least, genetic algorithms give in output a series of best performing solutions and not a unique one. This, applied to design translate in a large number of schemes with optimized performance that the artist can pick from.

All of this comes with a downside though. GA, in fact, can very easily produce results out of point of relative maximum or minimum. In other words, this kind of optimization does not guarantee the absolute optimal solution, but again, speed and the variety of exploration makes this a great candidate for these kind of optimizations.

2.c.i Applying GA to the Cantilever problem

The way we apply GA to the cantilever beam problem is the following. We assign as “Parents”, E and I with the aim of changing material properties and beam section to evaluate which are the best performing ones for the specific case.

By starting the process with an imaginary heavy member an imaginary weak material, the optimization process converges to a more reasonable result in few seconds.

$$I = 781250000 \text{ mm}^4$$

$$E = 200000 \text{ N/mm}^2$$

$$\text{Deflection at B} = 34.56 \text{ mm}$$

The way the algorithm converges to a solution is shown in Fig. 19, 20 and 21 and discussed here:

Step 1 - Generate random population of n chromosomes (suitable solutions for the problem)

Step 2 - Evaluate the fitness $f(x)$ of each chromosome x in the population

Step 3 - Create a new population by repeating following steps until the new population is complete

Step 4 - Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)

Step 5 - With a crossover probability cross over the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.

Step 6 - With a mutation probability mutate new offspring at each locus (position in chromosome).

Step 7 - Place new offspring in the new population

Step 8 - Use new generated population for a further run of the algorithm

Step 9 - If the end condition is satisfied, stop, and return the best solution in current population

Step 10 – Loop (i.e. go to step 2)

The aim of this simple exercise is, more than coming up with an optimized structure, to describe how GA convergences to a solution by starting with something that is far from an ideal desired output.

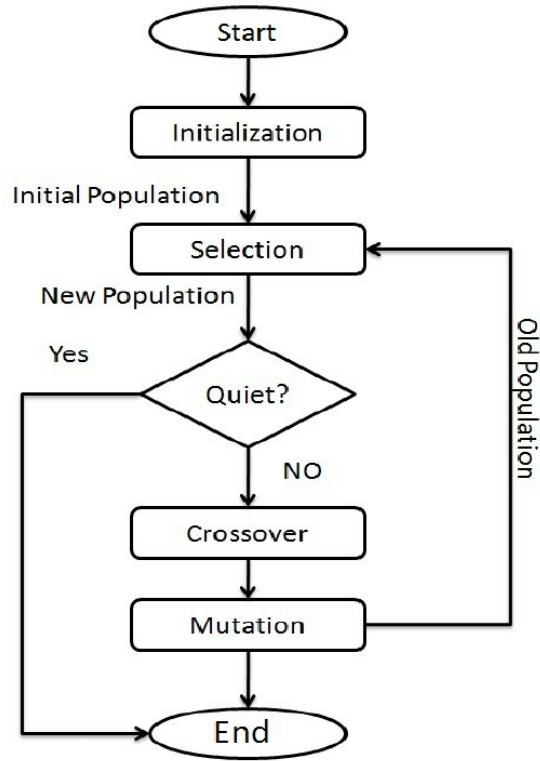


Fig. 19 GA Flowchart

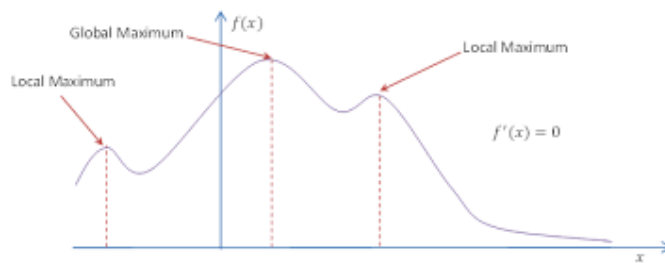


Fig. 20 GA 2D representation

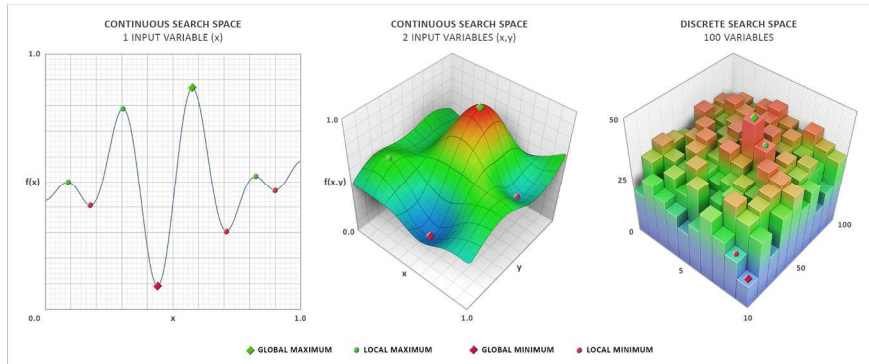


Fig. 21 GA Representation

Fig. 22 and Fig. 22.a shows the GA solution for the problem of the traveling salesman previously presented for the simulated annealing.

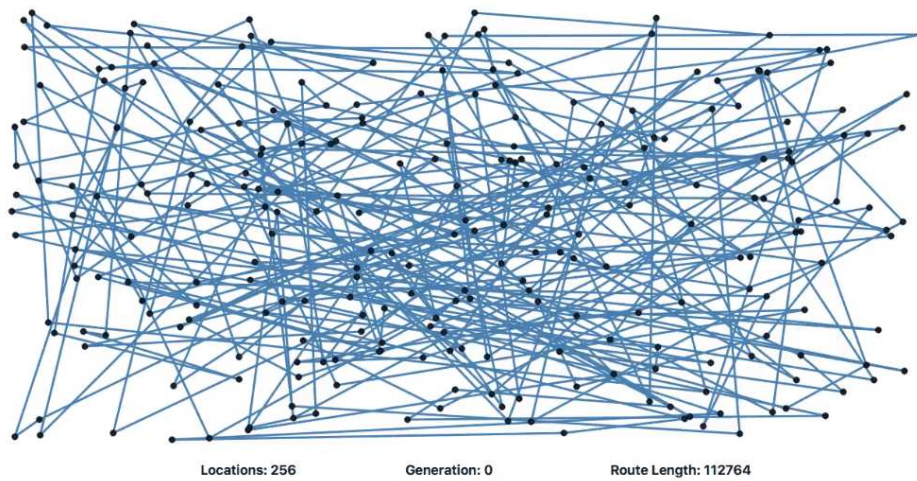


Fig. 22 Salesaman problem

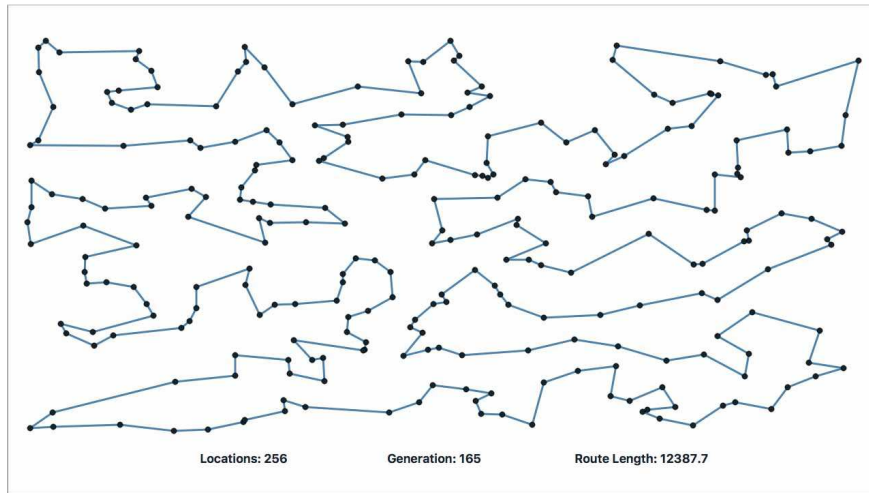


Fig. 22.a Salesaman problem solved with GA

Once the basics of algorithmic techniques are established we can look into how these tools can be applied to the design of sculptures as in the case study below.

CHAPTER 3

SOFTWARE SPECIFICATIONS

3.1 Rhino

Rhinoceros [Fig. 23] (typically abbreviated Rhino, or Rhino3D) is a commercial 3D computer graphics and computer-aided design (CAD) application software. Rhinoceros geometry is based on the NURBS mathematical model, which focuses on producing mathematically precise representation of curves and freeform surfaces in computer graphics (as opposed to polygon mesh-based applications).

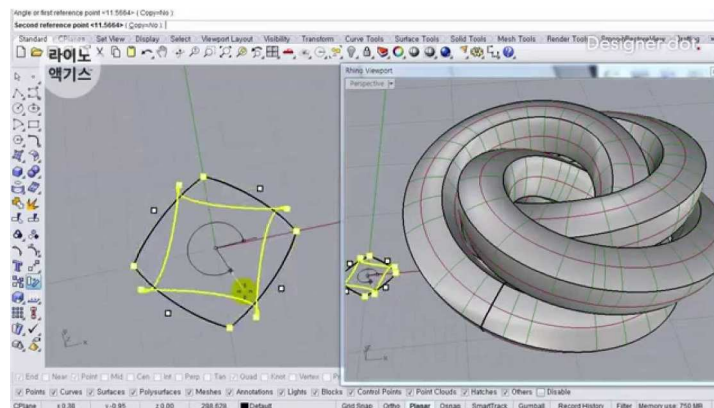


Fig. 23 Rhinoceros is widely used in processes of computational design.

Rhinoceros is primarily a free form surface modeler that utilizes the NURBS mathematical model. Rhinoceros' open SDK (Software Development Kit) makes it modular and enables the user to develop custom commands.

Rhinoceros supports two scripting languages, Rhinoscript (based on VBScript) and Python (V5.0+ and Mac).

3.2 Grasshopper

Grasshopper [Fig. 24] is a visual programming language and environment developed by David Rutten at Robert McNeel & Associates, that runs within the Rhinoceros 3D computer-aided design (CAD) application. Programs are created by dragging components onto a canvas. The outputs to these components are then connected to the inputs of subsequent components.

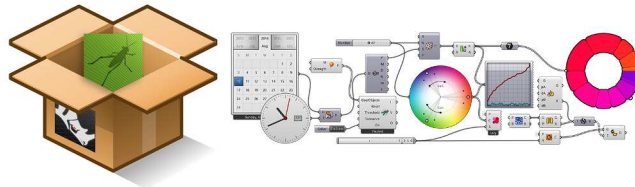


Fig 24 Grasshopper package for Rhino

Grasshopper is primarily used to build generative algorithms, such as for generative art. Many of Grasshopper's components [Fig. 25] create 3D geometry. Programs may also contain other types of algorithms including numeric, textual, audio-visual and haptic applications.



Fig 25 Grasshopper components

Advanced uses of Grasshopper include parametric modelling for structural engineering [Fig. 26], parametric modelling for architecture and fabrication.

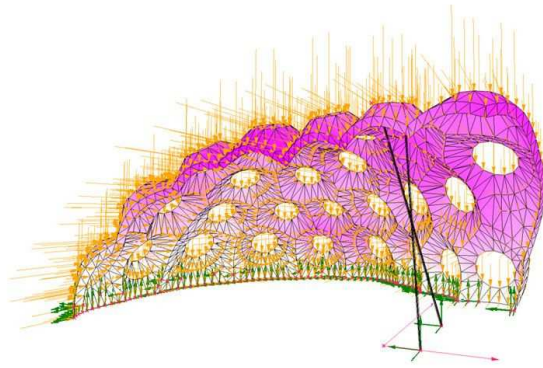


Fig 26 FEA solver for Grasshopper

The first version of Grasshopper was released in September 2007, and titled Explicit History. Grasshopper has become part of the standard Rhino toolset in Rhino 6.0 and later.

Grasshopper has been named as an endemic tool in the design world. The new Grasshopper environment provides an intuitive way to explore designs without having to learn to script and at the same time allows to write custom scripts within the application to push the boundaries of today's work. This thesis will use default components in grasshopper as well as custom scripts [Fig. 27] developed for the purpose of the optimization processes.

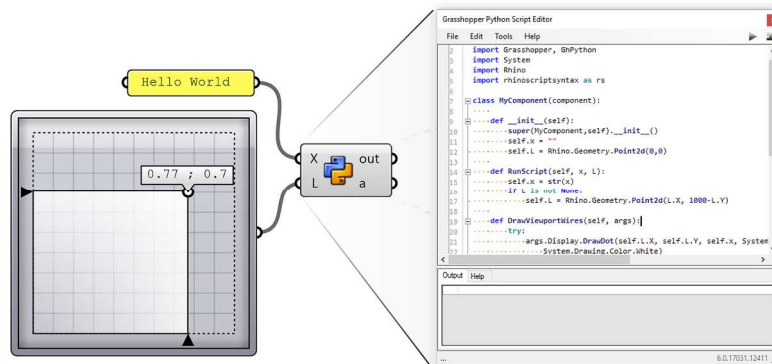


Fig. 27 programming in Grasshopper

3.3 Revit

Autodesk Revit [Fig.28] is a building information modelling software for architects, landscape architects, structural engineers, mechanical, electrical, and plumbing (MEP) engineers, designers and contractors. The original software was developed by Charles River Software, founded in 1997, renamed Revit Technology Corporation in 2000, and acquired by Autodesk in 2002. The software allows users to design a building and structure and its components in 3D, annotate the model with 2D drafting elements, and access building information from the building model's database.

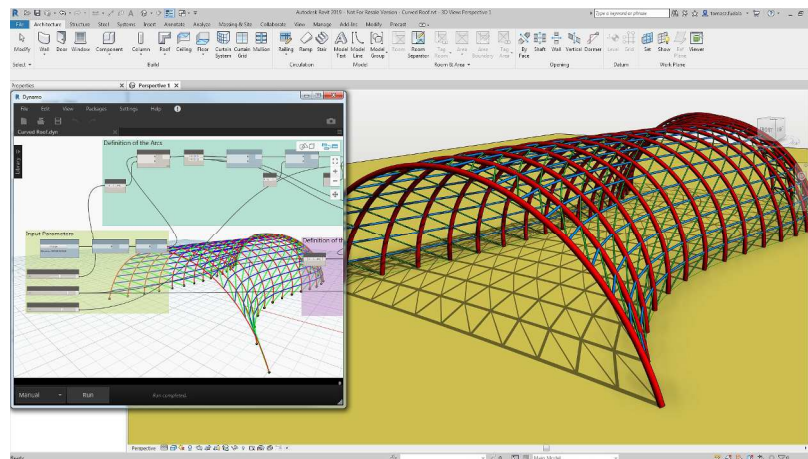


Fig. 28 Revit environment

Revit is 4D building information modeling capable with tools to plan and track various stages in the building's lifecycle, from concept to construction and later maintenance and/or demolition. In this thesis project, Revit has been mentioned as the old standard for drawing production but not implemented in the workflow for documentation since this work relies on digital submission to fabricators. Revit has been implemented instead, together with

Virtual Reality for coordination on the location of one of the sculptures discussed later.

3.4 Dynamo

Dynamo is a visual programming tool that works with Revit. Dynamo extends the power of Revit by providing access to the Revit API (Application Programming Interface) in a more accessible manner. Rather than typing code, with Dynamo you create programs by manipulating graphic elements called “nodes” [Fig. 29]. It’s an approach to programming better suited for visually oriented types, like architects, designers, and engineers.

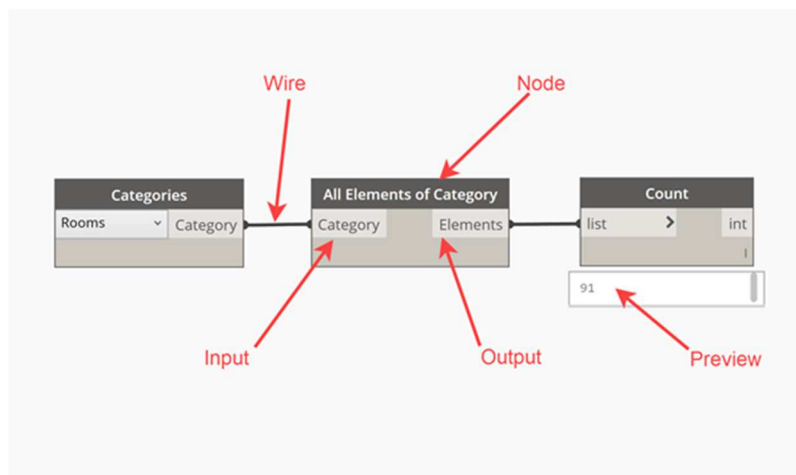


Fig 29 Dynamo nodes

In Dynamo, each node performs a specific task. Nodes have inputs and outputs. The outputs from one node are connected to inputs on another using “wires”. The program or “graph” [Fig 30] flows from node to node through the network of wires. The result is a graphic representation of the steps required to achieve the end design.

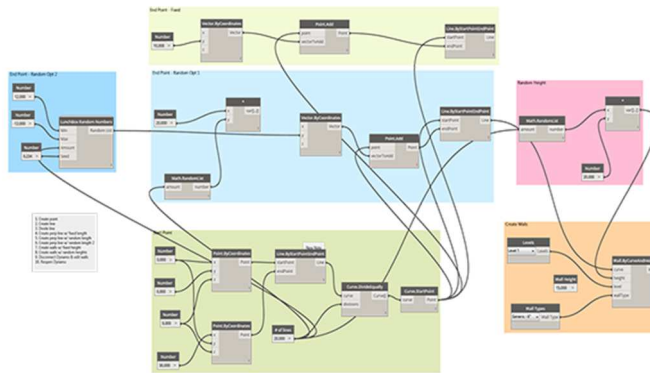


Fig 30 Dynamo Program

One of the strengths of visual programming and Dynamo, in particular, is ready access to a library of nodes [Fig 31]. Instead of having to remember the exact code you need to type to perform a certain task, in Dynamo you can simply browse the library to find the node you need.

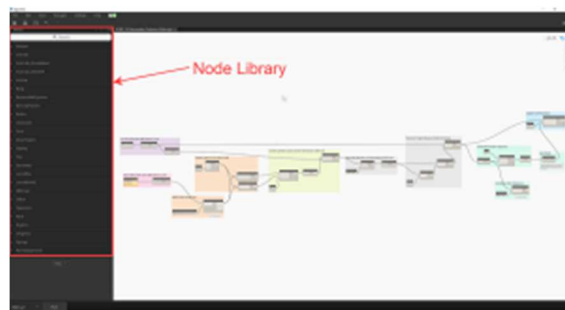


Fig 31 Library of nodes

Likewise, a contributing factor to Dynamo’s success is its user community. In addition to providing help on its forum, Dynamo users can also create node libraries or “packages” [Fig. 32] and upload them to a central repository. This repository can be searched directly from inside of Dynamo. To install the package, simply click the download button and it will install directly into Dynamo.

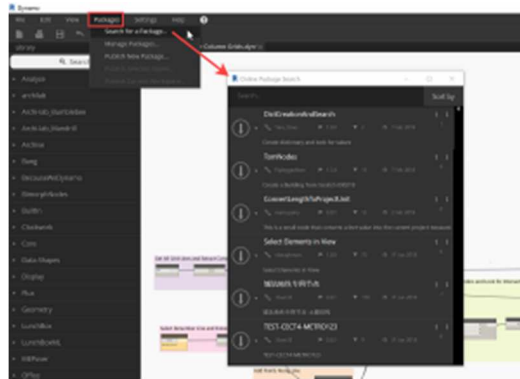


Fig 32 Dynamo Packages

Dynamo has been utilized in this thesis work to prepare model coming into Revit for Virtual Reality.

3.5 Kangaroo

Kangaroo [Fig. 33] is a Live Physics engine for interactive simulation, form-finding, optimization and constraint solving.

It consists of a solver library and a set of Grasshopper components.

Installation instructions and a draft manual are included with the latest download.

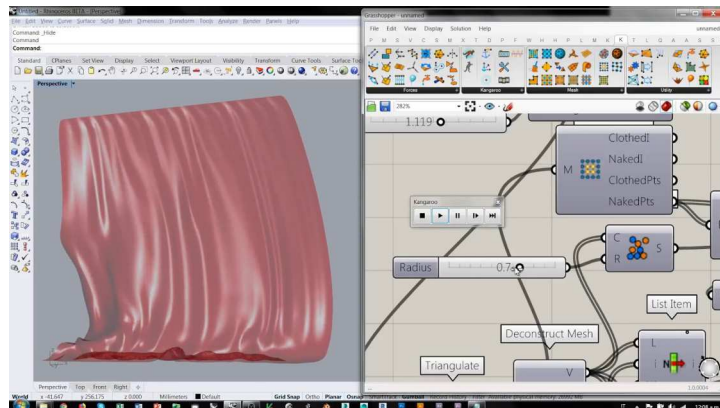


Fig. 33 Kangaroo for Grasshopper

3.6 Karamba

Karamba3D [Fig. 34] is an interactive, parametric finite element program. It lets you analyze the response of 3-dimensional beam and shell structures under arbitrary loads.

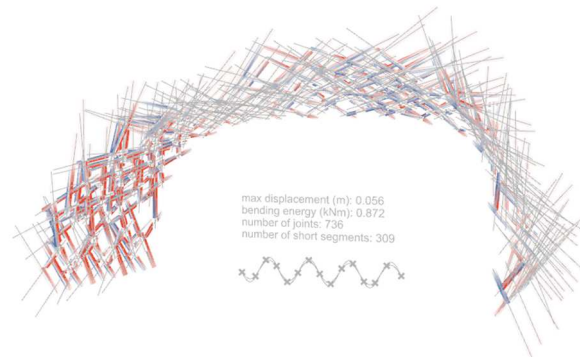


Fig. 34 Karamba for Grasshopper

3.7 Galapagos

Evolutionary solver plug-in for Grasshopper. Paired with parametric geometry allows for exploration of a large number of designs [Fig. 35].

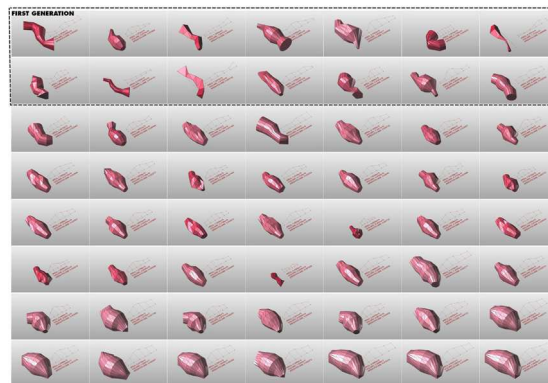


Fig 35 Galapagos for Grasshopper

3.8 CSI products

Computers and Structures, Inc. (CSI) is recognized globally as the pioneering leader in software tools for structural and earthquake engineering. SAP2000 [Fig. 36] is intended for use on civil structures such as dams, communication towers, stadiums, industrial plants and buildings and it has been used in this research to double check and confirm the results coming from the structural analysis conducted in Grasshopper with Karamba.

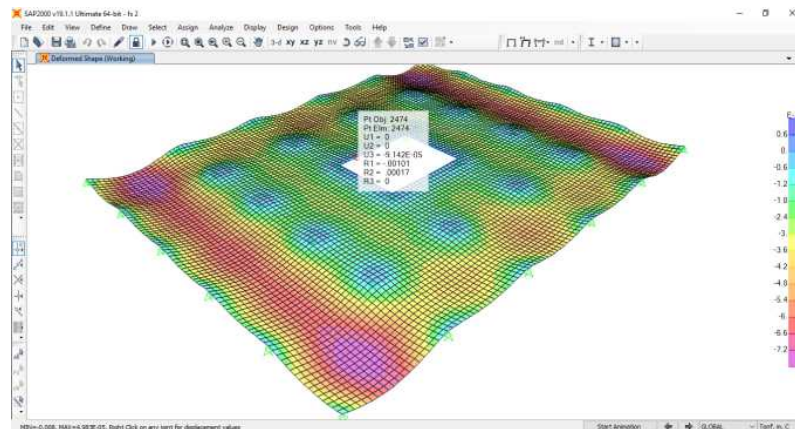


Fig 36 SAP 2000

3.9 Interoperability

The ability of graphical software to generate infinitely complex geometrical solutions using optimization techniques requires an equally concentrated effort to provide rational structural solutions. The growing experimentation with and implementation of Building Information Modeling (BIM), Parametric Modeling, and Structural Optimization requires software development and data management that are not a part of off-the-shelf software.

To better implement these techniques, part of the effort of this thesis work has been to bridge the gap between incompatible software to shorten the time between structural simulation (optimization) and structural reality (structural design).

The use of the techniques discussed in the following pages allowed for the rapid study of multiple SD level design options of the complex geometries shown in the case studies. Particular attention should be given to the flexibility and reaction time of these techniques in response to the Artist's visions.

3.9.i The big picture

Here we describe the general approach to a design problem, software involved and workflow process.

In all below case studies we start from a Rhino model, received from the Artist. The Rhino model gets then discretized using Grasshopper, a process that involves building parametric definitions and developing algorithms that recreate the model accurately in a nonnative environment.

The discretization of the model aims to make the structure easier to analyze computationally by dividing it into simpler components. The simplification process is different for each kind of geometry. Frames are broken down into straight line segments, surfaces into smaller three- or four-sided surfaces, and volumes into six-sided shapes. Surfaces are then split based on the intersections of frames and walls, and non-planar surfaces are simplified by creating a mesh from which faces are extracted.

This is just the starting point of the design process. In the following sections, we will show how easily, through the use of a custom developed interoperability platform [Fig. 37], structural models and BIM models can be created in SAP2000/ETABS and Revit from a source Rhino model. The basic flow of this exchange is summarized by the diagram in Figure below.

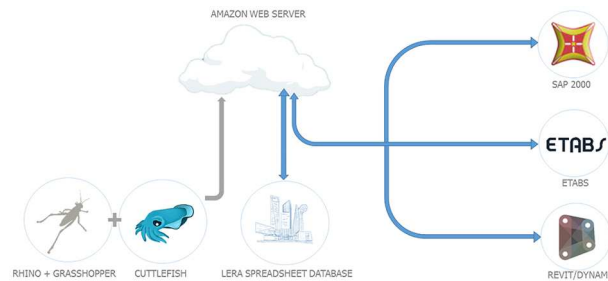


Fig 37 Custom developed interoperability platform

Cuttlefish is composed of two separate components. The first component, developed in C#, reads data from an existing Grasshopper model and writes this data to an SQL database housed on a cloud server (Amazon Relational Database Service). The data that is read from the Grasshopper model is saved as a set of points, which represent the endpoints for frames and rigid links, as well as vertices for surfaces and volumes of the geometry. Once the user provides a project name, the data is written and stored in the cloud in the form of tables based on the type of geometry it originates from.

The MySQL database consists of a set of tables [Fig. 38] that store information from different sets of geometry: a table for frames, a table for volumes, etc. Each table is further subdivided into fields that store geometric information from the model, as well as unique structural analysis results (Figure below).

The image shows a screenshot of a Microsoft Excel spreadsheet. The spreadsheet contains a large table with many columns and rows. The columns are labeled with letters A through Z, and the rows are numbered from 1 to 25. The data in the cells consists of various alphanumeric strings, likely representing database records. The spreadsheet is displayed in a standard Excel window with a menu bar at the top and a status bar at the bottom.

Fig 38 Example of MySQL database

The second component of the interoperability platform is a standalone docking application that serves as a container for software that does the translation work. This application, also developed in C#, reads the data stored in the cloud database and translates it to other software via their built-in API's. The software carries information relative to geometry as well as structural analysis results from the CSI products.

The model can either be built from scratch or added onto an existing SAP2000 or ETABS model. This comes particularly in handy during multiple iterations conducted in parallel with the optimization process.

All of the work involved in the conversion is performed behind the scenes, yet all the data can be download in an excel spreadsheet at any time to design member sizes.

Building analysis models from an AWS server

The models generated with the interoperability platform for CSI SAP2000 and CSI ETABS will be used to perform structural

analysis for the sculptural pieces and also for their specific components and connection points.

Besides the difficulties inherent in building complex models into Finite Element Analysis (FEA) software with a high level of accuracy, the even greater challenge solved with this process is to be able to react quickly to changes made by the Artist. The parametric modeling environment, combined with the ad-hoc platform to transfer geometrical information and structural design, enabled a smooth design process.

The other advantage of using the interoperability platform was for the different sets design spreadsheets developed for the design of the structural members. Analysis results, stored in the cloud database, were output upon request through the Interoperability Platform. This workflow allowed to work on the same spreadsheet with set of analysis data coming from different design iterations parsed and organized in a consistent way. This process decreased the possibility of introducing errors and increased productivity (reading data from the cloud was, in this case, ten times faster than extracting results from an FEA model).

3.9.ii An algorithmic approach to documentation

In order to document the design, a Revit model was prepared and used to create drawings in combination with the digital 3D model deliverable for fabrication. The process of generating the Revit model was integrated into the workflow by creating a link between the architectural Rhino model, the structural analysis model, and Revit. This was done extending the interoperability platform to Dynamo for Revit.

In order to translate the geometry from the Rhino model into Revit, the point coordinate data, as well as the column connectivity and the surface data, were extracted from the Rhino model using the Cuttlefish component in Grasshopper and stored as tables in the cloud.

This geometric data was then used to generate a Revit model of the structural system. A custom node was developed in Dynamo to read the point coordinate data and element connectivity from the cloud in order to generate the sculpture in Revit. Once the structure had been analyzed in ETABS and element section sizes had been calculated, the next step in the workflow was to assign section properties to the elements in Revit. The model is now ready to be placed on sheets that will be part of the final deliverable.

CHAPTER 4

CASE STUDIES

4.1 First case study – Ed Carpenter

The first case study presented in this research is a sculpture by artist Ed Carpenter to be installed in a confidential building in Manhattan. Installation is estimated for the end of 2020.

The sculpture as schematically shown in fig.39 was conceptualized by the artist as a 45' high piece composed by a large number of pipes supported by 10 columns.

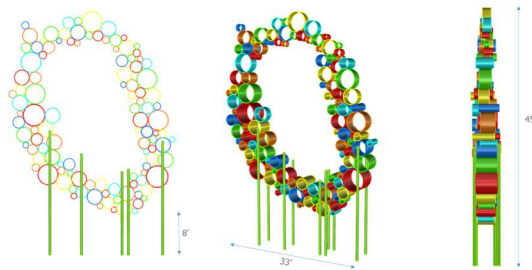


Fig. 39 3D view of Artist vision for the sculptural piece

The sculpture initially designed by an Engineering firm in Manhattan was facing anticipated deflection under wind load of 3'' (~8cm) as shown in fig. 40. Deflection in this case were not a problem, at least not yet. The sculpture in fact, was structurally sound and respected the deflection limits imposed by the code. The problem in this case was that the piece, given the current design, was over budget, and hence could not be built. The objective in this case was then to bring the project back in budget by optimizing the material usage while preserving the initial design and vision provided by the artist.

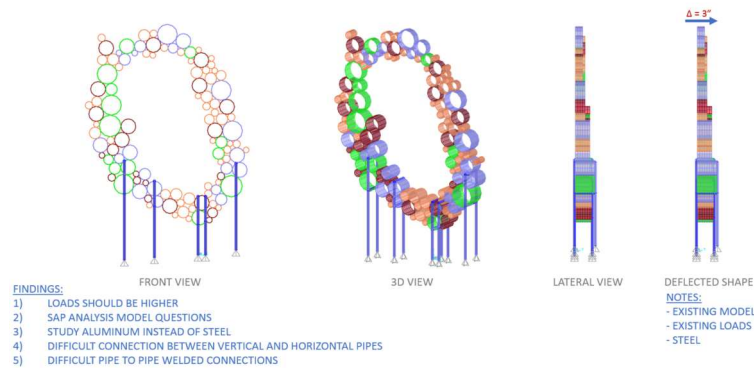


Fig. 40 3D view of Artist vision for the sculptural piece

4.1.a Design and optimization workflow

4.1.a.i Loading conditions

A first check of the existing loading assumption has been conducted. Fig 41 shows a comparison of the wind loads calculation (based on ASCE 7-10). In red there are the wind loads calculation from the original engineering design, in yellow there are the new wind loads based on my calculation. It is noticeable how the wind loads out of our calculations are 4 times higher than the original calculation. The reason for that is because of our assumption that the sculpture once installed on site will be covered because there will be construction around (4 buildings are being built on site). This, requested by the Artist, is to protect the piece until the surrounding works are completed.

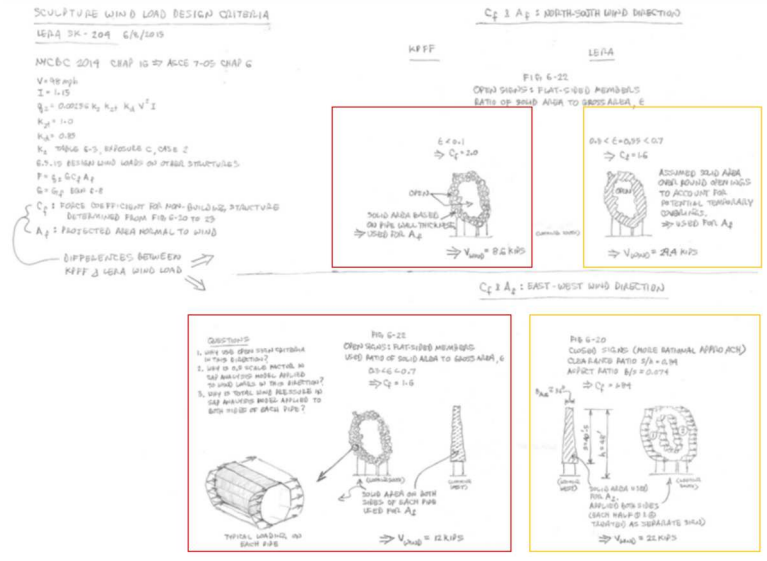


Fig. 41 Wind loads comparison. Existing design VS new design

This assumption has consequences in the calculation of the deflection of course. By implementing the new wind load in the F.E. model developed throughout the software SAP2000, in fact, the deflection spike up to a 21" number as shown in Fig. 42.

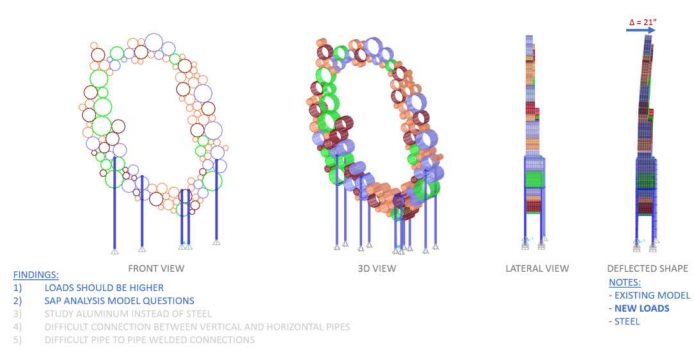


Fig. 42 Deflection under new Wind loads

4.1.a.ii Material

After checking the wind loads we moved into understanding the material utilized for the design. The engineering firm that was previously in charge of designing the piece had designed the sculpture to be in Steel. We wanted to make the sculpture lighter so we decided to explore Aluminum. This of course has its consequences. Aluminum has a strength that is 3 times less than steel. When inputting the mechanical properties of the Aluminum instead of the steel material of the SAP2000 model, we get what we already expected, a 60” deflection as shown in Fig. 43.

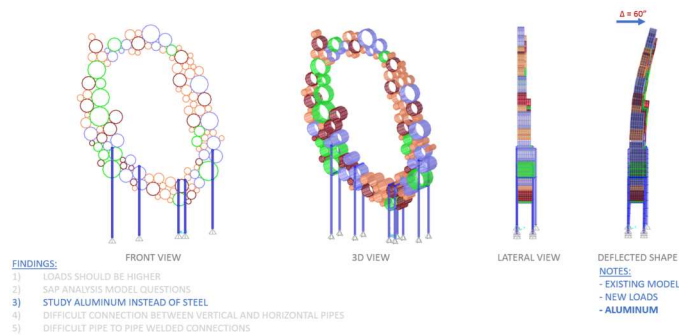


Fig. 43 Deflection increase due to change of material

4.1.a.iii Connections

The next logical step was to understand how the connection between the pipes and the columns worked. The previously engineered connection where a simple weld all over the sculpture. This process was working in advantage to strength but might have damaged the piece hence we studied a different solution. We decided to detail a saddle that would be shop welded on the column and upon which the pipe would be bolted on site [Fig. 44].

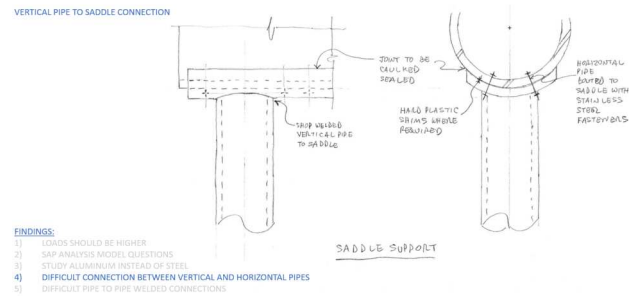


Fig. 44 Vertical pipe to saddle connection

This solution would definitely make the structure more flexible but at the same time produce a more elegant solution and prevent any damage to the Art piece. As expected, the deflection went up to 77” as shown in Fig. 45.

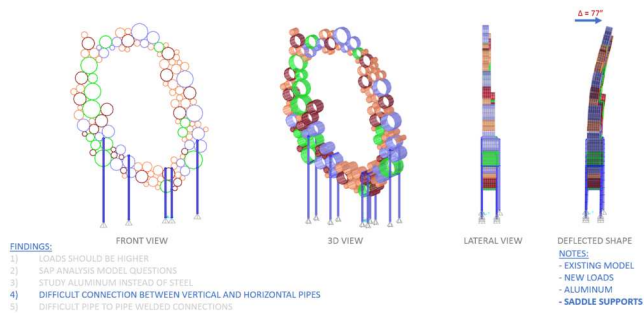


Fig. 45 Deflection increase due to pipe-saddle connection

4.1.a.iv Observations

At this point we not only had a structure that was over budget but also one that didn't work structurally. This problem definitely opened an opportunity for optimization techniques.

4.1.a.v Optimization steps

We decided to tackle the problem in a 3-step optimization as summarized in Fig. 46 to 49.

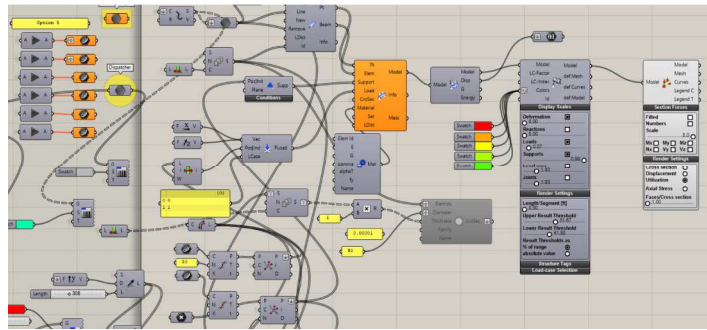


Fig 48 Stence of sculpture

Fig. 49 shows a visual summary of the three step optimization.

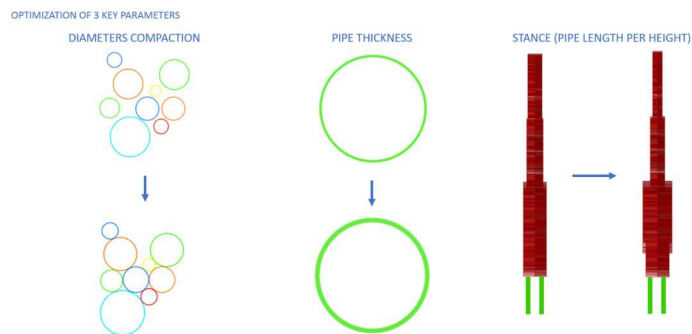


Fig. 49 Three step optimization

In order to do this, a custom defined Grasshopper definition was built and the plug in Galapagos and Karamba were used. Karamba is an FEA solver for grasshopper while Galapagos is an evolutionary solver that runs GA optimization. A localized screenshot of the grasshopper definition in Fig. 50 shows the input values for the optimization and the Galapagos interface running.

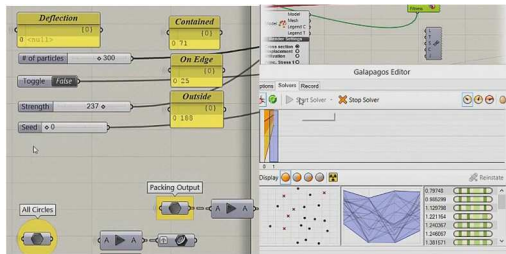


Fig. 50 Galapagos inputs and interface

The custom grasshopper definition allowed then to start the exploration for new solutions.

4.1.a.v.1 Improve performance through geometry manipulation

The first step that consisted in reorganizing the pipes to achieve a stronger structural system lead to 6 main solutions allowing for a 40% reduction in deflection as shown in Fig. 51.

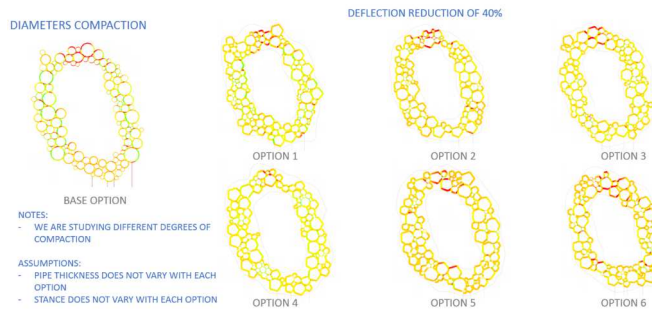


Fig. 51 Six best options out of the optimization process of Step1

To achieve this, a custom packing algorithm [Fig.52] has been developed in Grasshopper and then tied to FEA through Karamba and Galapagos to geometrically produce and calculate deflection for nearly 10,000 solutions amongst which the 6 best performing where then chosen and presented to the Artist.

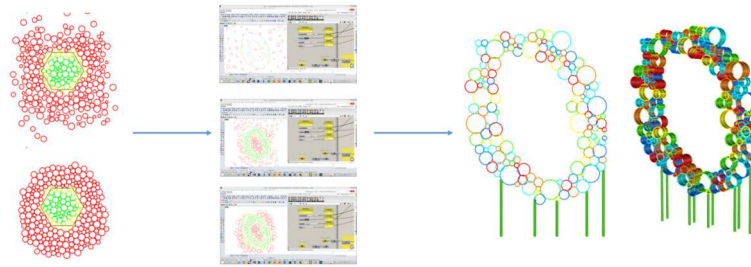


Fig. 52 Custom packing algorithm developed for step 1

Form the above 6 option, the Artist picked option 5 as his preferred solution so we continued the study with the new geometry.

It is important to explain the nature and math behind the packing algorithm. Wolfram Math World gives a good explanation of that and it has been my source for the development of the work.

A circle packing [Fig. 52] is an arrangement of circles inside a given boundary such that no two overlap and some (or all) of them are mutually tangent. The generalization to spheres is called a sphere packing. Tessellations of regular polygons correspond to particular circle packings (Williams 1979, pp. 35-41). There is a well-developed theory of circle packing in the context of discrete conformal mapping (Stephenson).

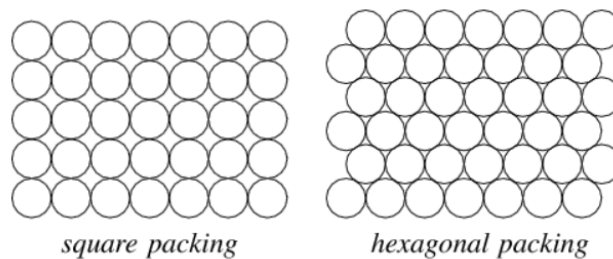


Fig 52 Circle packing

The densest packing of circles in the plane is the hexagonal lattice of the bee's honeycomb, which has a packing density of:

$$\eta_h = \frac{1}{6} \pi \sqrt{3} \approx 0.9068996821$$

Gauss proved that the hexagonal lattice is the densest plane lattice packing, and in 1940, L. Fejes Tóth proved that the hexagonal lattice is indeed the densest of all possible plane packings.

Surprisingly, the circular disk is not the least economical region for packing the plane [Fig. 53]. The "worst" packing shape is not known, but among centrally symmetric plane regions, the conjectured candidate is the so-called smoothed octagon.

Wells (1991) considers the maximum size possible for n identical circles packed on the surface of a unit sphere.

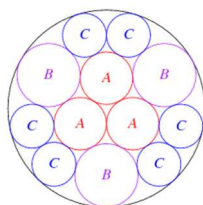


Fig. 53 Circle disk region

Using discrete conformal mapping, the radii of the circles in the above packing inside a unit circle can be determined as roots of the polynomial equations:

$$a^6 + 378 a^5 + 3411 a^4 - 8964 a^3 - 10233 a^2 + 3402 a - 27 = 0$$

$$169 b^6 + 24978 b^5 + 2307 b^4 - 14580 b^3 + 3375 b^2 + 162 b - 27 = 0$$

$$c^6 + 438 c^5 + 19077 c^4 - 15840 c^3 - 360 c^2 + 2592 c - 432 = 0$$

with:

$$a \sim 0.266746$$

$$b \sim 0.321596$$

$$c \sim 0.223138$$

The following table gives the packing densities η for the circle packings corresponding to the regular and semiregular plane tessellations (Williams 1979).

tessellation	η exact	η approx.
{3, 6}	$\frac{1}{12} \sqrt{12} \pi$	0.9069
{4, 4}	$\frac{1}{4} \pi$	0.7854
{6, 3}	$\frac{1}{9} \sqrt{3} \pi$	0.6046
$3^2 .4^2$	$(2 - \sqrt{3}) \pi$	0.8418
$3^2 .4 .3 .4$	$(2 - \sqrt{3}) \pi$	0.8418
$3.6 .3 .6$	$\frac{1}{8} \sqrt{3} \pi$	0.6802
$3^4 .6$	$\frac{1}{7} \sqrt{2} \pi$	0.7773
3.12^2	$(7\sqrt{3} - 12)\pi$	0.3907
4.8^2	$(3 - 2\sqrt{2})\pi$	0.5390
$3.4 .6 .4$	$\frac{1}{2} (2\sqrt{3} - 3)\pi$	0.7290
$4.6 .12$	$\frac{1}{3} (2\sqrt{3} - 3)\pi$	0.4860

Solutions for the smallest diameter circles into which n unit-diameter circles can be packed have been proved optimal for $n = 1$ through 10 (Kravitz 1967). The best known results are summarized in the following table, and the first few cases are illustrated above (Friedman).

The best-known packings of circles into a circular shape are shown in Fig. 54 for the first few cases.

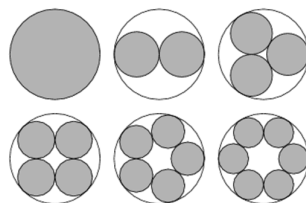


Fig. 54 Best known packing of circles into a circular shape

4.1.a.v.2 Improve performance through monitored material quantities

The second step [Fig. 55] conducted with the new geometry was solely looking at adding material where needed. This step compared 2 options, the one developed in this study that was parametrically exploring thousands of different configurations and picking the optimal material distribution and one that was the one adopted by the engineering firm that previously designed the sculpture. The step 2 in the optimization led to a reduction of the deflection of an additional 25%.



Fig. 55 Material distributed based on optimization VS more previous material distribution

4.1.a.v.3 Improve performance through controlled stance width

The third and last step of the optimization focused instead on the stance. As previously mentioned, this was a mix of structural and quantities optimization as can be understood from Fig. 56. This step lead to an additional 25% in reduction.

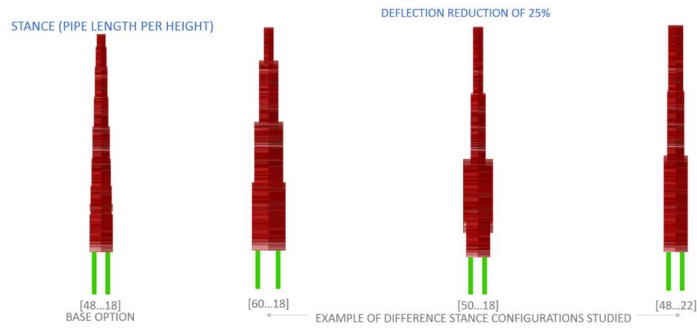


Fig. 56 Example of difference stance configurations in step 3

4.1.a.vi Connection design

To ensure full design and constructability of the structural Art piece particular attention has been given to the pipe to pipe connection as shown in Fig. 57. This check has been conducted in the above workflow but not highlighted to keep the focus on the optimization workflow rather than the design of a detail.

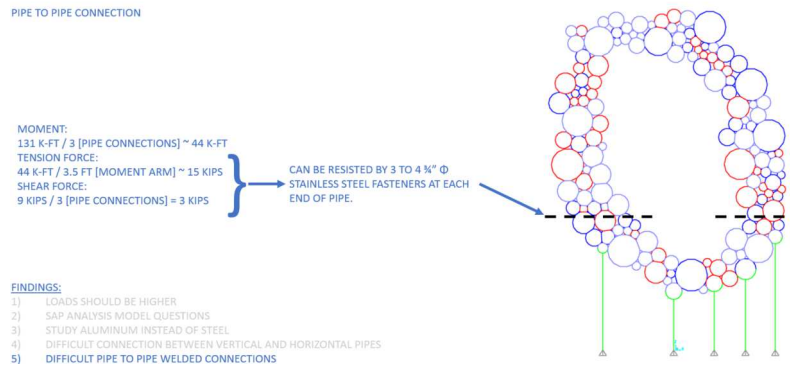


Fig. 57 Pipe to pipe connection

4.1.b Results

Summing up all the contribution we come up to a 90% total reduction bringing the deflection down to 10” as shown in Fig. 58. The above study is a unitize study hence we now need to convert it

in real numbers that would give us an indication of how the sculpture is performing.

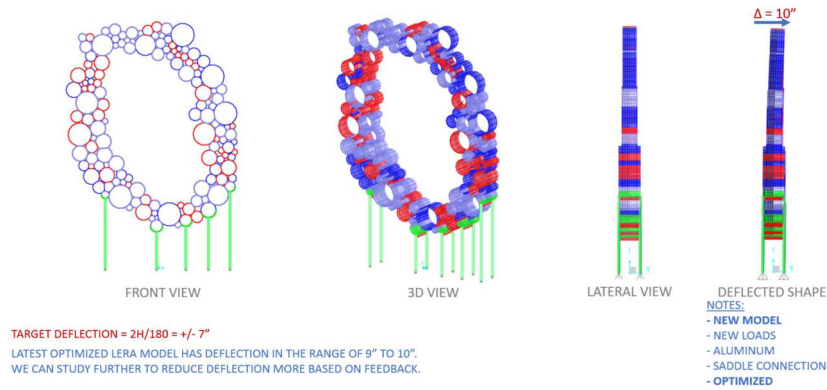


Fig. 58 Optimized solution performance

4.1.c Conclusions

Fig. 59 shows a comparison of the original structure VS the new optimized versions that achieved a 90% increment in performances thanks to the three-step optimization described in this summary. As can be understood by the visual, the vision of the Artist has been preserved while improving structural performance with custom optimization techniques.

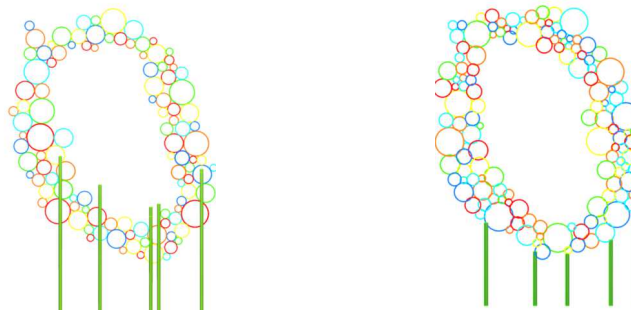


Fig. 59 Original and optimized solution

4.2 Second case study – Alyson Shotz

The second case study is a sculpture by artist Alyson Shotz that is now installed in the new Kimmel NYU building in Manhattan.

The sculpture as shown in fig. 60 was conceptualized by the artist as a three story high piece composed by a large number of double curved ribs hanging from the ceiling. Each rib is composed by a sandwich of a layer of aluminum enclosed within two layers of acrylic.

This case study will be presented in a one chapter format with no subchapters to stress out the seamless continuity of the process.

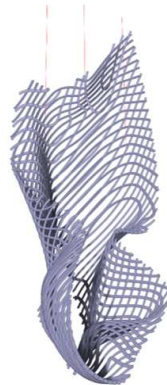


Fig. 60 3D view of Artist vision for the sculptural piece

The sculpture initially designed by another Engineering firm in Manhattan was facing anticipated deflection under dead load at the bottom of the piece in the range of 96” (almost 245cm). This point raised the concern of many in the room especially the artist who all of the sudden was facing a totally different design from her initial concept idea. The objective was then to optimize the structural system to preserve the initial design and vision provided by the artist.

Given that no techniques are available out of the box for such a complex problem, I came up with an algorithm developed by myself. The algorithm, conceptualized in Rhino 3D (a cad software for designers) and Grasshopper 3D (a visual programming environment for Rhino) is based on the idea of merging the design provided by the artist with structural performance that resemble the concept behind the research, join engineering with artistic design.

The first part of the algorithm I put together in Grasshopper, allowed me to reconstruct a 3D model of the meshed structure of the desired configuration for the sculptural piece. The reason behind this is highlighted in the next paragraph.

Having a mesh, in fact, allowed me to link the 3D model with FEA and in order to achieve that I utilized Karamba (an FEA component for Grasshopper 3D) that was directly tight to the above mentioned algorithmic definition. By doing so I was able to confirm the 96” of deflection anticipated by the structural engineering firm that was on the job.

One other thing that I noticed during this process was that the sculpture was very easily thrown off balance when the location of the hanging cables changed. In fact, even by moving one of the three cables at the top by only few centimeters would give a huge increase in the value for the deflection.

That was then my next step and the starting point of the optimization process. I connected Galapagos (an evolutionary solver for Grasshopper that allows to run genetic algorithms) with my algorithmic definition and in particular, I defined the location of the cables as variables and the deflection calculated by Karamba based on the dead load of the sculpture as the value to be minimized.

This process lead to a relatively complex Grasshopper definition shown in Fig. 62

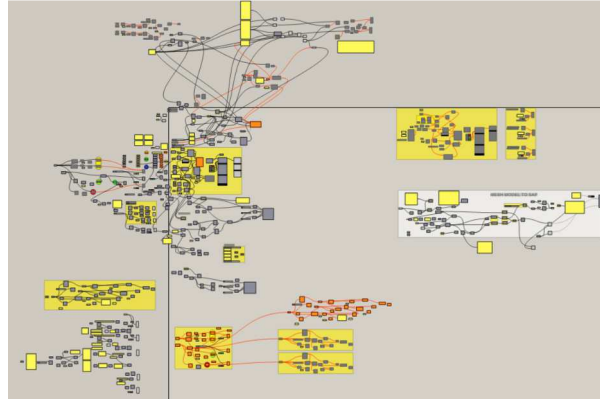


Fig 62 Grasshopper definition

After several attempts I collected a large number of data and picked the best performing cable configuration show later on in Fig. 63. This first step of the optimization allowed me to bring the value of the deflection down to 30". Going down 66" in deflection was already a huge achievement but still too much.

For that reason, I moved into the next step of the optimization process, the camber. By in fact, back calculating the results given from the structural analysis, I was able to write a custom algorithm in C# language to camber the piece and analyze it to verify that the deflected shape of the cambered piece would coincide to the initial model provided by the artist.

The way the algorithm works is to retrieve results for the deflection at each single point of the mesh along the entire structure and apply the displacement with a negative sign to each correspondent point of the mesh.

Fig. 63 shows the initial geometry provided by artist in blue, the cambered shape in red and the final relaxed shape in lavender. It's immediate to note that the relaxed shape is coincident with the desired shape.

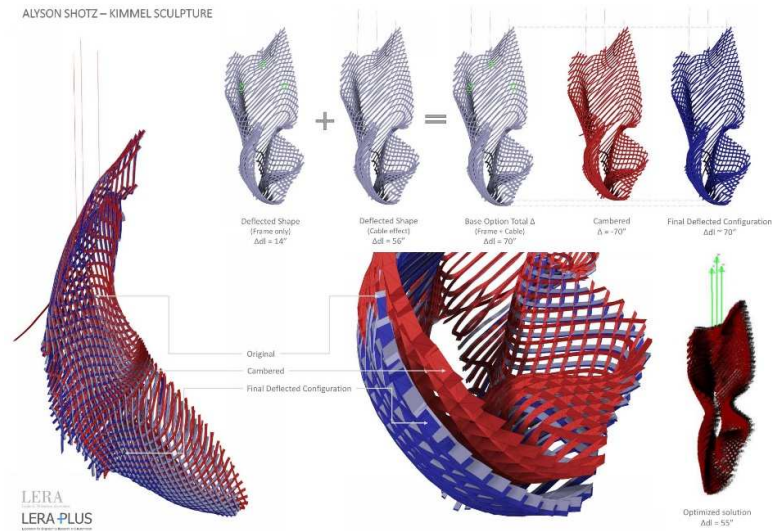


Fig. 63 Comparison between desired, cambered and relaxed shape.

At this point, all the issue relatively to the structural performance are solved but one more issue comes up on the fabrication side.

The above mentioned algorithm used to camber the piece, in fact, makes the fabrication process impossible. The ribs of the cambered shape are, in fact, not perpendicular to each other anymore and that for that reason the piece, if cut by the CNC machine, cannot be assembled as the ribs wouldn't fit together anymore.

This called for a third step in the optimization process and this time more for fabrication purposes. A third custom algorithm, in fact, was developed exclusively to deal with the perpendicularity of the ribs. The algorithm would get a single element of a mesh model, compute a small change in angle and compare this element against the other

in the desired perpendicular direction. The process would be performed on each single piece of the mesh and would stop when the 90 degrees angle configuration was reached.

The above mentioned algorithm has been developed discretizing the ribs into lines. This allowed to assume that each rib is in the form of: $Ax+By+C=0$

From this, the algorithm below can find and fix the perpendicular lines.

```
public class Perpendicular {
public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    int n=in.nextInt();
    ArrayList<Line> list_of_lines=new ArrayList<Line>();
    for(long i=0;i<n;i++){
        long a=in.nextLong();
        long b=in.nextLong();
        long c=in.nextLong();
        list_of_lines.add(new Line(a,b,c));
    }
    long p[]=new long[n];
    Arrays.fill(p,0);
    for(int i=0;i<n;i++){
        for(int j=1;j<n;j++){
            if(list_of_lines.get(i).slope()*list_of_lines.get(j).slope()
            )== -1){
                p[i]++;//num of perpendicular lines to i
                p[j]++;//num of perpendicular lines to j
            }
        }
    }
}
}
```

```
class Line{
public long a,b,c;
public Line(long a,long b,long c){
    this.a=a;
    this.b=b;
    this.c=c;
}
public double slope(){
    return a/b;
}
}
```

Findings from this case studies can be summarized as follow. First, the integrated design process between artist and engineer resulted to be very efficient in terms of design performance and coordination. Moreover, the process included, as noted in the last paragraph, even the fabrication side that concludes the design cycle.

The algorithms developed for this specific case study can be applied to a large number of sculptural pieces. The other advantages in the use of this algorithmic technique is that each algorithms can be easily integrated to cover specific tasks in case needed for any other sculptural piece.

The sculpture is now installed in the NYU building on 31st street and 1st avenue in Manhattan, New York.

Fig 64 to 66 are some shots of the fabrication process showing the size of the modules produces that were later on assembled in place as seen in Fig 67 to 69

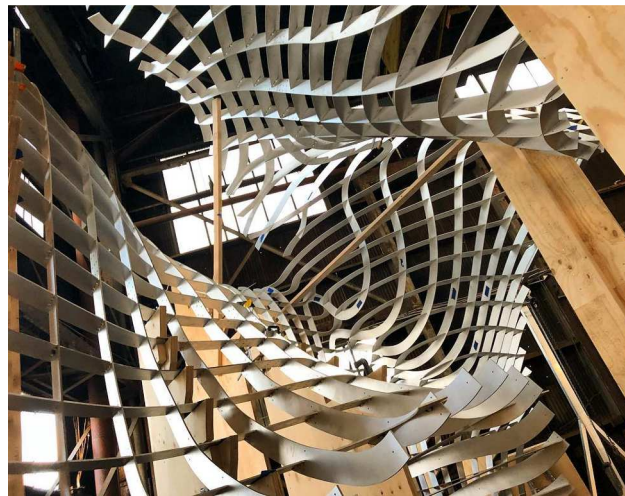


Fig 64 Fabricated module

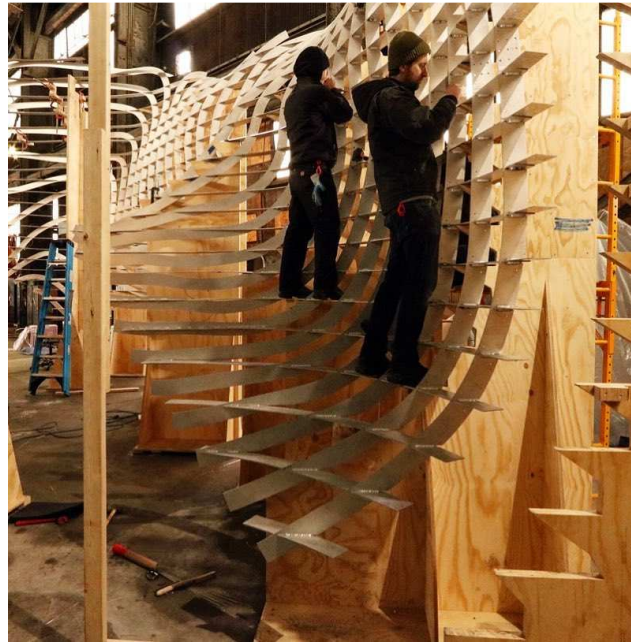


Fig 65 Fabricated module



Fig 66 Fabricated module



Fig 67 Installed Sculpture

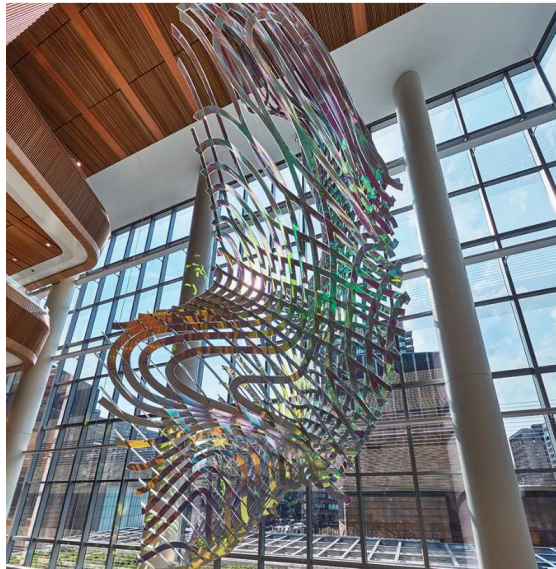


Fig 68 Installed Sculpture



Fig 69 Installed Sculpture

CHAPTER 5

FROM RESEARCH TO PRODUCT

During the last year of my PhD I wanted to merge all the above experiences under a product. The aim was to design an algorithmic driven sculpture that would be optimized for structural performance and completely conceptualized and driven by algorithms from conception (Schematic Design) to life (Fabrication).

5.1 The concept

Thanks to the advance in the computational field, sculptures today can embed different layers in their design. I will elaborate more on that:

- Artistic vision: As we all know Artists have a pretty specific vision for their pieces and most of the times little to no variations are permitted.
- Engineering thoughts: Structural design is the primordial need to any structure and sometimes does not walk hand in hand with the artistic vision.
- Computational thoughts: Computation is right in the middle of the Artistic vision and the Engineering thoughts. Computation helps to bridge that gap that goes from the 3D Artistic model to the current FEA software to the use of advanced algorithms to optimize the Artistic vision for structural performance without altering the Artistic vision.

The focus of this research is in that bridge. How do we pair conventional engineering tools with computational methods to preserve contemporary Artistic visions? The two case studies above have already partially answered this question and this chapter aims

on diving deep into it by exploring the full design process with a custom designed sculpture introduced in the next paragraphs.

The idea is to design a sculpture that would be easy to conceptualize but complex to analyze and fabricate with contemporary tools. The reason for that is because these are the future trends in the Art world. Artists are pushing the boundaries of complexity. They start with a concept that is easily realized in a 3D modeling environment but hides many Engineering and Fabrication challenges.

5.2 Preliminary design

The beginning of the research focused on two main basic Rhino commands:

Curves and polycurves;

Loft.

Following are the definitions provided by Mc Neel:

A Rhino curve is similar to a piece of wire. It can be straight or wiggled, and can be open or closed. A polycurve [Fig. 70] is several curve segments joined together end to end.

Rhino provides many tools for drawing curves. You can draw straight lines, polylines that consist of connected line segments, arcs, circles, polygons, ellipses, helices, and spirals.

You can also draw curves using curve control points and draw curves that pass through selected points.

Loft is a command in Rhino 3D that fits a surface through selected profile curves that define the surface shape.

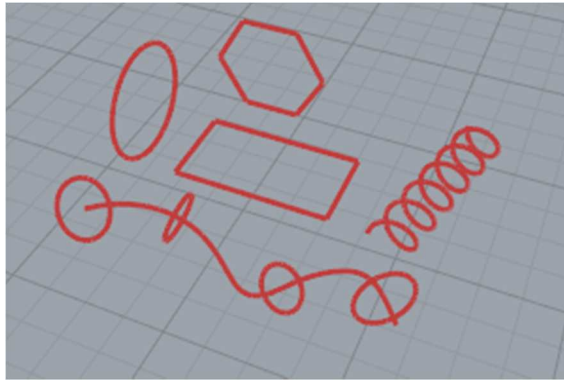


Fig 70 Rhino polycurve

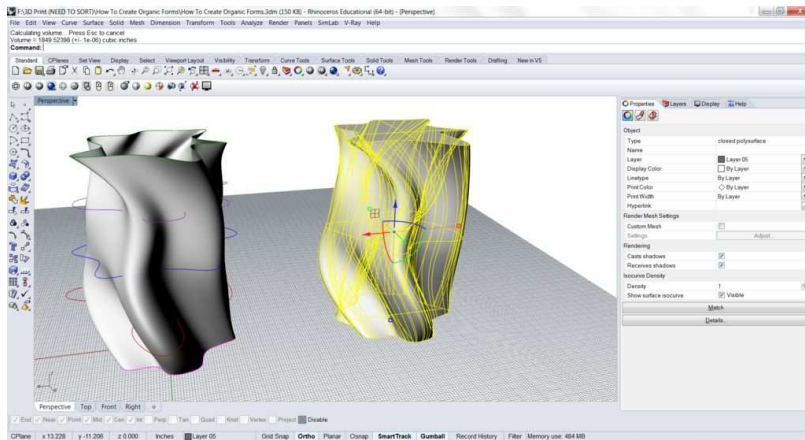


Fig 71 Rhino loft

Based on this approach I started by drawing 5 C-shaped [Fig 72] line segments spaced by 4cm.

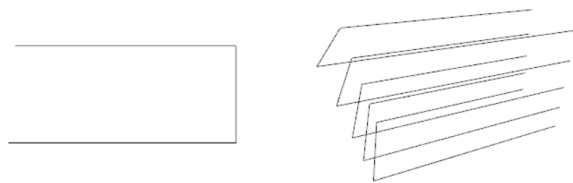


Fig 72 C shape lines in Rhino

Fig. 2D and 3D view of joined line segments to form a stack of c-shapes.

Later on, I referenced the c-shaped lines into grasshopper and subsequently divided them into segments of different lengths [fig. 73].

The points defined by this step are shown in the next figure and they will be driving the entire algorithmic definition and optimization process that will be described in the next paragraphs.

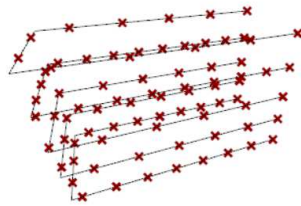


Fig 73 Subdivided C shape lines

The next step has been focused on creating curves [Fig 74] to give more movement to the backbone of the future sculpture.

The control points shown in the previous figure have then been fed into the Nurbs Curve command in Grasshopper to create 5 curves.

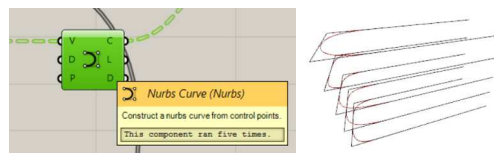


Fig 74 Lines to curves in Rhino

In order to make the constructing curves more dynamic a shift in the location of the points has been added [Fig 75]. Starting from two random function with the same slider as a driver, the X and Y

locations have been modified for each point. The points have been later on fed into the Nurbs Curve command and the resulting curves are shown in green in the next image.

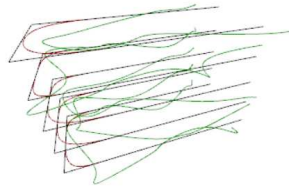


Fig 75 Shifting of points to add dynamic factor

An important player in this process is the slider highlighted in the Fig 76. The slider named “A” is the one that controls the movement of the points in x and y direction hence, as mentioned before, that will later on drive the optimization process.

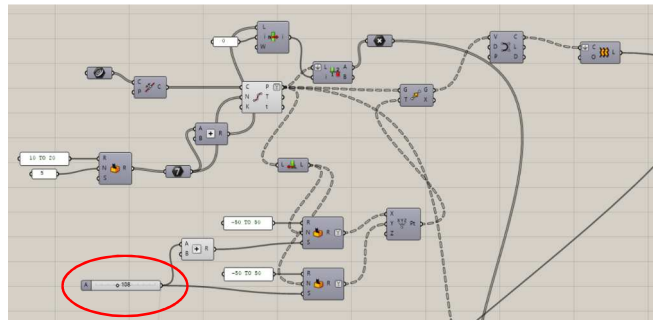


Fig 76 Grasshopper definition and Slider A

The next step of this process is creating the loft [Fig. 77] through these curves that will compose the fabric representation of the sculpture.

In order to do that the curves we just created are fed into the loft component in Grasshopper and the result is shown in the next figure.

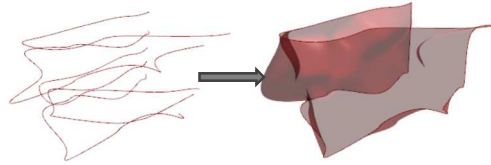


Fig 77 Curves to loft

Not many calculations were required at this stage. The approximate dimensions of the sculpture are 32cm x 18cm x 18cm. This is intended to be a mock up and once realized to serve as the starting point for a full scale piece.

3D printing is the preferred fabrication method to further stress out the concept of digital design realized into digital fabrication. The material is supplied by the Proto-pasta company. To understand its mechanical properties we need to back up and look at mechanical properties for other more common materials for 3D printing.

5.3 Materials for 3d printing

Simplify 3D provides a very useful table as a summary of the most common and tested materials for 3D printing purposes [Fig 78].

	ABS	Flexibile	PLA	HIPS	PETG	Nylon	Carbon Fiber Filled	ASA	Polycarbonate	Polypropylene	Metal Filled	Wood Filled	PVA
Compare Selected	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ultimate Strength	40 MPa	100 MPa	75 MPa	100 MPa	100 MPa	100 MPa	100 MPa	100 MPa	100 MPa	100 MPa	100 MPa	100 MPa	100 MPa
Stiffness	2.0 GPa	1.5 GPa	2.5 GPa	2.0 GPa	2.0 GPa	2.5 GPa	2.5 GPa	2.5 GPa	2.5 GPa	2.0 GPa	2.0 GPa	2.0 GPa	2.0 GPa
Durability	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000
Minimum Service Temperature	20°C	20-20°C	20°C	100°C	120°C	20-120°C	20°C	25°C	120°C	120°C	20°C	20°C	20°C
Crystallinity/Thermal Expansion	50%	10%	5%	5%	5%	5%	5%	5%	5%	5%	5%	5%	5%
Recycle	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Waterproof	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Printability	High	High	High	High	High	High	High	High	High	High	High	High	High
Printer Temperature	230-250°C	230-250°C	230-250°C	230-250°C	230-250°C	230-250°C	230-250°C	230-250°C	230-250°C	230-250°C	230-250°C	230-250°C	230-250°C
Bed Temperature	60-100°C	60-100°C	60-100°C	60-100°C	60-100°C	60-100°C	60-100°C	60-100°C	60-100°C	60-100°C	60-100°C	60-100°C	60-100°C
Chamber Heat	Required	Required	Required	Required	Required	Required	Required	Required	Required	Required	Required	Required	Required

process. The first step is then to mesh the sculptural piece represented in the Fig. 79.

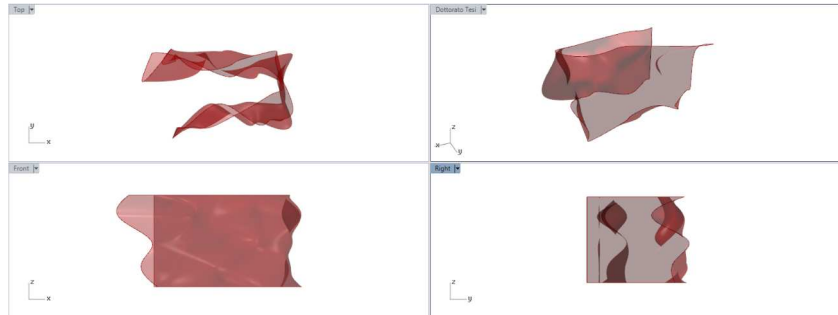


Fig 79 Sculpture

The loft created in the previous step gets fed into the mesh component [Fig. 80] in Grasshopper. Keeping U and V subdivision to 20 guarantees a fast speed in the creation of the analysis results explored later on.

H instead is a parameter that takes into account for overhanging and can be set to a limit after which the overhanging get approximated to a vertical or horizontal based on which one is closer to the actual angle.

Q is a Boolean parameter that if enabled equalized the spans across which U and V are created. In the case of this sculptural piece, given its double curvature Q has been set to FALSE.

The output quadrangular mesh is later passed into the “triangulate mesh” component. The mesh is now ready to be fed into the Karamba components to start the structural design process.

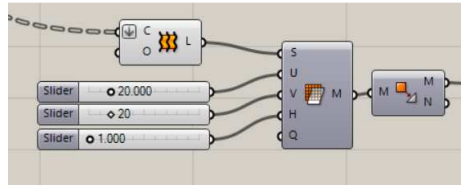


Fig 80 Mesh component in Grasshopper

5.5 Finite element analysis setup

Once the triangular mesh is formed we can start the analysis process using Karamba (FEA plug in for Grasshopper). The first step illustrated in the next figure is explained below:

The mesh is fed into the mesh karamba component [Fig. 81] and it gets named (“shell” in this case). The ID allows us to create the cross section component and assign thickness to the shell. In this particular case I am using 2mm as thickness.

As the last step of the analysis set up I am assigning gravity loads (i.e. self-weight of the sculpture). I will not be considering wind loads in this case because the sculpture is designed for indoor.

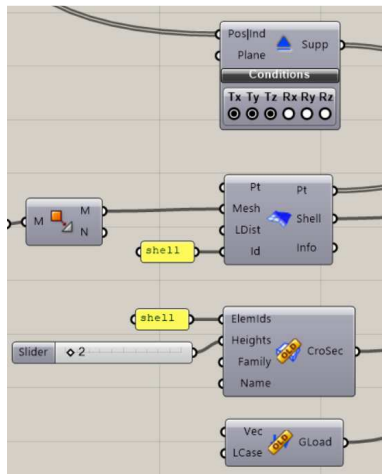


Fig 81 Karamba components attached to Grasshopper definition of sculpture

In the second step all the above mentioned values get fed into the “Assemble Model” component that, as the name suggests, assembles all the properties defined above together. Following this step there is the “Analyze” component that runs the analysis and outputs results.

As shown in Fig. 82, the results can then be processed and visualized using the shell view component. In the next figure an example of visualization of data behind the utilization of the sculpture is illustrated. The following figure shows the results for utilization visualized on the shell of the sculpture.

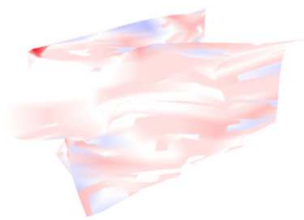
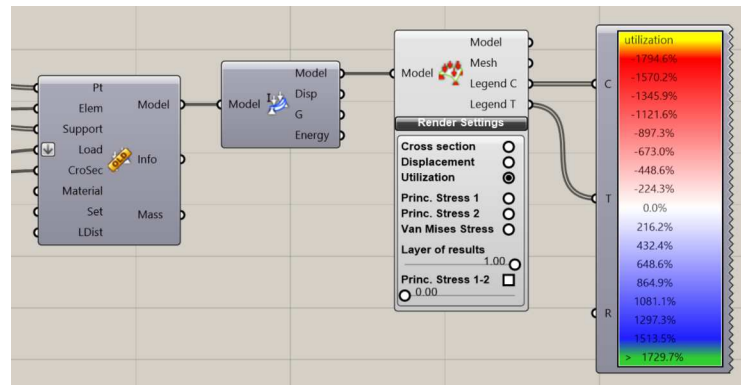


Fig 82 Karamba results visualized in Rhino

If we look at the stress distribution [Fig. 83] over the sculpture we notice that the stresses are far close from the ultimate strength of the material. The max stress is in fact equal to 2.7 ksi as show in the next figure.



Fig 83 Stress distribution over the sculpture

If we look at the deflection value coming from the “Analyze” parameter and presented in the form of worst displacement [Fig. 84] along the sculpture (Dmax), the value in this particular case results to be $D_{max} = 76.9$ mm as shown in the following figure.

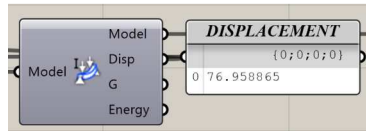


Fig 84 Displacements out of Karamba FEA

The problem seems to be clearly a deflection problem. D_{max} will then be the parameter that we will use to drive the optimization process.

Since we are building a mockup of the sculpture we will focus more on the optimization process than the value of deflection to hit. As a rule of thumbs we will then assume that we need to meet at

5.6 Setting up the optimization process

For the optimization process the Grasshopper plug in Galapagos [Fig. 85] has been implemented in the optimization process.

Galapagos inputs for the Genome and Fitness are assigned respectively to:

Genome: the parameter A introduced and described in the “Preliminary Design” chapter.

Fitness: the worst case deflection.

This will allow to explore different designs through the parameter A and evaluate the deflection live as the design is altered.

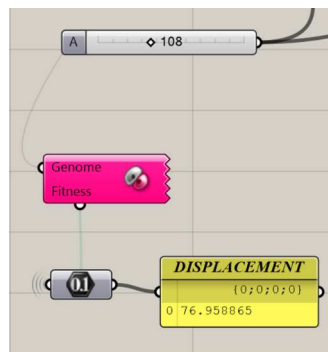


Fig 85 Galapagos plug in for Grasshopper

The aim behind the optimization is to minimize the deflection of the sculpture under self-weight by changing the shape using the parameters introduced at the beginning of the chapter.

It's interesting to note that the first iterations out of the optimization [Fig. 86] ran with Galapagos lead to very different designs but cannot converge to a solution [Fig. 87].

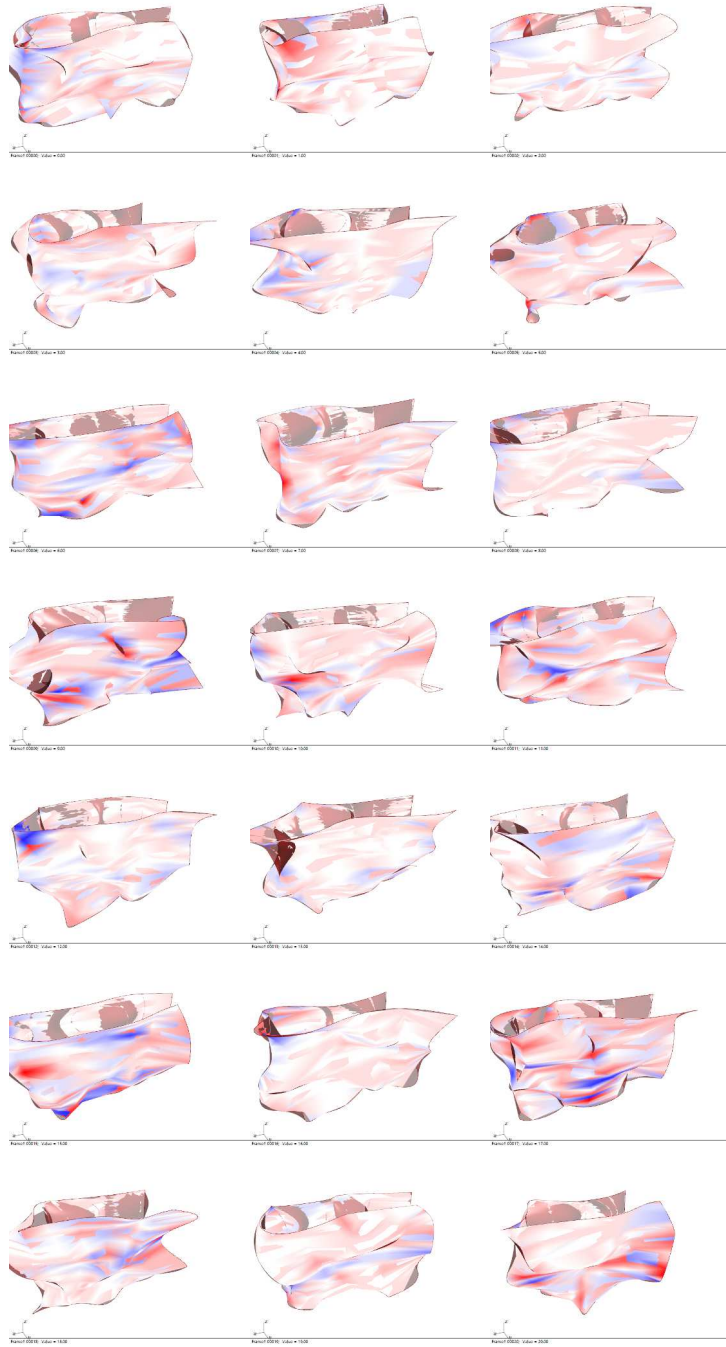


Fig 86 Iterations out of Galapagos

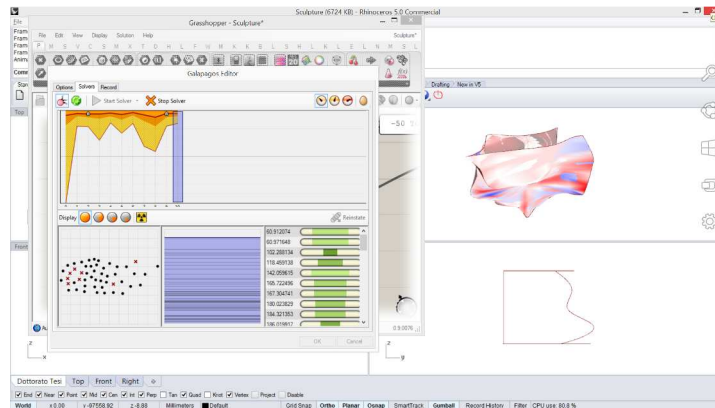


Fig 87 The solution does not converge

This leads to the second test where we introduce the variable of the thickness of the walls. Even in this case, in this second run, as shown in Fig. 88 below, the solution still cannot converge.

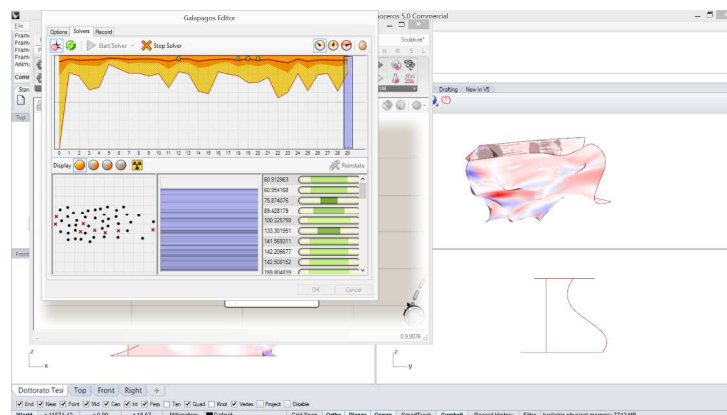
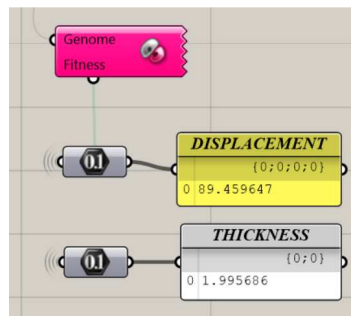


Fig 88 The solution still does not converge

The third attempt was to use a Simulated Annealing approach that led to the same conclusion, no convergence as shown in Fig 89.

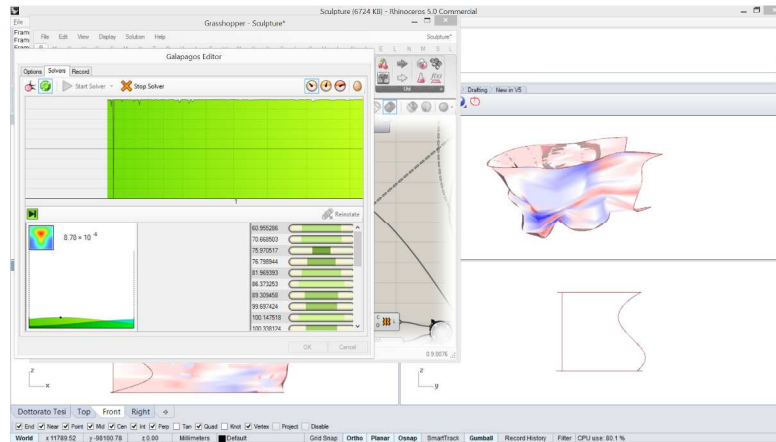


Fig. 89 Galapagos third run of iterations with no convergence

What we learned through this optimization was some of the design were failing for strength and the values of the deflection that were huge in that case were throwing the algorithm far out from points of minimum.

The output of this process led to the conclusion that this design was in need of a multi-objective optimization. Translated in engineering words, deflection seems to be the problem for the specific design (starting point of optimization) but when considering a wide range of design solutions that is not always true.

Octopus, a multi objective optimization tool for Grasshopper was then implemented as the next step of the research process.

Before diving into the actual design process with octopus is due defining multi-objective optimization.

5.7 Multi objective optimization

5.7.a Background

The “best” has been an historical standard to achieve differently in different fields. Economics is the field where multi-objective thinking arguably originated. Specifically, one of the first individuals to consider such methods was F.Y Edgeworth [Fig. 90].



Fig 90 F.Y. Edgeworth

5.7.b Basics of MOO

A generic multi-objective design optimization problem may be formulated as in the below equation:

$$\begin{aligned} \min \mathbf{J}(\mathbf{x}, \mathbf{p}) & \quad \text{where } \mathbf{J} = [J_1(\mathbf{x}) \quad \cdots \quad J_z(\mathbf{x})]^T \\ \text{s.t. } \mathbf{g}(\mathbf{x}, \mathbf{p}) \leq 0 & \quad \mathbf{x} = [x_1 \quad \cdots \quad x_i \quad \cdots \quad x_n]^T \\ \mathbf{h}(\mathbf{x}, \mathbf{p}) = 0 & \quad \mathbf{g} = [g_1(\mathbf{x}) \quad \cdots \quad g_{m_1}(\mathbf{x})]^T \\ x_{i, LB} \leq x_i \leq x_{i, UB} \quad (i = 1, \dots, n) & \quad \mathbf{h} = [h_1(\mathbf{x}) \quad \cdots \quad h_{m_2}(\mathbf{x})]^T \\ \mathbf{x} \in S & \end{aligned}$$

Here, J is a column vector of z objectives, whereby $i \in J \in \mathbb{R}$. The individual objectives are dependent on a vector x of n design variables as well as a vector of fixed parameters, p . The individual design variables are assumed continuous and can be changed independently by a designer within upper and lower bounds, x_{UB} and x_{LB} , respectively. In order for a particular design x to be in the feasible domain S , both a vector of m_1 inequality constraints, g , and m_2 equality constraints, h , have to be satisfied. The problem is to minimize – simultaneously – all elements of the objective vector. A number of names have been given to this type of problem: vector minimization, multi-criteria optimization, multi-attribute maximization and so forth. For the most part these are synonymous and we will refer collectively to this class of problems as multi-objective optimization (MOO) problems.

All design and engineering activity is fundamentally multi-objective in nature because of the existence of inherent tensions CJK-OSM3, 2004, Kanazawa between the four main objectives in product or system design [Fig. 91]: performance, cost, schedule and risk (Maier & Rechtin, 2000). With schedule and risk levels (e.g. probability of failure of a component) fixed, better performance can generally only be achieved by increasing cost. Pulling along one of the dimensions in the diagram of Fig.1 generally requires compromises along the other dimensions.

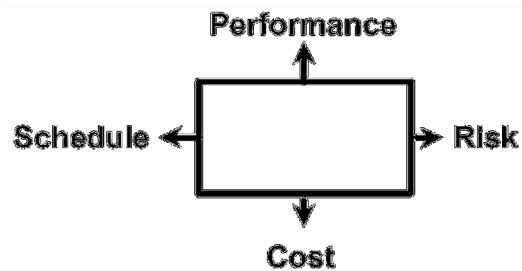


Fig. 91 Tensions during system design

There is general consensus that multi-objective optimization methods can be broadly decomposed into two categories: Scalarization approaches and Pareto approaches. While different names are used for these categories, the fundamental discriminator is always the same. In the first group of methods the multiobjective problem is solved by translating it back to a single (or a series of) single objective, scalar problems.

Pareto methods instead attempt to find a set of efficient solutions, x^*j , such that the objective vectors corresponding to those solutions are non-dominated in z-dimensional objective space.

In this research we will implement the Pareto method hence describe it below.

5.7.c Pareto method

Dominance (for maximization) is defined as follows: Let J_1 and $J_2 \in R$ be two feasible objective vectors.

Then J_1 dominates J_2 (weakly) if

$$\mathbf{J}^1 \geq \mathbf{J}^2 \quad \text{and} \quad \mathbf{J}^1 \neq \mathbf{J}^2$$

For strong dominance all elements of J_1 would have to be greater than the corresponding elements of J_2 .

Based on the notion of dominance, the simplest approach is a combination of design space exploration and dominance (Pareto) filtering. This has been applied to our numerical example and is shown in Fig. 92. A comparison with Fig 93 shows a much more complete approximation of the Pareto front of non-dominated solutions. This is appealing, but raises two important points:

1. A comprehensive or full-factorial evaluation of the design space is often impossible due to the n-dimensionality of the design vector, x , and the required computational effort for obtaining J , g and h .
2. The solutions obtained in this way are mere approximations of the Pareto Front. More precisely, the points only satisfy non-dominance.

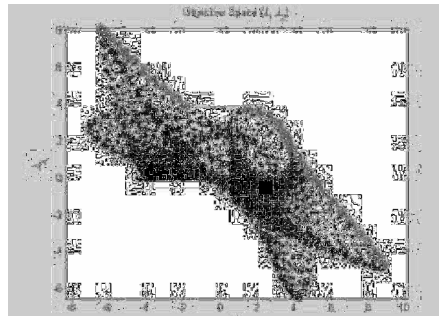


Fig 92 Approximation to Pareto Front via dominance filtering

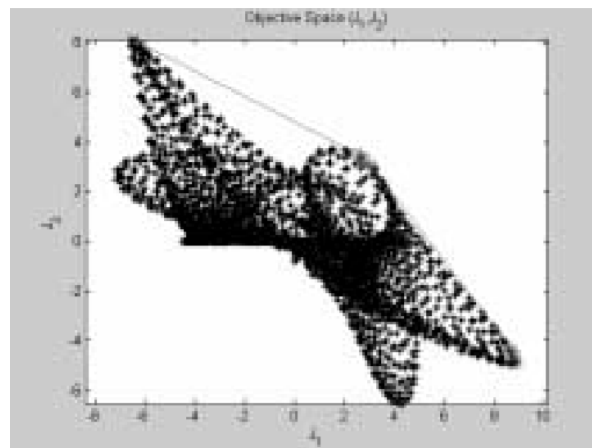


Fig 93 Optimal solutions from Weighted Sum Approach

5.7.d Octopus plug-in for Grasshopper

Once we defined the optimization method that we will implement for this optimization by using the Octopus plug-in we can proceed in describing the application to the design problem.

First of all let's remark that the main benefit for our design problem in transitioning to a multi-objective optimization is the ability to track both deflections and stresses while changing the sculpture geometry and thickness of materials.

The Octopus component is wired uniquely within the Grasshopper environment [Fig. 94]. On the left, Phen stands for phenotype. The phenotype is wired to the resulting geometry produced with the calculations. The G on the bottom right stands for genomes. It is wired to all the genomes used for the source of evaluation. The O on the bottom right stands for the objectives. The data components that you labeled are wired there. Phenot on the right stands for phenotype. That's the resulting solution.



Fig 94 Octopus plug in for Grasshopper

When double clicking the component, the interface of Octopus opens up [Fig. 95]. The plug-in represents the solutions within a 3D environment by using the objectives as axis. The more a solution satisfies an objective, the farther its shown on the axis. The left portion of the screen changes the representation of the results within

Moreover, Octopus not only allows to see the 3D representation of the optimization process but also allows for output the results in an excel format. This allows to reformat and parse data from the optimization process. All in all, it allows for a better understanding of the optimization process and more control over the data with respect to Galapagos that does not allow for any output of the solutions.

5.8 Setting up MOO optimization

Setting up the optimization with Octopus required to migrate to a gene pool to control the parameters of the design. Those parameters as can be seen in the Fig. 97 where connected as input to the “G” filed in octopus and will be the gene of pools for the process. The objective functions of the optimization, as previously defined, are the Displacement and Thickness and are then fed in the “O” parameter as objective of the process.

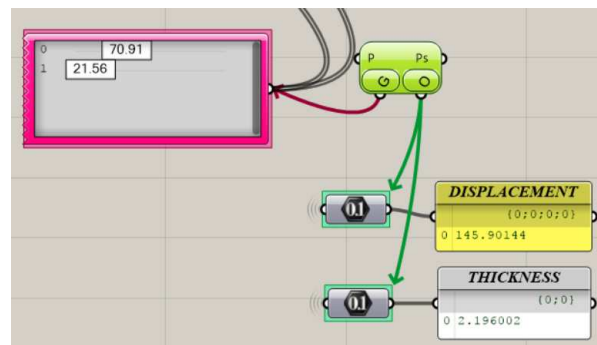


Fig 97 Octopus parameters

Exploration with Octopus was also faster than Galapagos and convergence was reached within 300 iterations.

The best 20 solutions were then picked and are presented in Fig 98 in the form of a series of designs.

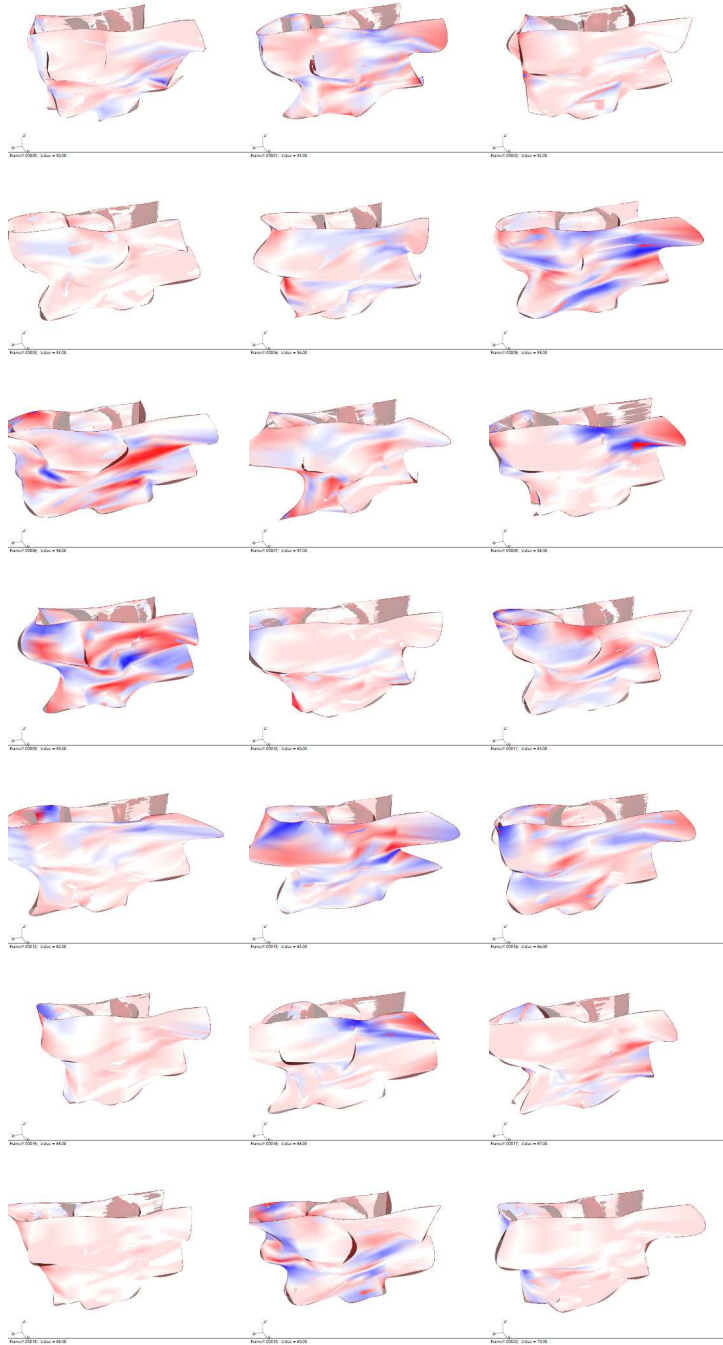


Fig 98 Best performing Octopus solutions

Amongst these, the 5th best solution [Fig 99] was the preferred one and the results from the optimized solution are listed below:

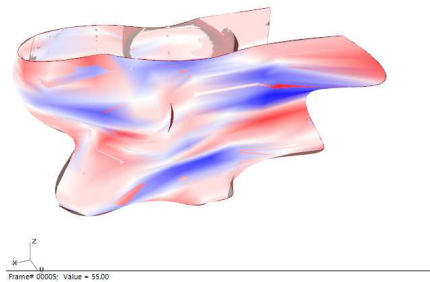


Fig 99 Option 5

Deflection: 0.19 cm = 0.074 inches

Max stress: 10 Mpa = 1.5 Ksi

Thickness: 0.4 cm = 0.15 inches

The reason behind the choice of the 5th solution were purely about Artistic vision. All the 20 solutions are in fact within 5% proximity in terms of results.

This leads to flexibility and more options for the Artist that is still choosing his design vision but with engineering thoughts incorporated within this time.

5.9 Fabrication

An additional perk about this process is that the design is ready to be fabricated since the 3D print used for the optimization is a mesh ready to be fed into a 3D printer machine.

In order to achieve that we paired with Gcreate in Brooklyn, a digital fabrication shop that took over the challenge and brought the sculpture to life. The fabrication process for the mockup measuring 32cm x 18cm x 18cm took close to a week for fabrication due to the fact that the material used needs more time to harden than a more common PLA. The presence of iron in fact, makes it harder for the PLA to solidify and bond with the next layer.

This was learned the hard way. The sculpture was in fact first fabricated at regular speed and after 2 days of work collapsed on itself looking like a melting iceberg. Unfortunately Gcreate did not save evidence of the attempt. The second attempt was instead successful and the final product after sanding and immersing the sculpture in a batch of acid to achieve a corten steel look is pictured in the Fig. 100.





Fig. 100 3D printed sculpture

CHAPTER 6

CONCLUSIONS, FURTHER DEVELOPMENT AND POTENTIAL APPLICATIONS

The Research aims on redefining the way we design and build by integrating three practices: Engineering, Art and Fabrication.

The engineering process is the bridge amongst the other two practices. Engineering is in fact, in charge of the design process. This process starts from the handoff from the Artist that shares his/her conceptual vision. This research stressed out how thought he use of computational design Engineering can play that key role without impacting on the artistic vision. In better words, we have demonstrated how we can optimize while preserving the artistic vision.

In downstream the Fabrication team is in charge of bringing the vision to life. Thanks to the digital fabrication techniques the fabrication process can today be highly driven by the Engineer. We have in fact shown how the use of subtractive fabrication (CNC machines) and additive fabrication (3D printers) in our case studies have allowed the Engineer to take charge of this last step of the design process.

The mock up realized for the third year is definitely a great physical representation of the above statement.

The research behind an experimental material like the composite Iron paired with the contemporary techniques of computational design have allowed to have a completely streamlined process of design.

This wants to be just the beginning of a bigger change in the industry. Together with Autodesk and in particular as member of the Autodesk Executive Members of North America we are currently working on a white Paper that will be presented during Autodesk University 2019 that aims towards presenting solutions for integrated Engineering and Fabrication processes. In particular the paper will highlight some of the points of this research in the deliverable to fabricators of digital models created by engineers. This to reduce the amount of back and forth and coordination and to increase profits on both sides.

Moreover, the process of optimization with Octopus is currently being implemented for a confidential application on a Sculpture to be installed in 2021 in North Carolina. The sculpture is similar to the one studied in the second case study in this research but this time will be hang from only one single cable so we are introducing a physic engine and nonlinear analysis to study the stability of the piece for long term displacement.

The digital fabrication shop in Brooklyn that we paired with for this research, Gcreate, and the Vancouver based material provider Proto-Pasta, are advancing the conversation with us with the aim of working together to scale up the sculpture mock up developed during the third year of this Research. The first step, more structural than anything else is to develop connection details that would allow the sculpture to sustain gravity and lateral loads. Next step will be to create hollow structural walls with a relatively thin thickness to keep the costs down. Material is in fact still pretty expensive and a dedicated 3D printer will probably have to be custom built by Gcreate to build the modules that will compose the final sculpture. The size for the 3D printer is in fact, as of now, controlling the

maximum dimensions for the modules (hence, the final sizes of the sculpture).

The hope is to finalize the design by July 2020, after which, we'll move into the next phase of looking for sponsors to create the first 3D printed large scale sculpture in composite material!

BIBLIOGRAPHY

Computational Design Thinking: Computation Design Thinking.
Achim Menges, Sean Ahlquist

Robotic Futures P. Juan, A. Menges, N. Leach

Material Computation. Menges, A.

Emergent Technologies and Design. M. Hensel, A. Menges, M.
Weinstock

Computational Design Methods and Technologies: Applications in
CAD, CAM and CAE Education. Ning Gu (University of
Newcastle, Australia) and Xiangyu Wang (Curtin University,
Australia)

Processing: Creative Coding and Computational Art
(Foundation) 1st Corrected ed., Corr. 2nd printing Edition. Ira
Greenberg

Dueck, G. and Scheuer, T. "Threshold Accepting: A General
Purpose Optimization Algorithm Appearing Superior to Simulated
Annealing." J. Comp. Phys. 90, 161-175, 1990.

Ingber, L. "Simulated Annealing: Practice Versus Theory." Math.
Comput. Modelling 18, 29-57, 1993.

Kirkpatrick, S.; Gelatt, C. D.; and Vecchi, M. P. "Optimization by
Simulated Annealing." Science 220, 671-680, 1983.

Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M.; Teller, A. H.;
and Teller, E. "Equation of State Calculations by Fast Computing
Machines." J. Chem. Phys. 21, 1087-1092, 1953.

Otten, R. H. J. M. and van Ginneken, L. P. P. P. The Annealing Algorithm. Boston, MA: Kluwer, 1989.

[Cvijovic1995]D. Cvijovic and J. Klinowski, "Taboo Search: An Approach to the Multiple Minima Problem", Science, 1995.

[Fiechter1994]C–N. Fiechter, "A parallel tabu search algorithm for large traveling salesman problems", Discrete Applied Mathematics, 1994.

[Gendreau2003]M. Gendreau, "2: An Introduction to Tabu Search", in Handbook of Metaheuristics, pages 37–54, Springer, 2003.

[Glover1977]F. Glover, "Heuristics for integer programming using surrogate constraints", Decision Sciences, 1977.

[Glover1986]F. Glover, "Future paths for integer programming and links to artificial intelligence", Computers and Operations Research, 1986.

[Glover1986a]F. Glover and C. McMillan, "The general employee scheduling problem: an integration of MS and AI", Computers and Operations Research, 1986.

[Glover1989]F. Glover, "Tabu Search – Part I", ORSA Journal on Computing, 1989.

[Glover1990]F. Glover, "Tabu Search – Part II", ORSA Journal on Computing, 1990.

[Glover1990a]F. Glover, "Tabu Search: A Tutorial", Interfaces, 1990.

[Glover1993]F. Glover and E. Taillard, "A user's guide to tabu search", Annals of Operations Research, 1993.

[Glover1998]F. W. Glover and M. Laguna, "Tabu Search", Springer, 1998.

[Knox1994]J. Knox, "Tabu search performance on the symmetric traveling salesman problem", Computers & Operations Research, 1994.

A case study of 3D printed PLA and its mechanical properties

S.Aravind Raja, E.Muthukumarab, Jayakrishna.Kc*

MULTIOBJECTIVE OPTIMIZATION: HISTORY AND PROMISE
Olivier L. DE WECK1 * 1 Dept. of Aeronautics & Astronautics, Engineering Systems Division, Massachusetts Institute of Technology

Octopus plug-in MULTI-OBJECTIVE GENETIC SOLVER, john davis

<https://jorgefontan.com/architectural-design-phases/>

<https://www.pinterest.com/pin/577375614708887958/?lp=true>

<https://archsmarter.com/what-is-dynamo-revit/>

https://en.wikipedia.org/wiki/Grasshopper_3D

https://en.wikipedia.org/wiki/Rhinoceros_3D

https://en.wikipedia.org/wiki/Autodesk_Revit

<https://toddschneider.com/posts/traveling-salesman-with-simulated-annealing-r-and-shiny/#salesman-app>

https://www.researchgate.net/figure/Simulated-Annealing-optimization-of-a-one-dimensional-objective-function_fig1_308786233

https://www.researchgate.net/figure/Flowchart-of-simulated-annealing-algorithm_fig7_298209081

https://www.researchgate.net/figure/Flowchart-of-tabu-search-algorithm_fig1_320508257

Tabu Search Algorithm (good gif at link below)

<https://www.globalsoftwaresupport.com/tabu-search-algorithm/>

http://www.timvoelcker.de/genetic_algorithm.html

<https://www.differencebetween.com/difference-between-local-and-vs-global-maximum/>

https://www.researchgate.net/figure/Flow-chart-of-genetic-algorithm_fig2_265208874

<https://www.rhino3d.com/6/new/grasshopper>

<https://www.rhino3d.com/6/new/grasshopper>

<https://www.rhino3d.com/6/new/grasshopper>

<https://www.autodesk.com/products/revit/features>

<https://www.karamba3d.com/>

www.fiverr.com

<https://didgrasshopper.wordpress.com/2011/08/19/galapagos-example/>

<https://www.instructables.com/id/How-to-Create-3D-Printed-Organic-Forms-and-Vases/>

http://docs.mcneel.com/rhino/5/help/en-us/seealso/sak_curve.htm

<https://www.simplify3d.com/support/materials-guide/properties-table/>