GAZİ ÜNİVERSİTESİ

1926

IMPLEMENTATION OF TOPOLOGY OPTIMIZATION IN ARCHITECTURAL FORM-MAKING DECISIONS: A STUDY ON TREE-LIKE STRUCTURES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
GAZİ UNIVERSITY

BY
Mariam Hazem Helmi SELMI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ARCHITECTURE

February 2023

ETHICAL STATEMENT

I hereby declare that in this thesis study I prepared in accordance with thesis writing rules of Gazi University Graduate School of Natural and Applied Sciences;

- All data, information and documents presented in this thesis have been obtained within the scope of academic rules and ethical conduct,
- All information, documents, assessments and results have been presented in accordance with scientific ethical conduct and moral rules,
- All material used in this thesis that are not original to this work have been fully cited and referenced,
- No change has been made in the data used,
- The work presented in this thesis is original,

or else, I admit all loss of rights to be incurred against me.

Mariam Hazem Helmi SELMI

06/02/2023

IMPLEMENTATION OF TOPOLOGY OPTIMIZATION IN ARCHITECTURAL FORM-MAKING DECISIONS: A STUDY ON TREE-LIKE STRUCTURES

(M. Sc. Thesis)

Mariam Hazem Helmi SELMI

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

February 2023

ABSTRACT

The architectural field benefits from advances in computational tools and the introduction of mathematical and numerical methods. Topology optimization (TO) is one method that has been well-known and implemented in different engineering subjects. It is concerned with finding the best material distribution that achieves structural efficiency with the least possible material usage. TO is relatively newly introduced into architecture, creating the potential to produce novel architectural forms. The use of topology optimization in architecture was investigated through a two-stage case study. In the first stage, three topology optimization tools (Millipede, tOpos, and Ameba) were analyzed, and an evaluation was conducted based on the computation time, volume fraction, and structural performance. A column and a cube representing architectural space are used to produce 5 different design solutions using the 3 different plugins. Based on the collected data, tOpos was found to be the most ideal program for the architectural design phase. In the second stage, a study was conducted on nature-inspired designs, which have the same purpose as Topology optimization. The primary objective of the topology optimization process is to decrease material consumption, which is also seen in nature-inspired designs. At this phase, the results achieved by altering design parameters are compared. Different numbers of columns working to support the same slab with the same support area were used and obtained results were evaluated according to form and structural performance. Increasing the number of columns as tree branches gave different results. As a result, this thesis study, which comparatively evaluates topology optimization for its use in different architectural designs, represents an applied example for designers.

Science Code	:	80115
Key Words	:	Topology optimization, Architectural design, Tree-like structures, Grasshopper plugins
Page Number	:	95
Supervisor	:	Assoc. Prof. Zeynep Yeşim İLERİSOY

MİMARİ BİÇİMLENDİRME KARARLARINDA TOPOLOJİ OPTİMİZASYONU UYGULAMASI; AĞAÇ BENZERİ YAPILAR ÜZERİNE BİR ÇALIŞMA

(Yüksek Lisans Tezi)

Mariam Hazem Helmi SELMI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Şubat 2023

ÖZET

Mimari alanı, hesaplama araçlarındaki gelişmelerden ve matematiksel ve sayısal yöntemlerin tanıtılmasından yararlanır. Topoloji optimizasyonu (TO), farklı mühendislik konularında iyi bilinen ve uygulanan bir yöntemdir. Mümkün olan en az malzeme kullanımıyla yapısal verimliliği sağlayan en iyi malzeme dağılımını bulmakla ilgilenir. TO, mimariye nispeten yeni girmiş olup, yeni mimari formlar üretme potansiyeli yaratmaktadır. Mimaride topoloji optimizasyonu kullanımını araştırmak için iki aşamalı bir alan çalışması yapılmıştır. İlk aşamada, TO için kullanılan 3 programlar (Millipede, tOpos, ve Ameba) analiz edilmiş ve mimari tasarımda mekânı temsil eden bir küp ve bir kolon üzerinden farklı biçimlenme türlerine göre 3 farklı programla 5 tip tasarım çözümü yapılarak, optimizasyon süresi, hacim fraksiyonu ve yapısal performans parametreleri altında bir değerlendirme yapılmıştır. Bu veriler üzerinden mimari tasarım aşamasında tOpos en ideal program olarak sonuç verilmiştir. İkinci aşamada ise Topoloji optimizasyonu ile aynı amacı güden doğadan ilham alan tasarımlara yönelik araştırması yapılmıştır. Topoloji optimizasyon sürecinin ana hedefi malzeme tüketimini en aza indirmek olup, bu hedef aynı amacı paylaşan doğadan ilham alan tasarımlarda da görülmektedir. Bu noktada tasarım kararlarının değişmesi ile elde edilen ürünlerin karşılaştırılması yapılmıştır. Aynı plak elemanın aynı mesnetlenme alana sahip farklı sayıda kolonlarla desteklenmesi üzerine yapılan çalışmada elde edilen sonuçlar biçimsel ve yapısal performans parametrelerine göre değerlendirilmiştir. Ağaç dalları niteliğinde kolon sayılarının artması farklı sonuçlar vermiştir. Sonuç olarak farklı mimari tasarımda kullanımı için topoloji optimizasyonunu karşılaştırmalı olarak değerlendiren bu tez çalışması tasarımcılar için uygulamalı bir örnek temsil etmektedir.

Bilim Kodu	:	80115
Anahtar Kelimeler	:	Topoloji optimizasyonu, Mimari tasarım, Ağaç benzeri yapılar, Grasshopper eklentileri
Sayfa Adedi	:	95
Danışman	:	Doç. Dr. Zeynep Yeşim İLERİSOY

ACKNOWLEDGEMENT

First, I would like to express my gratitude to my supervisor Assoc. Prof. Zeynep Yeşim İLERİSOY for her support and guidance over the years. Her patience, shared knowledge and given time, effort, and encouragement allowed me to successfully complete this journey. My gratitude is also extended to my family, especially my parents Hazem SELMI and Sounia ALMUHTASIB, and my siblings, who have supported me in every way with their unwavering love and trust. My second family, my uncle Mahmoud SELMI, his wife Hana ALFAR, and their children, who have been by my side throughout both my bachelor's and master's degrees, is also deserving of my gratitude. Lastly, my love and appreciation extend to my soul sisters Fatin ALJARAH and Abla ABDELRAZIK for helping me find a home abroad.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

Table **Page**

Table **Page**

Table **Page**

LIST OF FIGURES

Figure **Page**

Figure **Page**

Figure **Page**

SYMBOLS AND ABBREVIATION

The symbols and abbreviations used in this study are presented below along with their explanations.

Symbols	**Definition**
kN	Kilonewton
kN/m^2	Kilonewton per square meter
kN/m^3	Kilonewton per cubic meter
m	Meter
MPa	Megapascal

Abbreviation	**Definition**
2D	Two Dimensional
3D	Three Dimensional
BESO	Bidirectional Evolutionary Structural Optimization
Brep	Boundary representation
CPU	Central Processing Unit
ESO	Evolutionary Structural Optimization
FEA	Finite Element Analysis
FEM	Finite Element Method
GPU	Graphics Processing Unit
ML	Machine Learning
SIMP	Solid Isotropic Material with Penalization
TO	Topology Optimization

1. INTRODUCTION

Advances in computer tools allowed for the use of algorithmic and parametric design which aids in producing completely different designs in comparison with traditional design tools. Within these technological advances, some concepts from specific fields are being introduced into multiple different fields. Topology, which is a mathematical concept related to geometry, has been introduced and applied in other fields. The concept of topology has been explored in the design field and applied during different design stages. An important extension of the concept is topology optimization.

Topology is considered one of the newer branches of mathematics. The German mathematician Listing was the first to use the word topology in 1845 but its concepts were introduced before that. It is derived from the Greek words' topos, which means place or space, and logos, which means science. While topology studies the properties of surfaces and shapes, it is not concerned with length and angle. In other words, it examines the qualitative features of geometry in depth instead of quantitative ones (Kökcü, 2020). Topology is defined in Cambridge Dictionary as "The way the parts of something are organized or connected" and in Collins Dictionary as "A branch of geometry describing the properties of a figure that are unaffected by continuous distortion, such as stretching or knotting" (URL-1; URL-2). In topology, objects do not consist of just a set of rigid points but are imagined to be made of a flexible material, so they can be transformed into "different geometrical shapes" by deformation. As it is concerned with a more dynamic concept of objects rather than rigid geometry, topology is sometimes called "rubber geometry" (Zeeman, 1966).

Optimization in Collins Dictionary is defined as "making the best of anything" (URL-3). It is "an act, process, or methodology of making something (such as a design, system, or decision) as fully perfect, functional, or effective as possible" (URL-4). Optimization can be found in nature as different species evolve to optimize different functions and parts of their bodies (Martins & Ning, 2021). In mathematics, optimization is "a mathematical technique for finding a maximum or minimum value of a function of several variables subject to a set of constraints" (URL-3). Design optimization follows the mathematical concept in which an optimization problem including design variables, an objective function, and constraints needs to be defined to start the design optimization process. Although the optimization process is highly automated, human intervention is needed in deciding the different variables of the optimization problem before starting the process. Likewise, it is needed after the results are obtained to evaluate and revisit and edit the original problem. As optimization problems are mostly complex, the development of numerical computing allowed for more exploration of such problems along with solving them (Martins & Ning, 2021).

Topology optimization (TO) which connects the two concepts can be defined as the process of determining the optimal material distribution within a design domain in order to minimize (or maximize) an objective function while respecting certain restrictions (Yuksel, 2019). Topology optimization is used to reach the “optimum” design from all the different aspects and criteria specified. Topology optimization is used in multiple fields including architecture, structural engineering, mechanical engineering, product design, and even medicine. It is applied as a method for form finding based on specific criteria and needs. Topology optimization is done by specifying a design area or space in which boundaries and applied loads are specified. This design area specified is called a Design Domain. For optimization, the finite element method (FEM) is usually used, which breaks a huge system down into smaller parts called finite elements. FEM is paired with an optimization algorithm to determine the material distribution within a domain (Bialkowski, 2018; Sutradhar et al., 2010).

Early applications of topology optimization using the computer required writing codes that vary in complexity depending on the design problem. However, with the advances in computer software, multiple programs and add-ons are designed to conduct TO processes without the need to code. Although it might be a bit limiting, this allows for easier implementation of TO in different fields including architecture. These tools include grasshopper plugins that can be easily connected to other components and plugins. That is along with grasshopper being increasingly used in the architectural community which shows a need for exploring and evaluating its plugins.

Problem Statement

As topology optimization is relatively newly introduced in the architectural fields, more research on its possible applications is needed. A main objective of TO is to use the minimum material while reaching the maximum possible efficiency for the structure. As the minimization of used material is a main goal also in biomimetic architecture, tree-like structures are chosen as a case study to explore the potential of TO in architecture.

Research Questions

Main research question:

- How can topology optimization be used to produce different forms of tree-like structures?

To answer the main research question, other sub-questions must be answered first:

- How efficient are the available grasshopper plugins in implementing topology optimization during the conceptual stage?
- How do different domains and parameters affect the form of treelike structures?

Research Objectives

- Study the implementation of topology optimization to produce a structurally optimized architectural design.
- Examine the impact of various design parameters and user decisions on the form of topology-optimized structures.
- Explore the available grasshopper plugins for TO and study its capabilities.

Scope of the study

The research uses tree-like structures, which are nature-inspired, as a case study to explore the potential of implementing TO in architecture. The tools explored for the execution of case studies are limited to topology optimization plugins for grasshopper. Namely 3 plugins which are tOpos, Millipede, and Ameba. The reason behind focusing on grasshopper plugins only is that grasshopper is a popular platform and topology optimization plugins can be connected with other tools easily to perform other tasks and studies. That is along with the time limitation which does not permit exploring more tools. As the research focuses on studying and comparing the resulting forms from TO and their differences based on defined parameters, the fabrication process is beyond the scope of this research.

Methodology

The thesis is structured into mainly three parts (Figure 1.1). A comprehensive literature review on topology optimization, its available applications in architecture, available tools, and tree-like structures is conducted first. The literature review is followed by a further exploration of Grasshopper tools. Multiple basic architectural geometries are used to explore the different Grasshopper TO plugins. These plugins are compared based on multiple criteria including computation time and resulting geometries. The case studies are conducted to determine which plugin serves the research best. The plugin found to be most suitable is used for the implementation of TO to produce tree-like structures. Multiple domains with different numbers of columns and configurations are used to explore various possible forms. A discussion on the results of both case studies of tools and treelike structures is conducted.

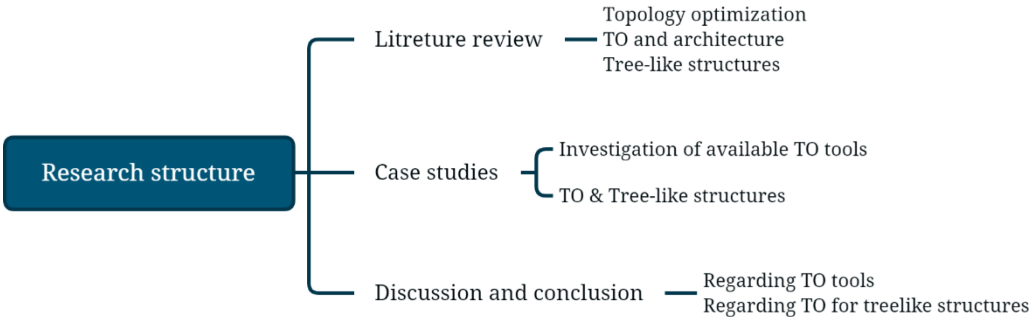


Figure 1.1. Thesis structure

2. TOPOLOGY OPTIMIZATION

There are three primary structural optimization methodologies: size, shape, and topological optimization (Figure 2.1). Size optimization is a method used for determining appropriate material distribution for a solid structure or the member thickness in a discrete construction. As a result of size optimization, the ideal cross-sectional area of structure parts or members is determined by adjusting the dimensions of each structural member. In a problem involving shape optimization, however, the design variable is the shape, and the purpose is to define the form that is optimal for the design. In shape optimization, only the geometry of the structure's exterior shell is modified, without the element connections. During topology optimization, the design domain is established and used to determine connectivity along with the number, shape, and location of holes within it (Bendsøe & Sigmund, 2003; Eren & Sezer, 2019).

The objective of size optimization in frame and truss constructions is to determine the ideal cross-sectional areas of members. The goal of shape optimization is to optimize the placements of nodes that alter the geometry without adding or removing elements or voids. Topology optimization, in contrast, can introduce a new topology and add or delete elements and voids (Vlah et al., 2020). Typically, topology optimization is employed at the conceptual stage, whereas size and shape optimization are employed during the detailed design phase (Zhu et al., 2021). In general, the first design stages of a structure involve a topology optimization procedure and a shape and/or size optimization to determine the final design. Some research including Hassani et al. (2013) and Yoely et al. (2018) provide methods for applying both topology and shape optimization simultaneously.

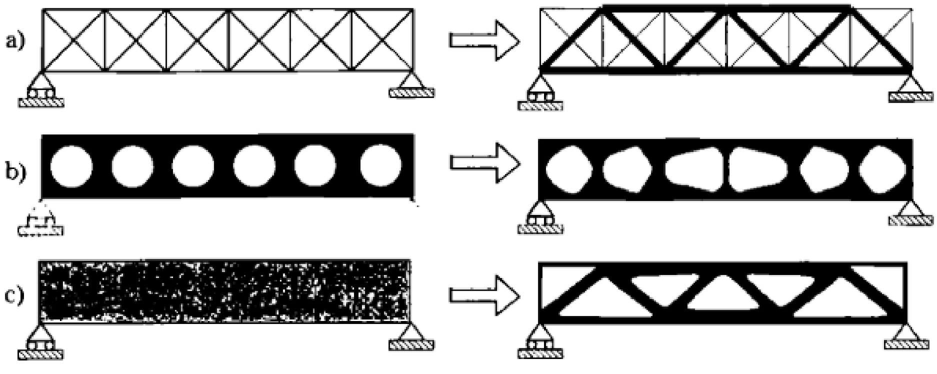


Figure 2.1. Categories of Structural optimization applied on a truss. A) Size optimization B) shape optimization C) Topology optimization (Bendsøe & Sigmund, 2003)

The primary distinction between topology optimization and other optimization techniques is that topology optimization develops entirely new structural layouts that are distinct from normal configurations (Feringa & Søndergaard, 2012). While topology optimization requires only a simple beginning volume, size and shape optimization require a parameterized preliminary model to initiate the optimization operation (Vlah et al., 2020).

Topology Optimization (TO) is a process used to detect the ideal allocation of material inside a design domain (Bendsøe & Sigmund, 2003). The primary engineering topic addressed by topology optimization is how to put materials inside a domain to obtain the most effective design (Sigmund & Maute, 2013). It is a sub-method of structural optimization that has widespread application in mechanical and automobile engineering. Nevertheless, it has architectural prospects and can open up several opportunities. As economic considerations and structural stability are vital in structural engineering, optimization greatly reduces resource consumption while achieving performance goals. Topology optimization can be utilized to produce unique, structurally sound architectural forms with the use of computational techniques (Naboni & Paoletti, 2018).

Multiple iterations of assessment and design revisions form the foundation of topology optimization. Typically, the objective function of a topology optimization procedure is minimized by determining the distribution of materials within a domain (Sigmund & Maute, 2013). Most of the time, a compliance function serves as the objective function during a topology optimization problem. Such a TO problem aims to increase stiffness by decreasing compliance (minimizing the objective function) (Deaton & Grandhi, 2014). Based on the provided loads, supports, and perhaps other design requirements, topology optimization allocates material inside the previously designated region. There are also user-specific design constraints like predetermined voids and solid sections, as well as specific performance targets (Naboni & Paoletti, 2018). Multiple parameters must be defined before starting the process of topology optimization. These include the design domain, position and type of supports, loads acting on the design space, material properties, and other optimization constraints that may exist such as symmetry (Feringa & Søndergaard, 2012).

After the optimization problem, design parameters, and boundary conditions are defined, the optimization process is started (Figure 2.2). Finite element method (FEM) is applied in which the domain is discretized into a specific number of finite elements (Cuellar Loyola et al., 2015). FEM is used to perform a finite element analysis (FEA) which is a mathematical method for displacement calculation. After the displacements get calculated by the FEA, Sensitivities of both the objective and constraints functions are calculated (Martens, 2018). The concept of sensitivity represents the change rate of the objective function in response to variations in the design variables. This establishes the way in which design variables are altered. The optimization algorithm is then run to update the design variables and drive the iterative feedback loop (Woo, 2020). The design is checked, and new iterations are made until the optimum result is reached.

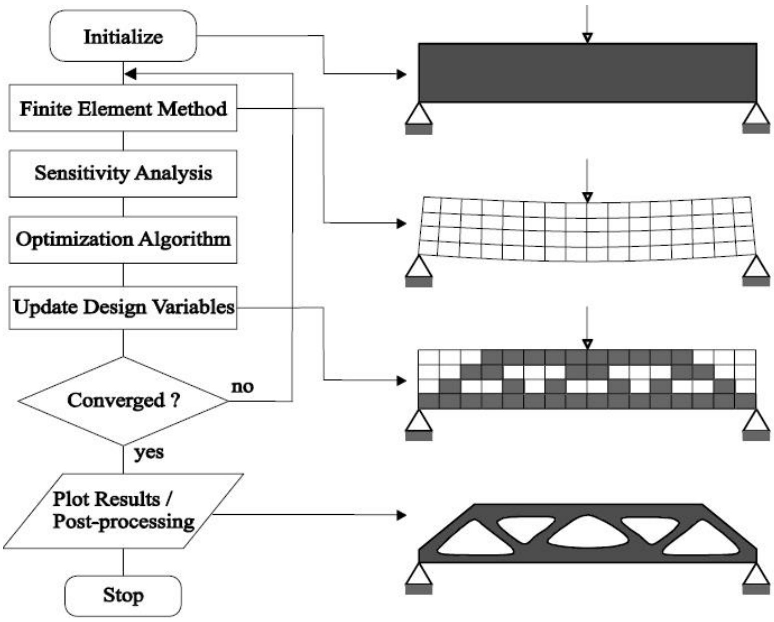


Figure 2.2. Topology optimization process (Cuellar Loyola et al., 2015)

2.1. Methods of Topology Optimization

After classifying the optimization methods into three main types (size, shape, and topology), there is a need to identify the main topology optimization methods (Figure 2.3). Although there are many available TO methods used in literature, the main ones include Level set, homogenization, SIMP, and evolutionary methods. One may argue that homogenization was the first TO technique to be used. However, it appears from the literature that SIMP and BESO are now the two most prevalent TO approaches.

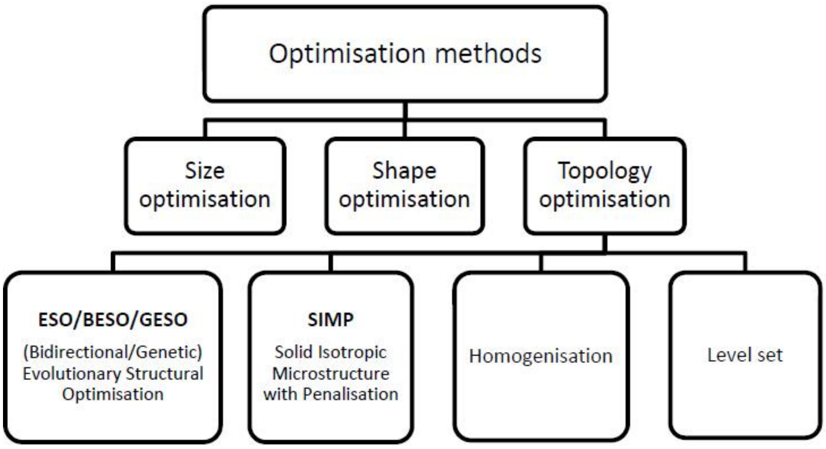


Figure 2.3. Optimization methods (Lundgren & Palmqvist, 2012)

Solid Isotropic Material with Penalization (SIMP) is a density approach that divides the domain into finite components and assigns the material density based on the design variables (Sigmund & Maute, 2013). The density varies between 0 and 1 with 0 representing no void, 1 representing solid and intermediate densities that could be represented as gray areas in a visual illustration (Figure 2.4). The middle densities are eliminated using a penalty method to provide definite densities of 0 (empty) and 1 (solid) (Deaton & Grandhi, 2014). In contrast, the discrete method known as Evolutionary Structural Optimization (ESO) uses density variables that are either 0 or 1, with no in-between densities. ESO is regarded as a hard-kill technique in which components that don't meet the requirements are eliminated. It has evolved into a bidirectional technique (BESO) where components can be taken out and put back in at any time. According to Sigmund and Maute (2013), BESO should be viewed as a discrete variation of the SIMP approach rather than a distinct category. The outputs are produced by both of them using the same sensitivity and density filters, but they differ in their update strategies, such as whether they are discrete or continuous.

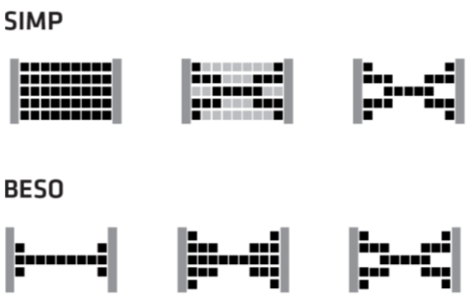


Figure 2.4. Difference between SIMP and BESO (Woo, 2020)

2.2. Applications of Topology Optimization in Different Fields

Aircraft and mechanical engineering have utilized topology optimization to build optimum, lighter, and more efficient components and elements. In fact, it can be said that topology optimization has started and is the most popular in these fields. Figure 2.5 shows the process of optimizing a jet engine bracket as discussed by Gebisa and Lemu (2017). With the development of additive manufacturing (3D printing), the manufacturability of topology-optimized products has increased.

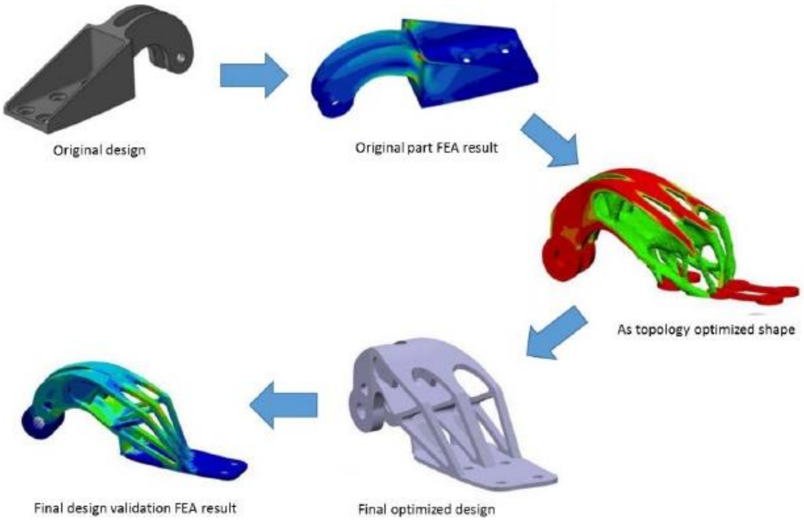


Figure 2.5. TO for a jet engine bracket (Gebisa & Lemu, 2017)

Another field in which topology optimization is researched is product design. The chair design in Figure 2.6 adopts a comprehensive strategy that, through topology optimization, incorporates many constraints and needs. Along with user requirements, structural, economic, and material factors have been taken into account (Hemmerling & Nether, 2014).

Figure 2.6. Topology optimization stages of a chair (Hemmerling & Nether, 2014).

Additionally, topology optimization has application potential in other fields, including medical applications like bone replacement. Using topology optimization to create a patient-specific facial bone replacement is proposed by Sutradhar et al. (2010). Brain MRI data were used to select the design region. Later, additional information about loads, boundary conditions, and various space restrictions was added to the design domain (Figure 2.7).

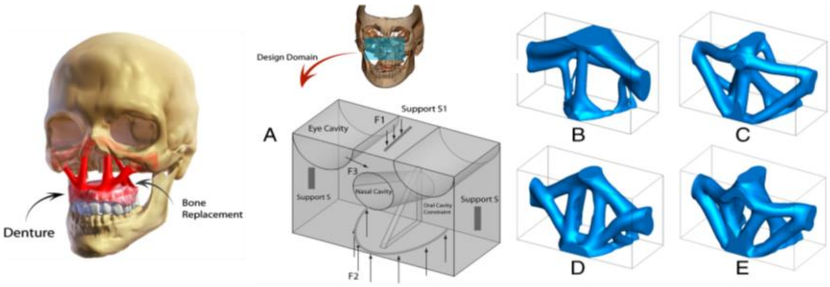


Figure 2.7. Bone replacement design using topology optimization (Sutradhar et al., 2010)

2.3. TO Tools and Plugins

While manually conducting a TO process is nearly impossible, writing a code for computation can also be challenging. Thus, numerous programs have a TO tool built into them making the process easier. While some of these tools are standalone, others are add-ons. A general comparison of a few of the TO tools that are used in literature is presented in Table 2.1. It is important to select the right tool for the intended TO problem. Thus, a detailed exploration and comparison are needed to decide on which program to use.

Table 2.1. Comparison of available Programs for topology optimization (Avdić, 2019; Bialkowski, 2018; Shao, 2020; Tyflopoulos et al., 2018)

Tool	Environment	Method	Accessibility	Dimension
Millipede	Grasshopper	SIMP*	Free	2D&3D
TopOpt	Grasshopper	SIMP	Free	2D
Topos	Grasshopper	SIMP	Free	3D
Ameba	Grasshopper	BESO	Commercial	2D&3D
Karamba3D	Grasshopper	BESO	Commercial	Discrete 3D
BESO3D RMIT	Python	BESO	Free	2D&3D
Ansys	Stand-alone	SIMP	Commercial	2D&3D
Altair Optistruct	Stand-alone	SIMP	Commercial	3D
Topostruct	Stand-alone	Homogenization	Commercial	2D&3D
Autodesk Fusion 360	Stand-alone	SIMP	Commercial	3D
Tosca	Abaqus	SIMP	Commercial	2D&3D
*Some resources refer to Millipede's method as homogenization. It is more likely that the method is SIMP based on the plugin's supplied parameters.				

Literature that compares between topology optimization plugins and tools is limited. Rodríguez et al. (2019) provides an overview of 5 different commercial software programs that include a topology optimization algorithm (CREO, ABAQUS, ANSYS, COSMOS and NASTRAN). However, it does not include a clear comparison between the programs. Also, it provides a case study that is only implemented in one program (ANSYS) for the goal of showcasing the topology optimization process. In contrast, Tyflopoulos and Steinert (2022) compares between topology optimization results of 3 commercial software programs: SolidWorks, ANSYS, and ABAQUS. Three case studies are used (a bell crank lever, pillow bracket, and small bridge) to compare between the programs (Figure 2.8). It compares the results based on design cycles, optimization time, weight, maximum Von Mises stresses, and the minimum factor of safety (FOS) against yield.

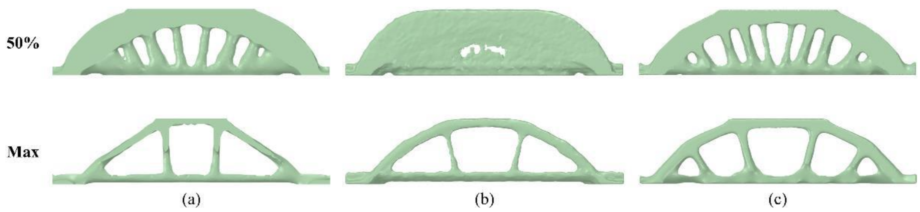
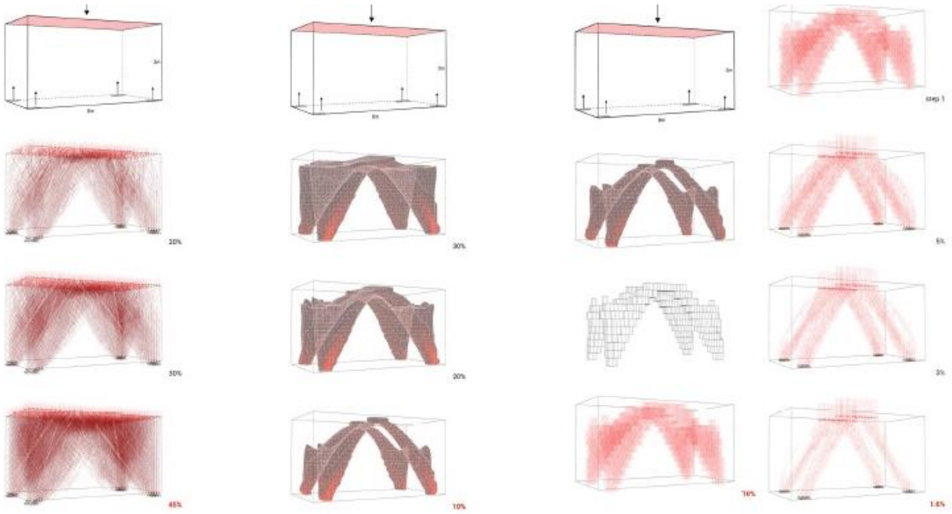


Figure 2.8. Topology optimization studies on a bridge (a) Solid-Works, (b) ANSYS Mechanical, and (c) ABAQUS. (Tyflopoulos & Steinert, 2022)

Shao (2020) compares between SIMP and BESO in discrete digital design and proposes a new hybrid method that is supposedly more efficient. The reasoning behind it is that neither BESO (using Karamba3D) nor SIMP (using Millipede) produce the best results. It topology optimizes 5 different structures (a cube, truss, column, floor, and beam) using both Karamba3D and Millipede and later using a hybrid method. In the hybrid method, the structure is optimized using millipede, transformed into a discrete grid and then re-optimized in Karamba3D. An illustration of the 3 experiments can be seen in Figure 2.9 where a one-floor building structure is optimized using the 3 methods. The hybrid method resulted in more efficient structures according to the comparison done in the article.



(a) (b) (c)

Figure 2.9. Topology optimization studies on a one-floor building structure (a) using Karamba3D (b) using Millipede (c) using a hybrid method (Shao, 2019)

Grasshopper plugins

There are five grasshopper plugins that can be used for TO. TopOpt is only intended for 2D topology optimization, and it doesn't appear that it is being further developed. The plugin is not being further investigated as attempts to obtain any results failed. Karamba3D is a plugin that is not specifically dedicated to topology optimization. It is a structural engineering tool that offers different analyses for trusses, frames, and shells. Instead of the continuous results obtained by other plugins, the tool for ESO included in Karamba3D optimizes discrete structures. Karamba3D is hence eliminated from the comparative study. Thus, tOpos, Millipede, and Ameba are the plugins to be studied.

Ameba is introduced, described in detail, and given some straightforward implementation examples by Zhou et al. (2018). tOpos is also introduced, and its features are discussed in multiple publications (Bialkowski, 2016, 2017, 2018). No article introducing Millipede was found. There is however an online manual available that covers the plugin and its features (URL-5). Unfortunately, millipede has been discontinued, hence no new updates or enhancements are being made. With that being said, the latest available version of each plugin will be used for study and comparison. Thus, Ameba version 2 and tOpos version 1 are used in this study. It is unclear how many versions Millipede has, and which one is currently available online, but it is the one used in this study.

While Millipede, tOpos, and Ameba are based on different TO methods, they all have a similar way of generating outcomes in terms of obtaining continuous structures as opposed to discrete ones. Table 2.2 presents and compares between the three plugins. Millipede is a structural analysis and optimization plugin for grasshopper. Millipede can be used for 2D and 3D topology optimization together with other analysis procedures. On the other hand, tOpos is a grasshopper plugin dedicated to only 3D topology optimization. Ameba is another plugin built explicitly as a topology optimization tool. Both 2D and 3D geometrical models can be topology optimized using Ameba. While Millipede and tOpos are free to download and use, Ameba is a commercial software that offers a free version with a limited element number. Though material is defined in all plugins by Poisson's ratio and Young’s modulus, Millipede also includes a list of predefined materials to select from while having the option to define a new one. The geometries that need to be defined in grasshopper for domain, loads, and supports vary from one plugin to another. Millipede and Ameba require these geometries to be Breps while it is a mesh in tOpos. Brep (boundary representation) is an analytically defined geometry containing one or multiple surfaces joined together. The way it is defined makes it always smooth no matter how zoomed in it could be. A mesh, on the other hand, is simply defined by points and polygons connecting these points which makes it not smooth when zoomed in. Practically, a mesh can be invalid in many ways and difficult to perform some operations on but is easily transported to other apps. A Brep can be easier to handle, and more accurate but hard to transform to other apps. Also, it is fairly easy to convert a Brep into a mesh, but the other way around could be challenging (URL-6). This indicates that if the geometry in hand is a mesh, it can be hard to use Millipede or Ameba for TO as they require a Brep definition. Also, loads and supports have to be selected as different geometries from the domain one in Millipede and tOpos. Ameba, however, requires the loads and supports to be selected as surfaces from the same Brep defined as domain. This means that the Brep needs to be designed and its surfaces split with consideration on which ones will be selected as loads and supports. When compared to millipede, tOpos can work on graphics processing unit (GPU) which provides much faster computation compared to the conventional central processing unit (CPU). However, the GPU option occasionally stops working after a few iterations or displays an error message after just one optimization process. It is also unstable and does not operate with all graphics cards. Ameba on the other hand is cloud computation based and its speed can be affected depending on the internet connection available. The maximum number of iterations needs to be specified in all the plugins. This could be a negative point as there is no option to allow the procedure to run freely until reaching the optimal geometry. Nevertheless, it was noted that tOpos and Ameba's computation does sometimes stop before reaching the maximum number of iterations which indicates the goal was achieved faster than expected. While all three plugins result in meshes, Millipede and tOpos have different options for displaying results. That includes an Iso mesh in both plugins, a voxel mesh in tOpos, and stress lines in Millipede. tOpos also includes a Preview of densities as a blurred point cloud or a Dot style preview of element's data. Parts of the domain can be set as a fixed solid or void area in tOpos and Millipede. Ameba, however, includes a no-design domain option which is a fixed solid area but there is no option to define a fixed void area. This can be overcome by trimming the areas that are intended to be void from the geometry before defining it as a domain.

Table 2.2. Comparison between Millipede, tOpos, and Ameba

	Millipede	tOpos	Ameba
TO method	SIMP	SIMP	BESO
License	Free	Free	Commercial
2D/3D	2D&3D	3D	2D&3D
Material properties	List of predefined materials/custom material	Poisson's ratio and Young's modulus	Poisson's ratio and Young's modulus
Domain setup	Brep selection	Mesh selection	Brep selection
Loads	Volume (Brep selected as a load different than the domain Brep)	4 options: Point, linear, surface, and volume loads (Mesh selected as a load different than the domain mesh)	Surface (Selected from the same Brep selected as domain)
Supports	Brep selected as support (different than the domain Brep)	Mesh selected as support (different than the domain mesh)	Surface (Selected from the same Brep selected as domain)
Computation	CPU	GPU, CPU options	Cloud computation
Iterations	Max number of iterations has to be specified	Max number of iterations has to be specified	Max number of iterations has to be specified
Resulting geometry	Mesh, Iso mesh, stress lines	Iso mesh, voxel mesh, DensPreview, ElePreview	Mesh
Options	Fixed density region (void or solid)	Fixed and void mesh	Multiple load cases, non-design domain
Notes	Optimization sometimes results in disconnected geometry	GPU is sometimes unstable, stops during iterations or after one TO process	Optimization sometimes results in disconnected or distorted geometry

While Ameba uses a surface to define the load, Millipede uses a volume to do so. On the other hand, there are 4 alternative load options offered by tOpos. The optimization outcomes are significantly different when one is picked over the others. A comparison of the various load types and the resulting topology optimized structures based on each load type is shown in Figure 2.10. When a curved line at the center is given as a load, material is distributed throughout the domain with a concentration around the middle line. When a volume load going through the center is given, however, the material seems to be distributed only in the middle part of the domain which could be associated with a volume being heavier thus the algorithm avoided having a large structure with cantilever concave parts. For the third option, the whole outer surface of the domain is given as the load which results in a solid structure covering the whole domain with the inner part being optimized. The final alternative is investigated by applying a point load on the domain's center. The result was the most minimal with material distributed from the center where the load is and branches to four-column-like parts. This example shows how changing one design parameter, in this case load, results in totally different structures.

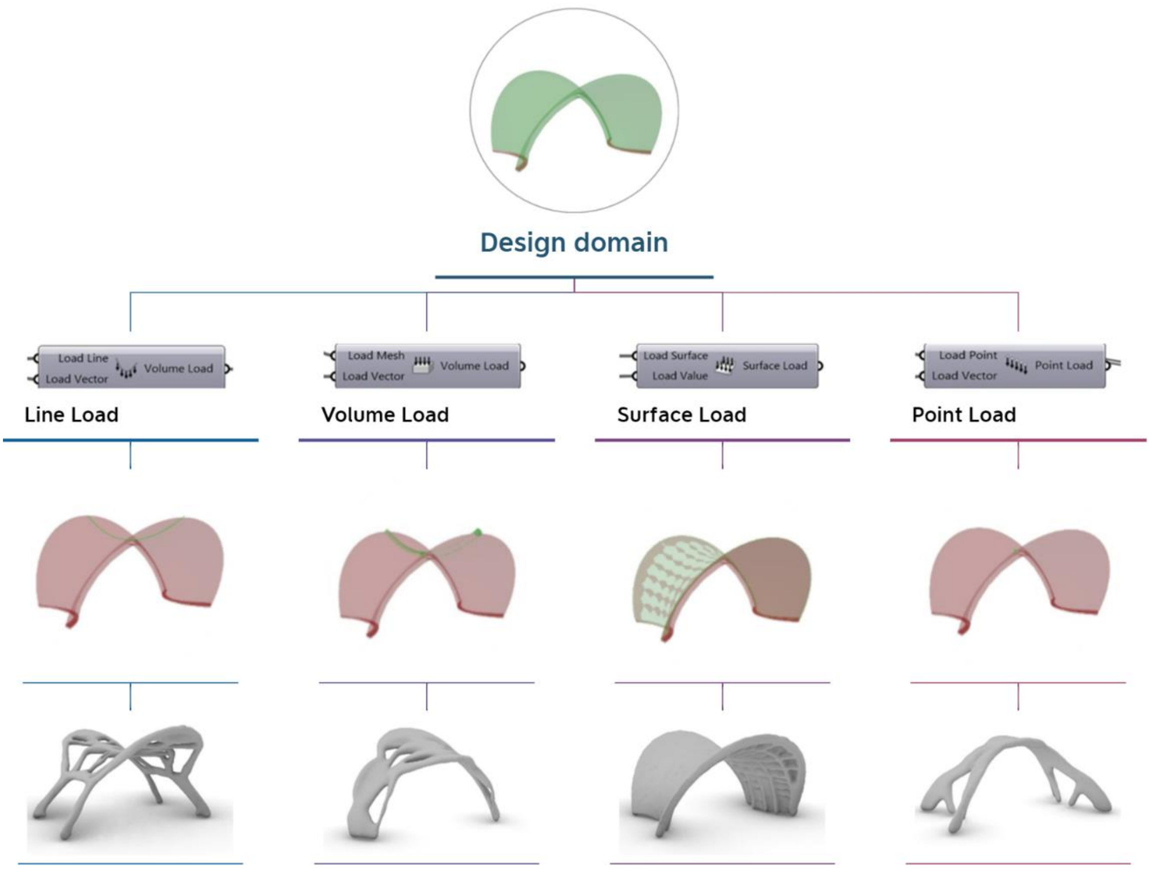


Figure 2.10. Comparison between different load types available in tOpos plugin

This chapter explored the basic concept of topology optimization, its different methods, and applications in other fields. It also briefly explored available software for implementing TO with taking a deeper dive into grasshopper plugins which are a main focus of this research. The preliminary exploration and comparison between the three grasshopper plugins Millipede, tOpos, and Ameba show that the three have a potential in producing valid TO results. Although the three plugins have their similarities and differences, they generally produce similar continuous results which allow them to be compared. Thus, a thorough study that uses the same domains and boundary conditions is needed to provide clearer insight into the performance of the three plugins.

3. TOPOLOGY OPTIMIZATION IN ARCHITECTURE

In contrast to other fields like mechanical and aerospace engineering, topology optimization is less common in the subject of architecture. However, as computer-aided design has advanced and sustainability has become a priority, TO can now offer an excellent tool for creating innovative designs while utilizing less material (Naboni & Paoletti, 2018). Traditionally, the process of designing an architectural structure is linear. After beginning to create the architectural shape and geometry, the architect turns the work off to a structural engineer (Zegard et al., 2020). Together, architects and engineers may create architectural elements that fulfill both aesthetic and structural requirements using topology optimization (L. L. Beghini et al., 2014).

3.1. TO Large-Scale Implemented Projects

Topology optimization is a fairly new method, particularly within the field of architecture, hence there are very few completed large-scale projects that utilize it (Figure 3.1). Qatar National Convention Centre is one project in which TO is used to create an organic architectural form that supports the cantilever outer roof while minimizing material consumption. A tree-like structure was obtained by determining the material properties and loading conditions. While two support points are given 100 m apart, the top of the structure is fixed to remain flat (Burry & Burry, 2010). TO is used also in the 100 Mount Street project by SOM to produce an exoskeleton that achieves maximum performance by minimizing its structural weight, increasing openings, and having the least possible cross-section. The cross-braced exoskeleton structure was solved with TO in order to prevent torsional and bending forces in the structure whose structural core was placed on the opposite side. Thus the structure reaches a balance (Schittich, 2015). In the Akutagawa River Side project, TO was used to determine the form with the best performance and least material in shaping the reinforced concrete structure on two facades. Dynamic earthquake loads are taken into account in optimized reinforced concrete walls (Januszkiewicz & Banachowicz, 2017).

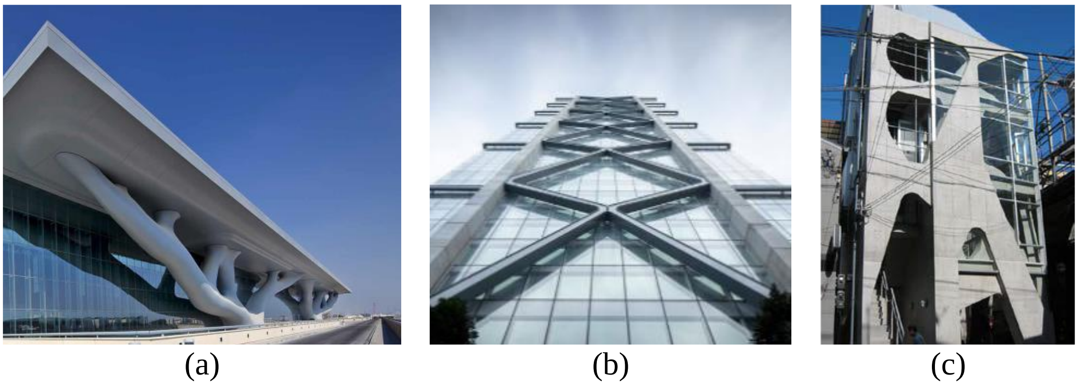


Figure 3.1. Built large-scale TO projects (a) Qatar National Convention Centre (b) 100 Mount Street (c) Akutagawa West side project (Motro, 2011; Naboni & Paoletti, 2018; URL-7)

3.2. Research on Topology Optimization for Architecture

As the interest in implementing TO in the architectural field had increased in recent years, multiple researchers dedicated their work to exploring the possibility, challenges, and alterations that could be used to implement TO more effectively in the architectural field. Yan et al. (2022) introduce three strategies to design topologically optimized structures for solving practical architectural design problems. The strategies introduced are applied to various domains including shells, high-rise structures, and beams to study their efficiency. Another study presents a new approach for using structural topology optimization to evaluate and generate bioinspired architectural design (Mizobuti & Vieira Junior, 2020). Different 2D and 3D domains (many of which are shells) are topology optimized and the results are compared to existing structures in nature (Figure 3.2).

Figure 3.2. Example of 2D and 3D TO results that are compared to nature (Mizobuti & Vieira Junior, 2020)

Lin et al. (2021) provide a thorough study of how to implement topology optimization for element generation in the architectural field. The parameters of topology optimization and their influence on the results are discussed. An overview of the possibilities of the parameters: design domain, load scenario, and boundary condition for different structures are discussed. The parameters for different architectural elements including walls and shells are discussed (Figure 3.3). A case study on beams is conducted to investigate the influence of changing parameters on the result of topology optimization. A more detailed study is conducted on shells. A shell structure is topology optimized twice using Abaqus/Tosca once with no holes and the other with holes introduced (representing architectural requirements such as roof lighting or ventilation functions). A combined analytic hierarchy process (AHP) and Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) method are used in the study to analyze and evaluate topology optimization results. The analysis is based on aesthetical attributes to compare the different results.

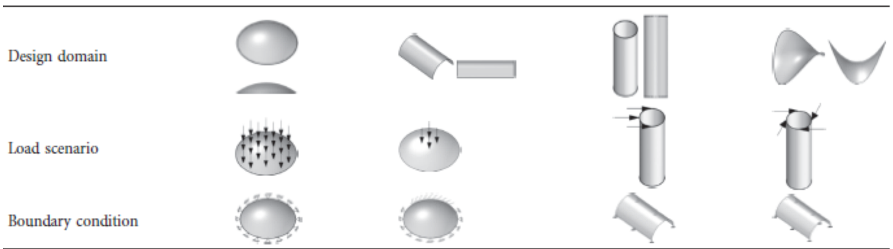


Figure 3.3. Different parameters for shell structures (Lin et al., 2021)

Another case study also explores the different aspects of TO in architectural design (Xie, 2022). It discusses the limitations of the conventional approach to TO when it comes to architectural elements and examines suggestions on how to overcome these limitations for a better implementation of TO in architectural design. Dapogny et al. (2017) also analyze how geometric constraints applied in the conceptual design stage of architectural design using TO could affect the results and produce different designs. It explores multiple scenarios varying in complexity, evaluates and compares the different results. One example is a domain with different support types and non-design areas (Figure 3.4). In the first three cases, the design domain is the same and the only difference is in the type and location of supports. This little difference in support results in changes in the result of topology optimization. In the first case, the displacement is fixed in the two right corners, while the left corners are free to move in the x-direction. The optimized structural system includes flat bars in compression transferring loads to an inverted arc in tension in the middle of the structure. Two other inclined bars in compression work with some horizontal bars in tension to transfer the load to the ground. In the second case, additional support was added on each side, resulting in different optimization outcomes. By setting supports along the sides rather than just the corners, the optimized shape material is concentrated in the center of the structure. In the third case, all the supports are fixed. The lower part of the structure is no longer needed (it was needed in the first 2 cases to act against horizontal displacements resulting from the free-to-move supports). The solution resembles an arc in compression with some vertical supporting bars. In the last case, a non-optimizable region is specified. Although the support is the same as in the first case, the result is different. The limitation in the design domain changes the load distribution and design shape. It is noticeable that the type, shape, and size of structural elements change to adapt to the newly defined design domain.

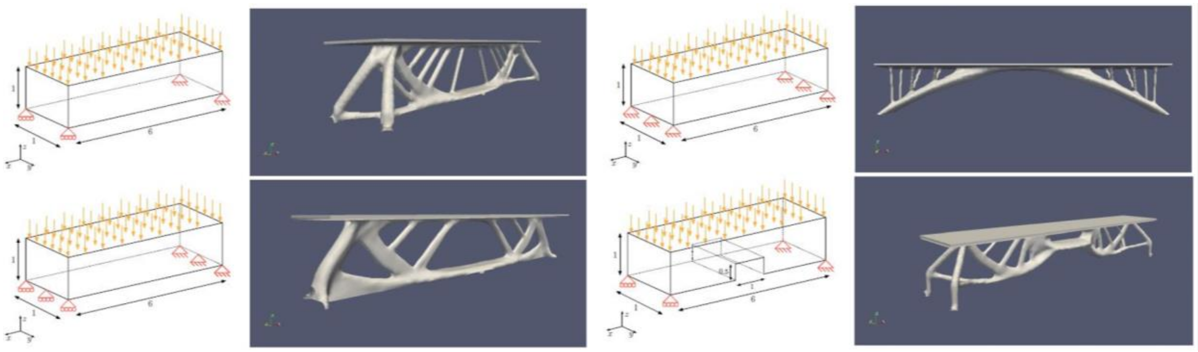


Figure 3.4. Specifying different support types and non-optimizable regions produce different optimization results (Dapogny et al., 2017)

3.3. Research on The Applications of TO for Different Structures

Although there aren't many full-scale implementations yet, interest in topology optimization's potential has grown recently, and recent research examines a variety of methods in various structure types. Other than studies working on generally exploring the implementation of TO in the architectural field, multiple studies focusing on designing specific architectural elements can be found in literature. The following tables explore the different available research on TO for architecture and classify them based on structure type.

Table 3.1 shows research on applying TO for slab design. While Magan (2016) and Martens (2018) explores the process of design with consideration of some type of fabrication constraints, no actual fabrication is done in these studies. The other two projects go further into exploring the fabrication process of topology optimized slabs. Both projects use some sort of mold to fabricate the slab and differ in that the Unikabeton Project in Søndergaard and Dombernowsky (2017) uses a removable mold while the others explore using stay-in-place molds. Stefanaki (2020) differs from the other studies mainly in material as it uses glass instead of concrete.

Table 3.1. Research exploring topology optimization of slabs

	Study goal	Structure type	Material	Used program	
(Magan, 2016)	To study the application of TO in concrete slab design with formwork constraints	Slab	Concrete	Python	
(Søndergaard & Dombernowsky, 2017)	To examine the process of combining the robot-fabricated concrete casting molds and the TO computational procedure	Slab (mold)	Expanded polystyrene (EPS)/ Concrete	OptiStruct	
(Jipa et al., 2016; Meibodi et al., 2017)	To investigate how large-scale building components can be prefabricated using additive manufacturing (AM)	Slab (Stay-in-place formwork)	Sandstone/ Concrete	Grasshopper Millipede (Prototype A)	
				Abaqus (Prototype B)	
(Martens, 2018)	To explore a method for applying TO for the design of concrete slabs with 3D printing constraints	Slab	Concrete	Rhino Python	
(Stefanaki, 2020)	To examine the process of designing large-scale glass slabs using topology optimization	Slab	Glass	ANSYS	

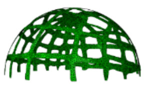
Examples of pavilions designed with the help of TO are found in Table 3.2. While both projects are fabricated to scale using some sort of polymer, Naboni et al. (2020) includes a more complex process of design and fabrication using various methods including TO.

Table 3.2. Research exploring topology optimization of pavilions

	Study goal	Structure type	Material	Used program	
(Bao et al., 2020)	To demonstrate a design and robotic fabrication workflow for a topology-optimized structure	Pavilion	Translucent polymer	Grasshopper Ameba	
(Naboni et al., 2020)	To investigate the feasibility of using FDM to create a lightweight cellular lattice structure	Lattice Structure (Pavilion)	Bio-based Polylactic Acid	Grasshopper Millipede Karamba Ansys	

Table 3.3 includes research on shells and trusses. Some explore the process of design only while others explore the fabrication too whether that is in a scaled or true-to-size fabrication.

Table 3.3. Research exploring topology optimization of shells and trusses

	Study goal	Structure type	Material	Used program	
(Søndergaard et al., 2013)	To demonstrate an automated process that combines the design, optimization, and manufacturing of space-frame structures	Space truss	Metal rod (wire for prototype)	Grasshopper MATLAB	
(L. Beghini et al., 2014)	Exploring the use of the graph static method to produce a topologically optimized structure	Roof truss	Steel	-	
(Søndergaard et al., 2016)	To establish a new approach for the design and fabrication of timber space frames that integrates topology optimization	Space Frame	Timber	Grasshopper MATLAB	
(Bartels & Houben, 2016)	To investigate a parametric structure optimization process that considers the impact of 3D printing	Freeform shell	Concrete (linear material model is used for analysis)	Grasshopper Python Abaqus	
(Melcher, 2020)	Comparison between topology optimized shell, continuous shell, and grid shell under asymmetrical loads	Shell structures	ABS plastic	Rhino Grasshopper Abaqus	
(Naous, 2020)	Designing a topology-optimized glass shell and examining its manufacturability	glass cast shell	Glass	Rhino Grasshopper ANSYS	

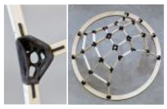
Pillars designed with the help of TO are showcased in table 3.4. Gaudillière et al. (2019) demonstrate a project in which a lost formwork produced using 3D concrete printing is employed to assemble a topology-optimized column.

Table 3.4. Research exploring topology optimization of pillars

	Study goal	Structure type	Material	Used program	
(Bhatia, 2019)	To explore the process of using 3D-printed sand molds to manufacture topology-optimized structural glass columns	Column (mold)	Sand / Glass	Ansys	
(Gaudillière et al., 2019)	To present a process for designing and manufacturing a structure using a 3D printed lost formwork	Pillar	concrete	-	

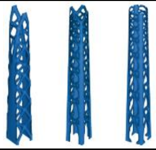
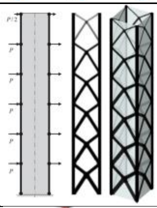
Table 3.5 include research exploring the design of nodes using TO. While all include some sort of fabrication process, they differ in the goal of the study and the material explored.

Table 3.5. Research exploring topology optimization of nodes

	Study goal	Structure type	Material	Used program	
(Galjaard et al., 2015)	To examine the feasibility of producing metal structural elements using additive manufacturing (AM) techniques	Structural node	1.4404 stainless steel	OptiStruct	
(Prayudhi, 2016)	To create a free-form structure during which additive manufacturing and topology optimization technologies are implemented	Free-form shell with topology optimized nodes	Carbon fiber-reinforced polyester	OptiStruct	
			Stainless steel powder		
(Damen, 2019)	To investigate the use of topology optimization as a means for designing structural cast glass components with shorter annealing times.	Grid shell node	Wax filament / Silica plaster / Cast glass	Ansys	

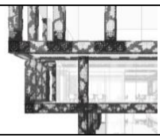
As the weight and material usage of the structure are especially crucial in high-rise structures, some studies explore the application of TO in high-rise buildings’ design (Table 3.6).

Table 3.6. Research exploring topology optimization of high-rise buildings

	Study goal	Structure type	Material	Used program	
(Stromberg et al., 2011)	To study the application of manufacturing constraints in topology optimization at the conceptual design stage	high-rise building	Steel	-	
(Stromberg et al., 2012)	To present a new method for designing structural braced frames for high-rise buildings that combine TO with other more conventional methods	high-rise buildings	Steel	MATLAB	
(Kingman et al., 2015)	To Examine the application of topology optimization on tall buildings and steel components	high-rise building	Steel	Altair HyperWorks/ Altair Optistruct	

On the topic of exterior shells of buildings, other studies explore designing building facades without the focus on high-rise buildings (Table 3.7). Woo (2020) explores the design of a building envelope using TO. Piccioni (2019), on the other hand, studies a more complex case of designing for structural stability using TO and thermal comfort using other methods. A partial prototype is prepared during the study.

Table 3.7. Research exploring topology optimization of building envelopes

	Study goal	Structure type	Material	Used program	
(Woo, 2020)	To explore the use of topology optimization in the architectural design process of the building envelope	Patterned building shell	Steel	Rhino Grasshopper Karamba	
(Piccioni, 2019)	To investigate the capability of additive manufacturing in producing a facade panel that achieves the best possible thermal and structural performance while having complex geometry	Facade panel	PETG	Grasshopper Millipede	

Other studies move from designing a single building component to looking at the application of TO in designing the building as a whole (Table 3.8).

Table 3.8. Research exploring topology optimization of building forms

	Study goal	Structure type	Material	Used program	
(Avdić, 2019)	Using topology optimization as a method for nature-inspired building forms	Building forms	Steel* *Material properties are used	Grasshopper Karamba Millipede GHPython Ansys	
(Dijk, 2020)	Using topology optimization as a form-finding tool in architecture with a focus on masonry structures	Masonry buildings	Masonry	Rhino Grasshopper Python	
(Guerguis & Principe, 2020)	To research feasible methods for the design and manufacturing of bioinspired topology optimized structures	House prototype	Polymer	Grasshopper Millipede Fusion 360	

Other projects which cannot be grouped by type with others are shown in Table 3.9. Søndergaard et al. (2018) shows the process of designing and fabricating a prototype that represents the different architectural elements of a wall, canopy, and corner structure. Vantyghem et al. (2020) describe a method for producing a post-tensioned girder in which TO and 3D concrete printing are explored within the project. Snijder et al. (2020), however, takes a different approach of designing a glass swing as a case study for exploring the design and optimization process of structures. It includes a TO process as one art of many design and optimization stages.

Table 3.9. Research exploring topology optimization of other structures

	Study goal	Structure type	Material	Used program	
(Søndergaard et al., 2018)	To explore the process of designing and fabricating a topology optimized structure using EPS formwork	Prototype	Expanded polystyrene (EPS) / concrete	Optistruct	
(Vantyghem et al., 2020)	To introduce a revolutionary design and production method that integrates post-tensioning, topology optimization, and 3D printing of concrete	Post-tensioned girder	Concrete	Grasshopper Fusion 360	
(Snijder et al., 2020)	To investigate the process of designing and fabricating a vector active glass structure	Glass Swing	Glass / Steel	Grasshopper Karamba3D Oasys GSA	

The previously discussed research can be categorized not only based on structure type but in regard to the material and fabrication process. The structure material can be divided into mainly six different types (Figure 3.5).

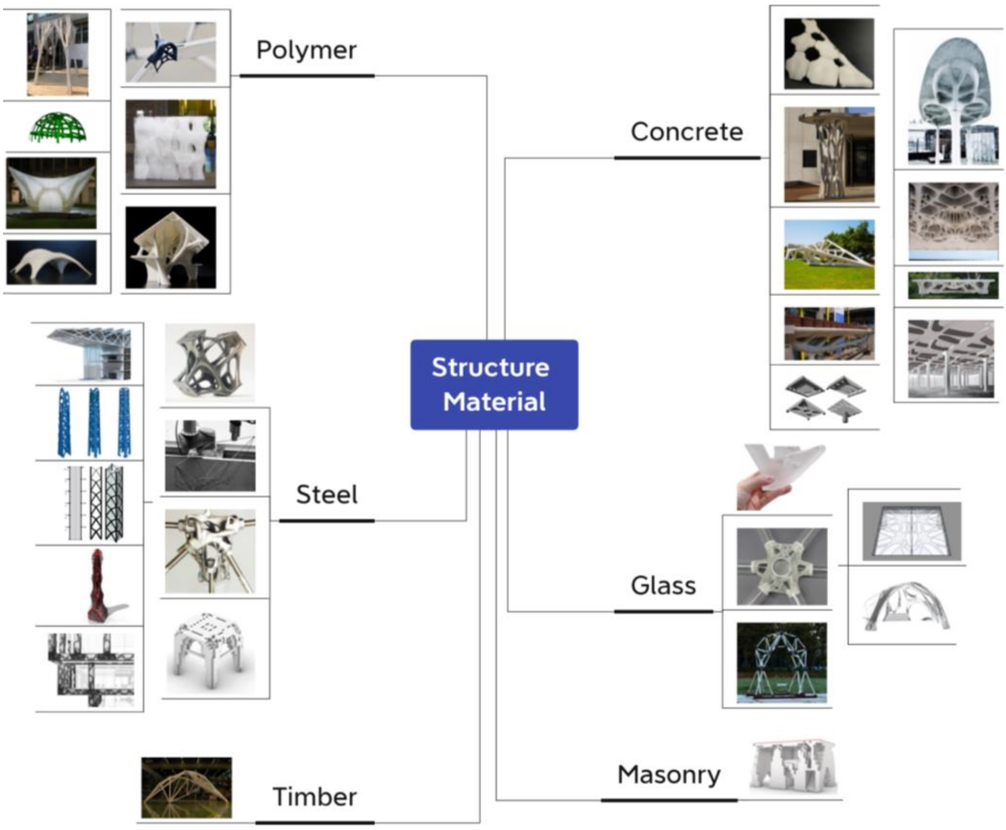


Figure 3.5. Research categorized based on material

When it comes to fabrication, the research ranged from projects not including any fabrication process, to scaled prototype fabrication to true-to-size manufactured projects (Figure 3.6.). Fabricated projects varied in their methods but can be generally classified as those that used some sort of mold and others that directly fabricated the elements.

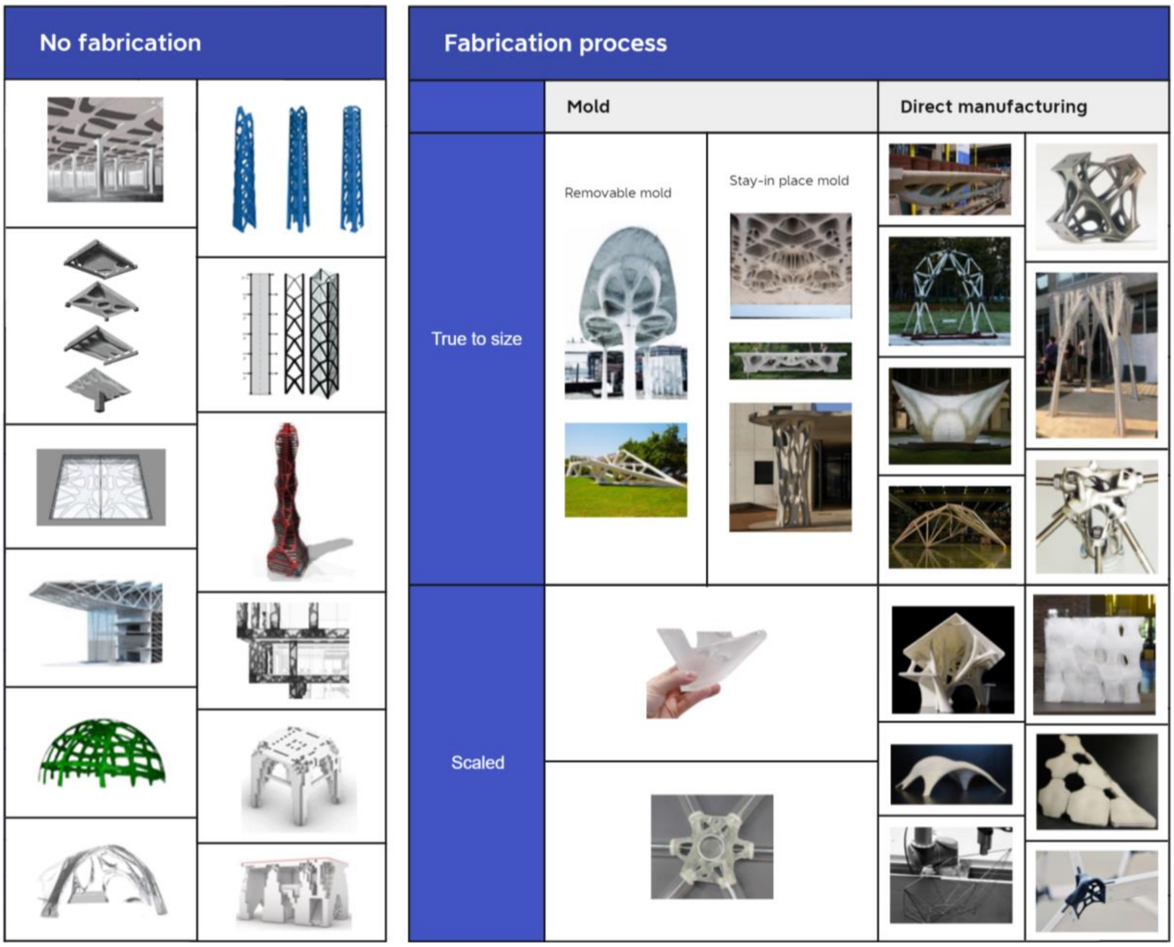
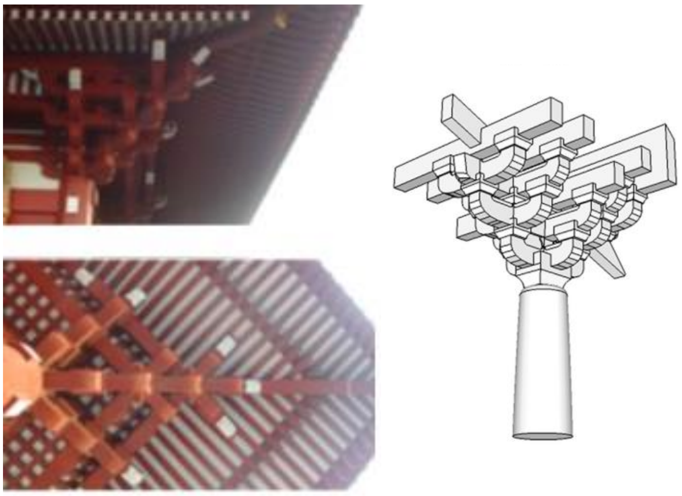


Figure 3.6. Research categorized based on fabrication

This chapter explored available research on the different applications of topology optimization in the architectural field. It was established that there is a clear potential for applying TO in the field to produce novel structures. However, minimal research has explored the impact of various design and boundary parameters on the final form. This research aims to explore precisely this influence using tree-like structures as a case study.

4. TREE-LIKE STRUCTURES

Tree-like structures are one of the most significant structures relating to biomimicry which is concerned with studying and learning from nature. At first, the inspiration from nature was limited to decorative elements such as floral decorations added to column heads. Eventually, the mechanical and structural aspects of natural elements started to be explored. With advances in computational and fabrication tools, it became much more feasible to study and construct such structures. Tree-like structures, also known as branching, dendritic or arboreal structures, are found in Chinese Dougong brackets. It is one of the earliest known examples of a structure having a sequence of cantilevers representing those of a tree and having a load transmission path similar to it. Antonio Gaudi, however, was one of the first to explore complex structurally rational architectural forms inspired by nature. In his famous work, Sagrada Familia, tree-like columns support the structure above it in a resemblance of a forest (Figure 4.1). Gaudi used physical models to study the structure and its load transmission and produced one of the first and greatest examples of tree-like concrete structures (Rian & Sassone, 2014). The main issue that these structures resolve is the transmission of load both vertically and horizontally through its trunk and branches-like parts. Multiple parameters are considered including the number of branches, angles, and lengths (Arslan Selçuk et al., 2022).



(a)

(b)

Figure 4.1. (a) Chinese Dougong brackets (b) Sagrada Familia Church, Gaudi (Lauriola, 2017; Rian & Sassone, 2014)

Later, Frei Otto studied lightweight structures including branching structures in which he addressed the structural significance of tree-like structures. Otto is known to be one of the first to introduce methods that systematically implement nature-inspired forms. He introduced methods such as the “minimum path system” which results in an increased load-bearing capacity of the structure while minimizing material consumption. Multiple works of Calatrava such as Orient Station also represent and explore the possibilities of tree-like structures (Figure 4.2). Orient Station was unique in that the branch-like elements were curved compared to other projects, such as Stuttgart Airport terminal building, that include linear branches (Arslan Selçuk et al., 2022; Rian & Sassone, 2014). The Stuttgart Airport columns seem to also branch at nearly the middle while the orient station’s columns start branching much closer to the base.

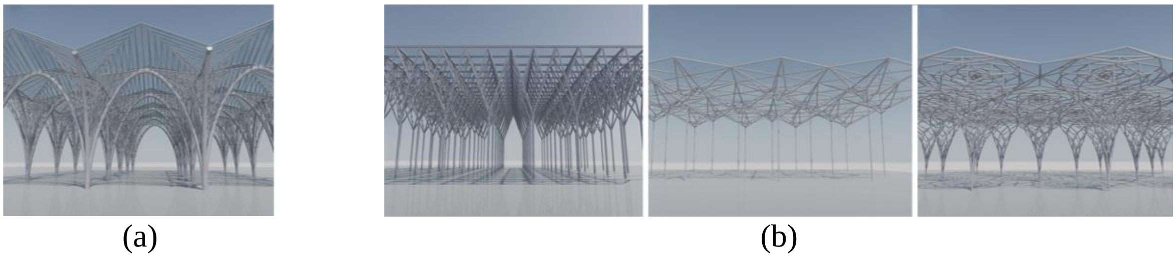
(a) (b)

Figure 4.2. (a) Calatrava’s Orient Station (b) Stuttgart Airport terminal building (Abdelsabour, 2019; Joye, 2007)

On a smaller scale, Agkathidis and Brown (2013) discuss the design, optimization, and manufacturing procedure of a canopy resembling a tree-like structure (Figure 4.3). The form of the canopy along with its structural behavior is similar to those found in nature while the planar mesh was designed based on Voronoi principles.

Figure 4.3. Tree-like canopy (Agkathidis & Brown, 2013)

Apart from built projects, tree-like structures have been studied in the literature using different methods and from multiple aspects. Multiple studies, however, have a focus on studying such structures using new computational tools. Freitas and Leitão (2019) creates a simulation of Calatrava's Orient Station using computational tools as a case study to explore and compare how different parameters affect the design (Figure 4.4.). Taking the original design as a base, multiple design variations that ensure similar structural efficiency were created after conducting structural analyses including displacement calculations. Arslan Selçuk et al. (2022) also reproduce and evaluate tree-like structures based on Frei Otto's "minimum path system" method using parametric tools. As the results are analyzed based on strain and displacement values, it was observed that applying such a method allowed to produce tree-like structures with better structural performance. The study concluded by discussing the potential parametric tools have in producing more advanced and diverse tree-like structures. In another study, tree-like structures are explored as a preliminary study to establish an algorithm that could be used to produce optimized bar structures (Dixit et al., 2021). The study simulated tree-like structures based on the bending moment which was minimized during the process and argued that following such a method would allow for more efficient cantilever tree-like structures.



(a) (b)

Figure 4.4. Orient Station studies (a) original design rendering (b) Different case studies (Freitas & Leitão, 2019)

Other studies takes use of further advances in technology such as artificial intelligence to explore other methods of designing tree-like structures. Du et al. (2022) uses machine learning (ML) to solve shape-finding problems of such structures. It uses ML to find the load-bearing center of each unit to then find the optimal coordinates of the tree-like structure's nodes. This allows for finding the best solution to the design problems whether they include simple or complex conditions such as non-uniform loads or irregular curved roofs (Figure 4.5).

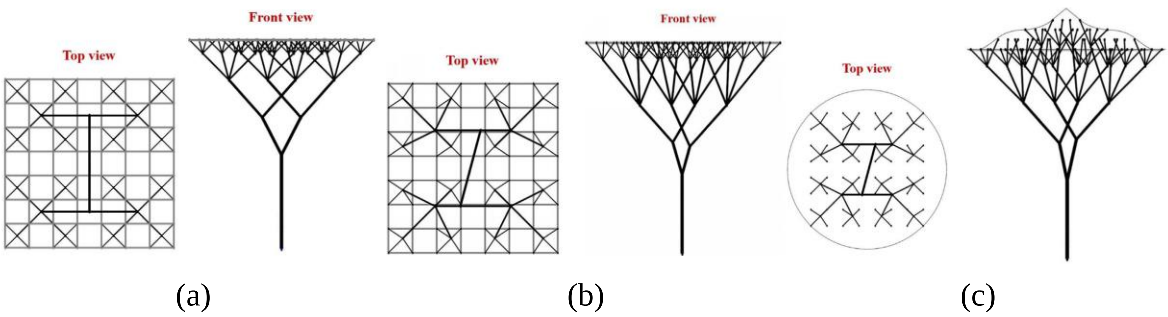


Figure 4.5. Tree-like structures produced using ML shape-finding method (a) uniform load, horizontal roof (b) non-uniform load, horizontal roof (c) non-uniform load, irregular curved roof (Du et al., 2022)

In further exploration of tree-like structures, some studies including Wang et al. (2021) explores specifically the design of joints connecting the elements of tree-like structures. The cited study introduced a method of designing the joints based on generative design and additive manufacturing. It applies the method to design a three-branch joint and analyzes the results which concludes by describing the potential of applying such a combined method in producing more effective tree-like structure joints.

4.1. The Use of TO in Tree-like Structures

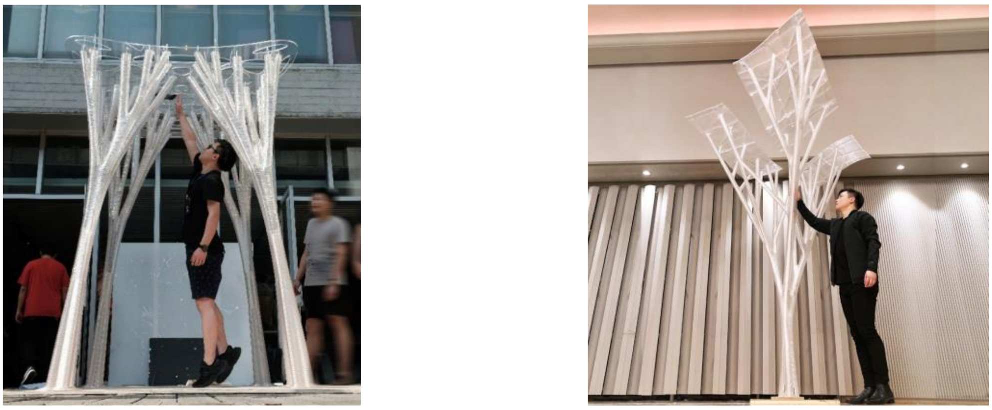
Topology optimization generally produces structures that could be associated with and compared to those in nature. Thus, it is convenient to use TO to produce tree-like structures which resemble those found in nature.

The previously mentions Qatar National Convention Centre (QNCC) project is a large-scale implementation of tree-like structures obtained with the use of TO (Figure 4.6). The structure was designed to resemble Sidra trees which are locally famous in Qatar

(Januszkiewicz & Banachowicz, 2017). It is an example of a tree-like structure branching at the base opposite to other structures going straight up to a certain level before starting to branch.

Figure 4.6. Qatar National Convention Centre (Dapogny et al., 2017)

Bao et al. (2020) exhibits a process of optimizing and fabricating a tree-like pavilion (Figure 4.7.). Two versions of the pavilion are produced in which the first is simple with a flat top and the other is more complex with varying heights and curvatures of top surfaces. Along with the steps taken during the design and optimization stages, the article discussed the fabrication of the pavilion.



(a) (b)

Figure 4.7. Tree-like pavilion (a) version 1 (b) version 2 (Bao et al., 2020)

Other studies not involving a fabrication process can be found in literature (Figure 4.8.). Frattari et al. (2013) explores the process of designing a bridge using topology optimization to maintain a tree-like structure. In another study, tree-like structures are explored as one of the case studies on implementing TO for producing structures that combine efficiency and aesthetics (Lauriola, 2017). The bridge tree-like elements seem to branch somewhere in the middle while it branches almost at the base in the other case study.

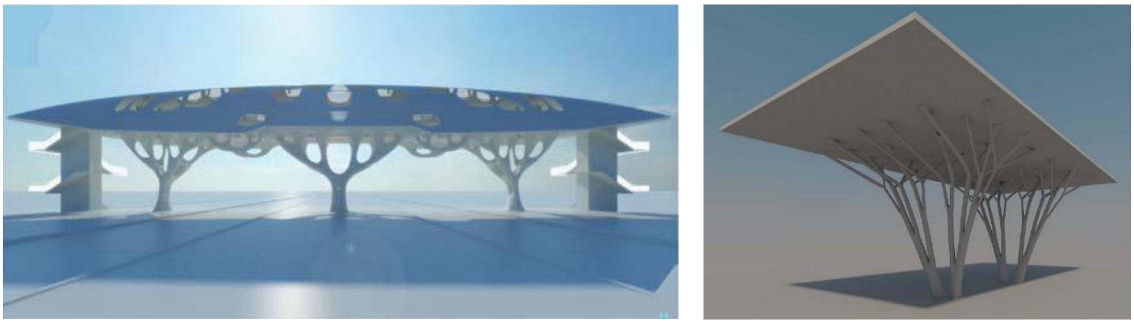


Figure 4.8. Research on tree-like structures (Frattari et al., 2013; Lauriola, 2017)

Another study presents a method for applying topology optimization in designing dendriform structures (Peng, 2016). It provides multiple 2D and 3D examples that explore the different aspects and parameters including height, weight ratio, roof geometry, etc. (Figure 4.9). It concluded that it is feasible to use topology optimization methods targeted toward continuum structures to design tree-like ones.

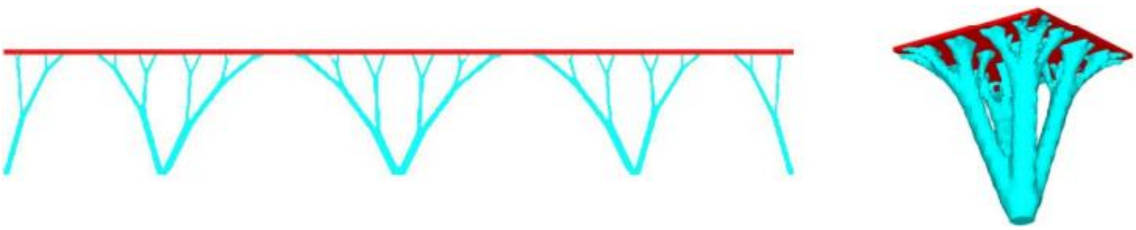


Figure 4.9. Examples of results obtained using the followed research method (Peng, 2016)

Similar to the previously mentioned study by Wang et al. (2021), another study explores the design of tree-like structure joints (Wang et al., 2020). This study differs in that TO is used as a design method and its result is compared with other joint types (Figure 4.10). The three joints are analyzed using displacement and stress calculations in which the TO joint was found to be having better structural performance while using less material.

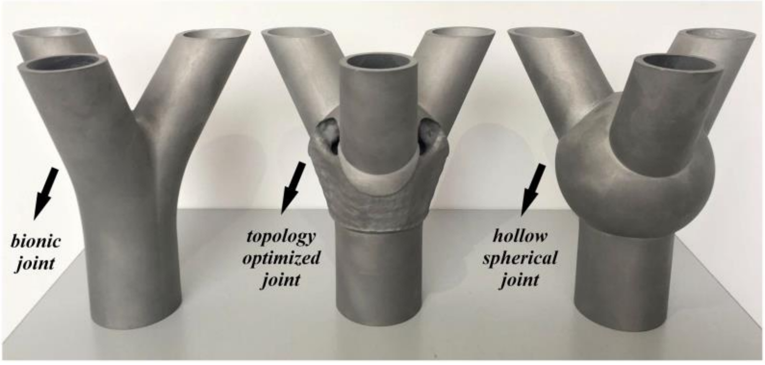


Figure 4.10. 3D printed joints (Wang et al., 2020)

It can be seen in this chapter that the exploration and applications of tree-like structures take many angles. Some research works on more of a discrete element approach while others explore the designing of the structure as a whole (more of a solid continuum structure). Few examples were found studying the application of TO for designing tree-like structures. With that being said, the literature lacks a comprehensive study that explores the effect of different design decisions on the form of tree-like structures. Thus, this study examines topology optimized tree-like structures by using a variation of column and slab decisions. A focus on the structure as a whole is taken as an approach in his research. Solid domains varying in their column number and angle of branching are introduced and the resulting geometries are analyzed and discussed.

5. METHOD OF STUDY

Throughout history, tree-like structures have been explored and designed using various methods. This research examines the use of topology optimization to design such structures and explores the effect of different parameters on the resulting form. Multiple studies are conducted using different domains, load types, and other parameters. The case studies are divided into two parts, the first includes research on the available tools for TO to determine which is more suitable for the research. After deciding on the tool to be used, the second part explores the application of TO for tree-like structures (Figure 5.1).

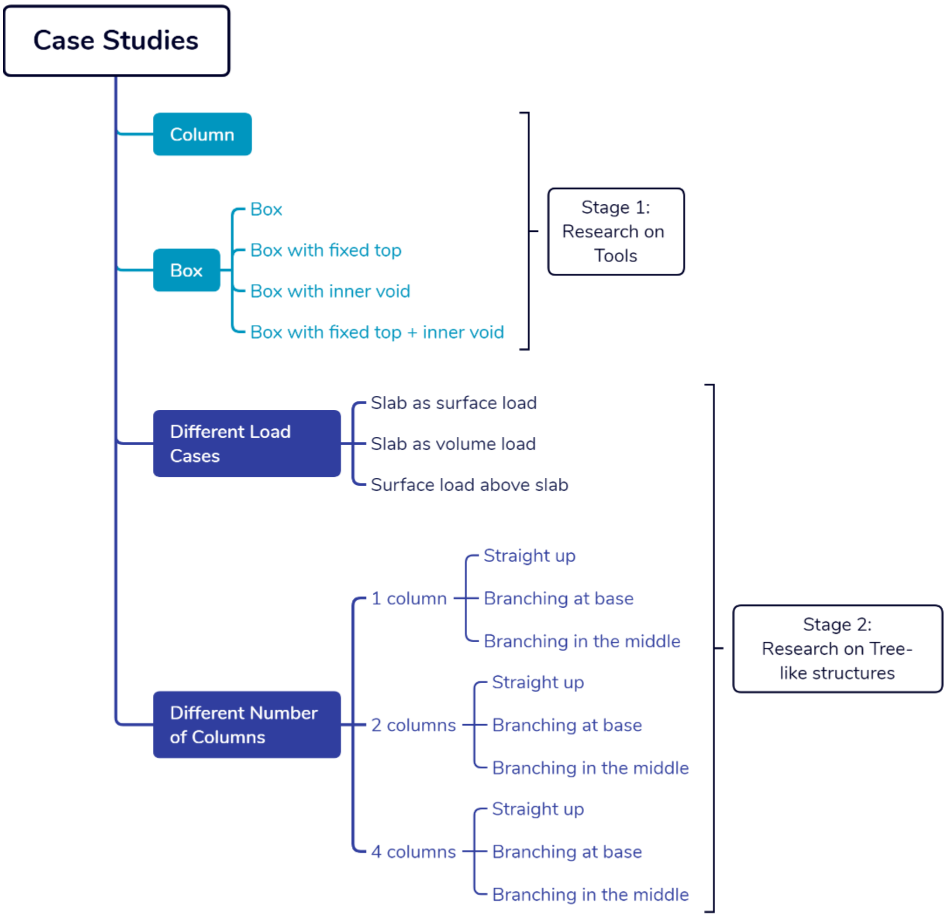


Figure 5.1. Graph showing the different case studies

Multiple programs can be used for exploring TO in architecture. Hence, the three grasshopper plugins: Millipede, tOpos, and Ameba are studied and compared as a first stage (Figure 5.2). The design domain is set up using Rhino and Grasshopper. TO process is conducted for each case study in the three programs. Other tools in Rhino and Grasshopper are later used to smooth and analyze the results. The Grasshopper plugin

Dendro is used for smoothing the results and Simscale, a cloud simulation software is used for the analysis of the results. Before the resulting geometries can be analyzed in Simscale, they need to be transferred from meshes to polysurfaces, and any overlapping or self-intersecting surfaces need to be removed. This pre-analysis processing of the geometries is done in Rhino.

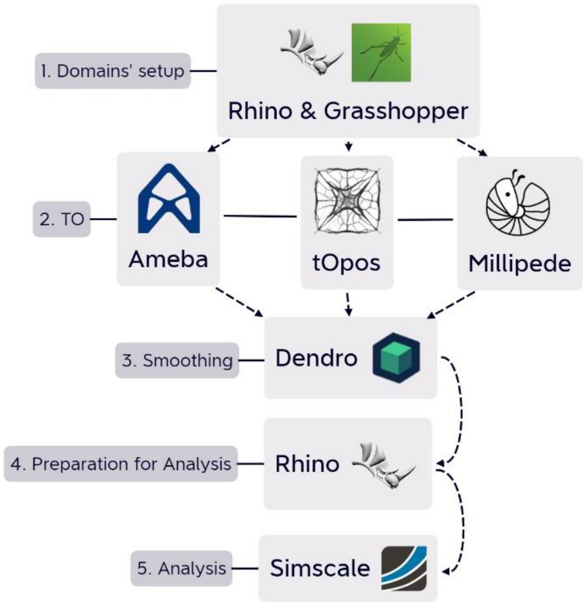
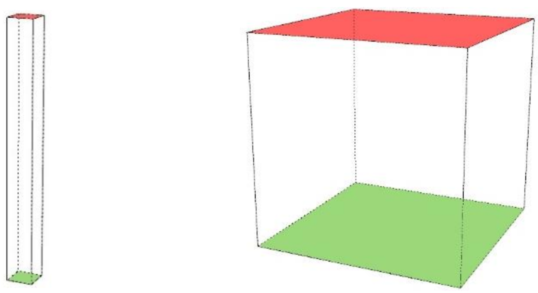


Figure 5.2. Used programs for stage 1 of the study

To explore the best-suited plugin for this research, domains are chosen to represent the simplest form of design spaces that could be used to produce tree-like structures. A column and a simple box are used for the case studies and are sized proportionally for their use in a building (Figure 5.3).

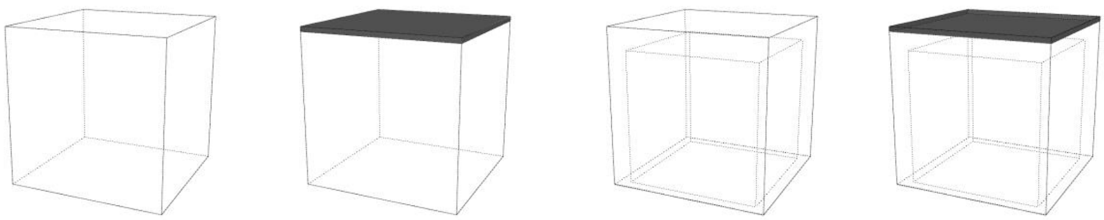


*Detailed sizing and loads are provided within the case study part

(a) (b)

Figure 5.3. Case studies' domains showing the specified load and support area; (a) Column (b) Box

The box, which is used as a representation of architectural space, is diversified by identifying solid and void areas within it to further explore the abilities of the 3 plugins (Figure 5.4). The box is simply defined in the first case with no other restrictions. In the second case, a top slab of 0.3 m thickness is defined as a fixed solid area which means it will be a non-design area filled with material. In the third case, the inner part of the box is removed which leaves 0.75 m thick side walls and a 1.5 m thick top. For the final case, a fixed solid area is added to the emptied box. In addition to the fixed domains, the selected material is fixed as concrete and is identified by the properties: Young's modulus (E) =48 GPa, and Poisson ratio (ν)=0.2.



(a) (b) (c) (d)

Figure 5.4. Box variations used for in this research (a) Box (b) Box with a fixed top (c) Box with inner void (d) Box with an inner void and fixed top

The basic code for topology optimization in Millipede can be seen in Figure 5.5. The main components of the domain, load, support, and material are defined in the model which is used for the optimization. Multiple volumes can be identified as part of the domain with the possibility of specifying if the volume represents a void area. The load is also identified as a volume and its value is specified in N/m^3. The support type can be identified by setting the movement and rotation possibilities in the three directions. The optimization iteration and targeted volume fraction are specified after the model is built.

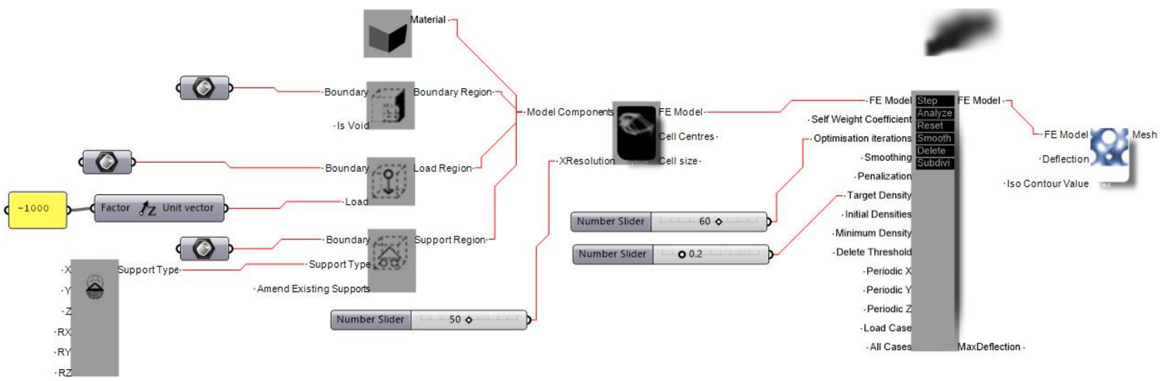


Figure 5.5. Basic TO code in Millipede

The basic code for tOpos is pretty similar to the Millipede's (Figure 5.6). It contains defining the model using domain and boundary conditions which include the load and support. The load type (between the four previously discussed types) and load value are identified along with support which has no options to change its type. It is easy to switch between the GPU and CPU modes and to start, stop and reset the computation process.

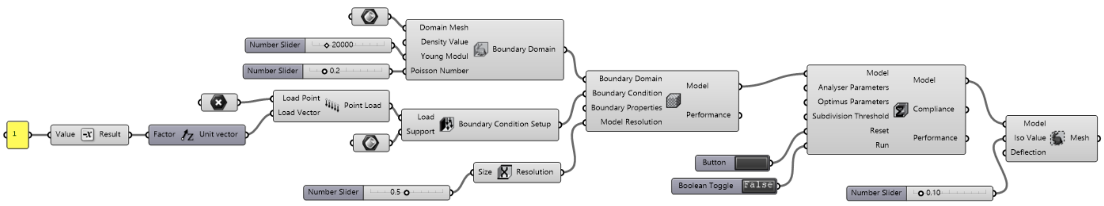


Figure 5.6. Basic TO code in tOpos

Ameba's code works somewhat differently compared to the other two (Figure 5.7). The domain is defined first as a Brep from which the load and support surfaces are defined and then used to create the model to be analyzed and optimized. The load is identified as a surface and its value is given in N/mm^2. The support has the option to be set as fixed or allow movement same as in Millipede. Material properties and other optimization settings can be edited in the pre-info settings.

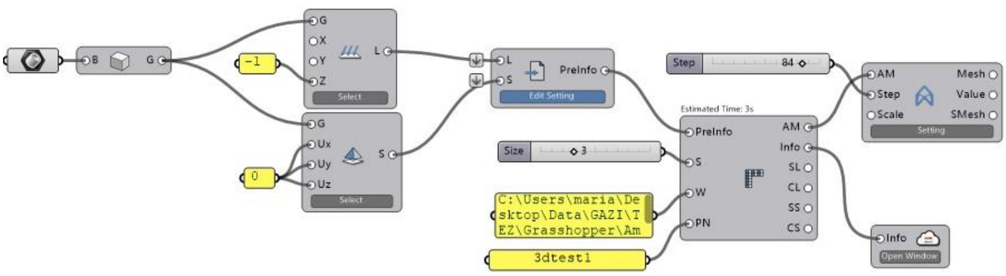


Figure 5.7. Basic TO code in Ameba

The area in green in Figure 5.3 represents the support that is provided as a fixed support in all cases. The red area represents the load in which its volume and magnitude are fixed for each case. Although loads are specified as volumes in Millipede and as surfaces in Ameba, the sizing and load value are calculated to provide the same amount of load for all the plugins. tOpos allows the identification of load in multiple ways including surface and volume. A surface is used as a load for tOpos throughout this research (Figure 5.8).

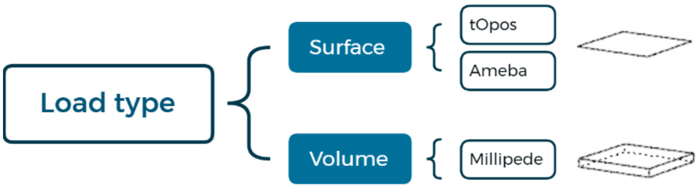


Figure 5.8. Load types

The plugins are studied and compared based on different parameters (Figure 5.9). How the domain and design parameters are set is one aspect. Analysis of the resulting geometry including its volume fraction, structural performance, and visual standpoint is also included in the comparison. The computation time is another key comparison parameter considered in all case studies.

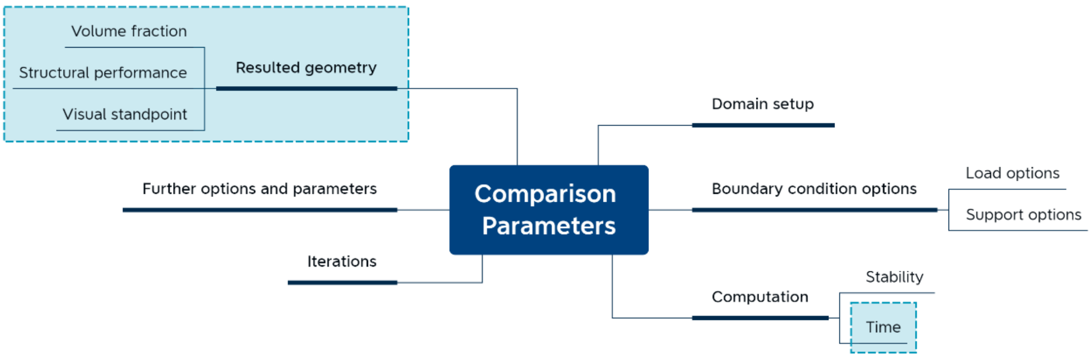


Figure 5.9. Comparison parameters between the 3 plugins

For the second part of the case studies, the application of topology optimization in designing tree-like structures is explored. The used programs for the second part are shown in Figure 5.10. This stage differs in that the TO process is done using one plugin instead of three as the plugin found most suitable is used to explore the tree-like structures. Also, Dendro is not used in this stage as tOpos's results are not in much need of smoothing compared to results from the other two plugins.

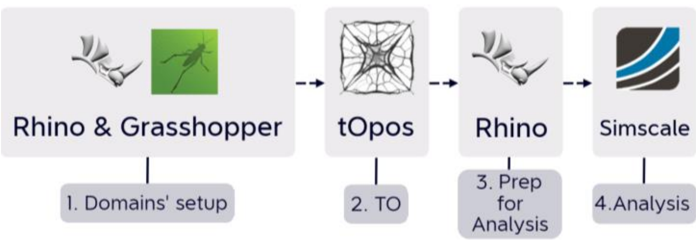


Figure 5.10. Used programs for stage 2 of the study

In these studies, diverse column types, which are the basis of tree-like structures, are examined using tOpos plugin. In the first stage, a single column is studied using different loading types to determine the load type to be used for the main case studies. Subsequently, a single column is used in the main case study, and step by step the number of columns and their form and connections are changed and the variation in the results is investigated. A single column, 2 columns, and 4 columns are used for the case study (Figure 5.11). In each case, three different configurations are used: a straight-up column, a branching at the base, and a branching in the middle configuration.

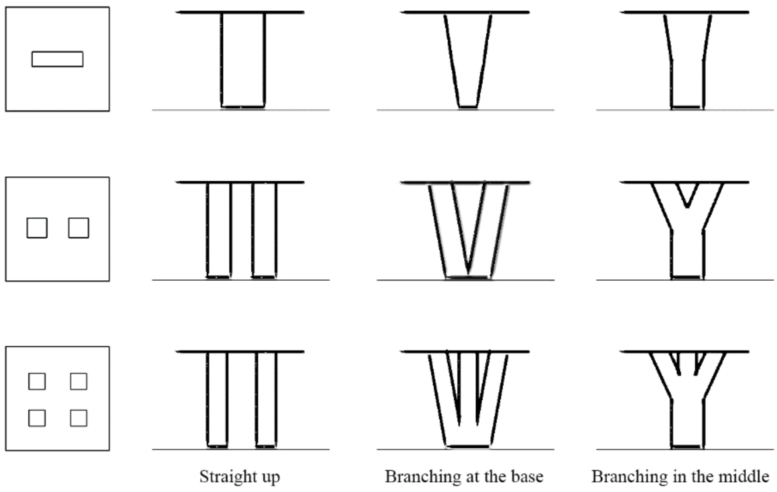


Figure 5.11. Column configurations

The domain's height is fixed to 10 m in all the case studies with a 10×10× 0.5 m slab at the top. An area of 4 m^2 is fixed for all the column tops throughout the different case studies (Figure 5.12). The single square column used for exploring the different load cases is of a 2×2 m base, the single rectangular column in the second stage is of a 1×4 m top, the two columns are of a $\sqrt{2} \times \sqrt{2}$ m top and the four columns are of a 1×1 m top.

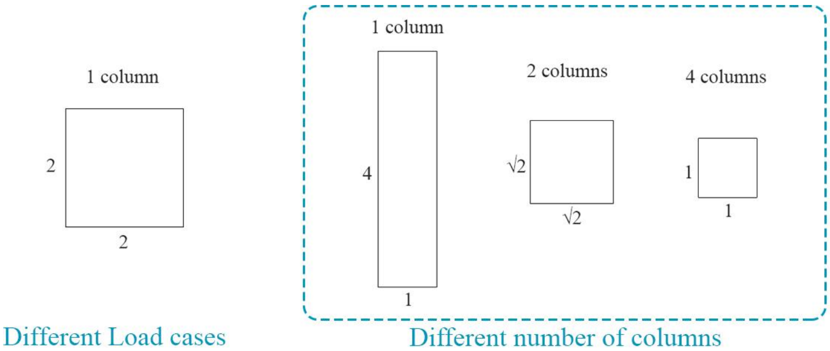


Figure 5.12. Domains' column ratios

The results of the 9 main cases seen in Figure 5.11 are then analyzed and compared based on multiple factors (Figure 5.13). Mainly, the resulting forms are compared visually along with a simple structural analysis of stress and displacement conducted using Simscale. Visually, the results are compared based on the number of columns that the optimization process produced, the inclination of these columns, whether they are connected to each other, and in which location.

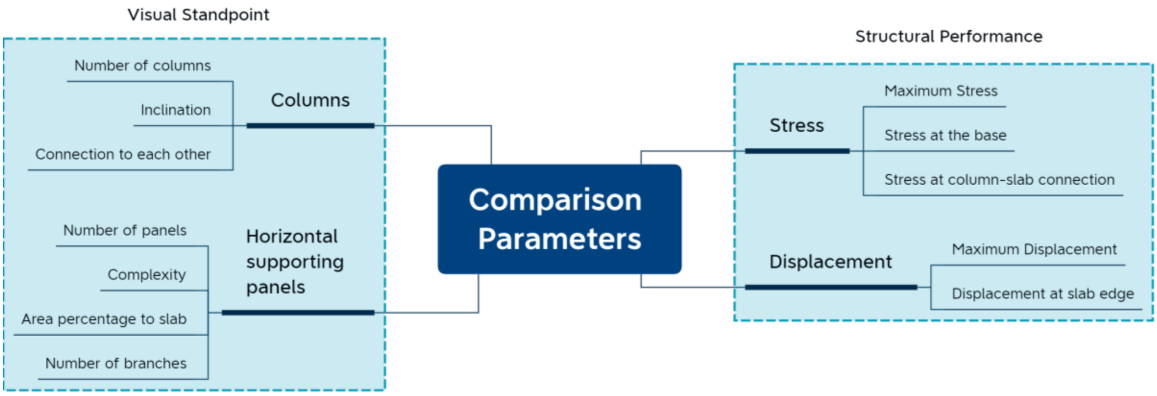


Figure 5.13. Comparison parameters between tree-like structures

6. CASE STUDIES

The main case study in this research explores producing tree-like structures using topology optimization as a design method. Before moving into the main case study, a preliminary case study regarding the available grasshopper plugins is conducted to measure which one is more suitable for the exploration of tree-like structures.

6.1. Research on TO Tools

The 3 grasshopper plugins Millipede, tOpos, and Ameba are studied using 2 main case studies, a column and a box. Four variations of the box are explored as part of studying the efficiency of all the plugins (Figure 6.1).

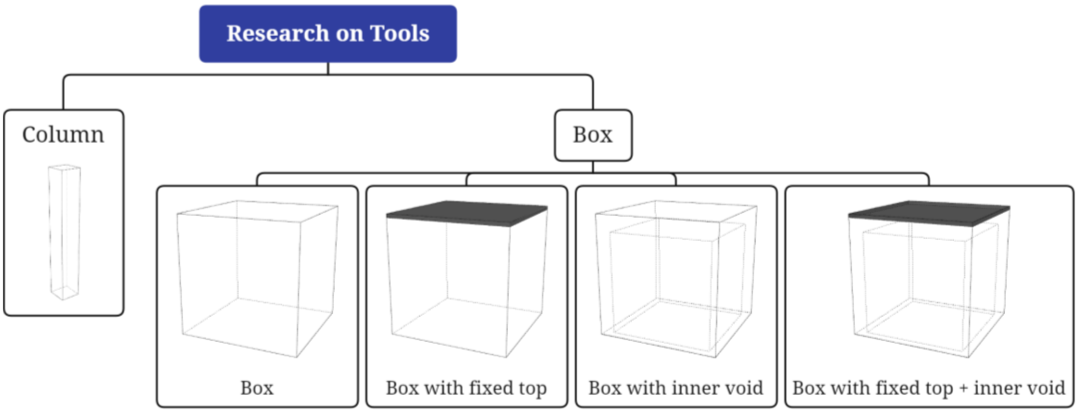
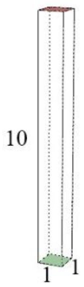
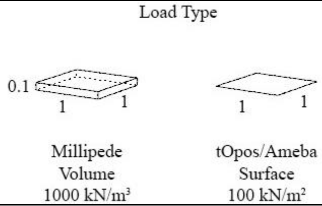


Figure 6.1. Case Studies used for exploring grasshopper plugins

A column of 1 by 1 m base and 10 m height is used as a first case study. The domain and parameters are shown in table 6.1. In order to see the formation stages of architectural design structures within the scope of the study, the results of different iteration numbers were recorded. The obtained results give the opportunity to better analyze the plugins used in different design constraints.

Table 6.1. Column’s domain and set parameters

<table>
<tr><td rowspan="3"></td><td>Dimensions (m)</td><td>Load (kN)</td><td>Target Volume Fraction</td></tr>
<tr><td>1x1x10</td><td>100</td><td>0.20</td></tr>
<tr><td colspan="3"></td></tr>
</table>

The results of the column case study in the three plugins can be seen in Table 6.2. Five different results are obtained for each plugin as the maximum number of iterations is set to 20, 40, 60, 80, and 100 respectively to show the evolution of the form as the number of iterations increases. Ameba's results show the most progression and change as the iterations increase while Millipede showed the least change.

Table 6.2. Comparison between the results in the 3 plugins (Left: original results, right: smoothed results)

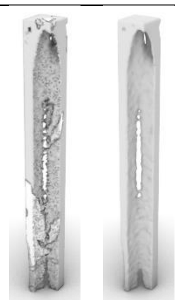
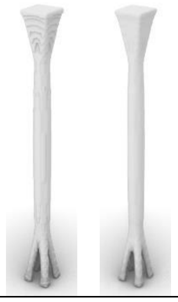
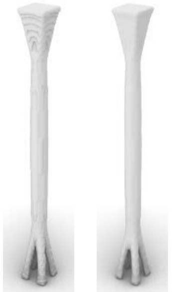
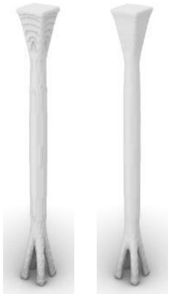
<table>
<tr><th>Millipede</th><th>tOpos</th><th>Ameba</th></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="3">20 iterations</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="3">40 iterations</td></tr>
</table>

Table 6.2. (Continuation) Comparison between the results in the 3 plugins (Left: original results, right: smoothed results)

		
60 iterations		
		
80 iterations		
		
100 iterations		98 iterations

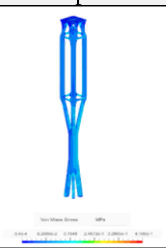
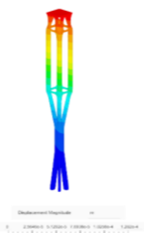
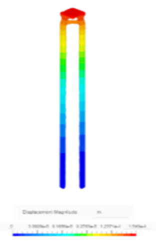
The reached iterations, volume fraction, and computation time for all the variations of the column case are shown in Table 6.3. Ameba stopped at 98 iterations when the maximum number given is 100 as it reached its goal while tOpos and Millipede reached the maximum number given. When looking at the volume fraction, Ameba reached the required volume fraction of 0.2 at the 80th iteration while tOpos reached the same volume fraction at the 100th iteration case. Millipede, however, failed to reach the specified volume fraction as the 100th iteration had a volume fraction of 0.28. tOpos was the fastest in computation as it sometimes took seconds to compute while Ameba needed a few minutes. Millipede took the most time spending hours computing.

Table 6.3. Comparison between the reached volume fraction and optimization time of each plugin

	Max iterations:20		
	Reached iterations	Reached volume fraction	Time
Millipede	20	0.31	1 h 6 min
tOpos	20	0.19	22 sec
Ameba	20	0.56	4 min
	Max iterations:40		
	Reached iterations	Reached volume fraction	Time
Millipede	40	0.29	2 h 7 min
tOpos	40	0.21	38 sec
Ameba	40	0.35	8 min
	Max iterations:60		
	Reached iterations	Reached volume fraction	Time
Millipede	60	0.29	3 h 6 min
tOpos	60	0.21	52 sec
Ameba	60	0.23	12 min
	Max iterations:80		
	Reached iterations	Reached volume fraction	Time
Millipede	80	0.28	4 h 6 min
tOpos	80	0.21	1 min 9 sec
Ameba	80	0.20	16 min
	Max iterations:100		
	Reached iterations	Reached volume fraction	Time
Millipede	100	0.28	5 h 16 min
tOpos	100	0.20	1 min 22 sec
Ameba	98	0.20	20 min

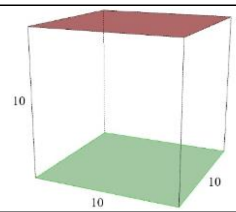
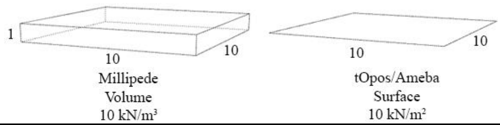
Final results that were given a maximum iteration number of 100 are re-meshed and exported to Simscale for analysis. The maximum Von Mises Stress and Displacement are shown in Table 6.4.

Table 6.4. Max Von Mises Stress and Displacement of results

	Millipede	tOpos	Ameba
Von Mises Stress (MPa)			
	Max; 0.305	Max; 0.411	Max; 0.742
Displacement (m)			
	Max; 0.156×10^{-4}	Max; 1.282×10^{-4}	Max; 1.546×10^{-4}

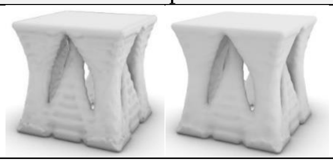
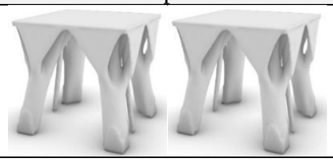
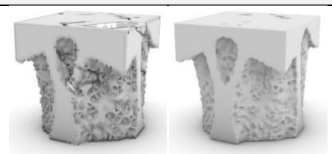
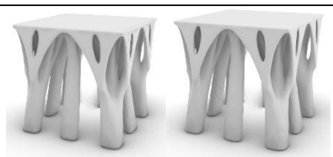
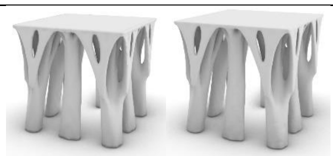
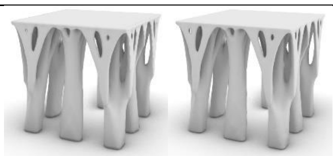
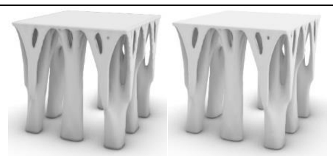
The box is used to produce 4 different studies with a slightly different iteration each time. In the first case study, the box is simply defined as a domain with values shown in Table 6.5.

Table 6.5. Box domain and set parameters

	Dimensions (m)	Load (kN)	Target Volume Fraction
	10x10x10	1000	0.20
	Load Type 		

The results of the box case in the three plugins are shown in Table 6.6. Same as the last case, five different iteration numbers are given to show the progression of the results throughout the iterations.

Table 6.6. Comparison between the results in the 3 plugins (Left: original results, right: smoothed results)

Millipede	tOpos	Ameba
		
20 iterations		
		
40 iterations		
		
60 iterations		
		
80 iterations		
		
100 iterations		99 iterations

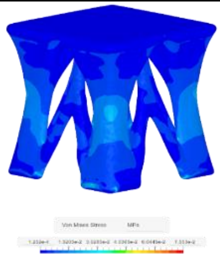
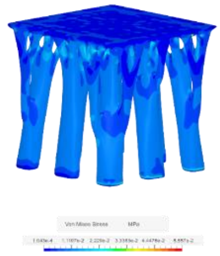
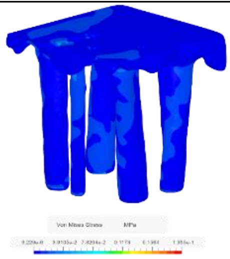
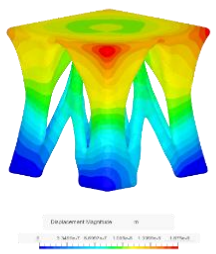
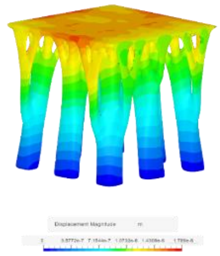
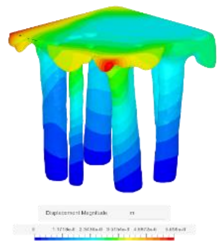
Data regarding the iterations number, volume fraction, and time of computation for the box case is shown in Table 6.7. Once again, Millipede took the most time computing, and its computation time was longer than the column case. This can be associated with the difference in domain's volume. This means that Millipede's computation time is highly affected depending on the sizing of the domain while Ameba and tOpos had almost the same time. The two plugins reached the required 0.2 volume fraction while Millipede failed to do so.

Table 6.7. Comparison between the reached volume fraction and optimization time of each plugin

	Max iterations:20		
	Reached iterations	Reached volume fraction	Time
Millipede	20	0.36	1 h 52 mins
tOpos	20	0.27	17 sec
Ameba	20	0.54	8 mins
	Max iterations:40		
	Reached iterations	Reached volume fraction	Time
Millipede	40	0.29	3 h 37 mins
tOpos	40	0.21	30 sec
Ameba	40	0.35	12 mins
	Max iterations:60		
	Reached iterations	Reached volume fraction	Time
Millipede	60	0.28	5 h 20 mins
tOpos	60	0.20	46 sec
Ameba	60	0.24	17 mins
	Max iterations:80		
	Reached iterations	Reached volume fraction	Time
Millipede	80	0.28	6 h 59 mins
tOpos	80	0.20	53 sec
Ameba	80	0.21	21 mins
	Max iterations:100		
	Reached iterations	Reached volume fraction	Time
Millipede	100	0.28	8 h 58 mins
tOpos	100	0.20	1 min 5 sec
Ameba	99	0.20	24 mins

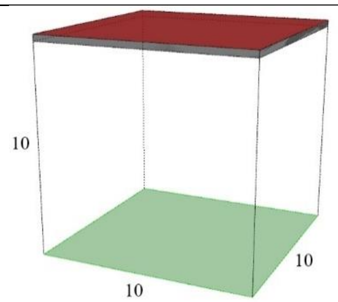
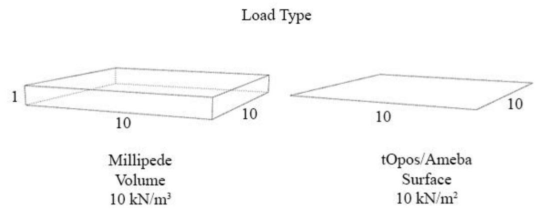
Same as in the column case, the final results are taken into Simscale for calculation of maximum von mises stress and displacement values (Table 6.8). Although values were generally very close in both cases of stress and displacement, Ameba's results were the largest.

Table 6.8. Max Von Mises Stress and Displacement of results

	Millipede	tOpos	Ameba
Von Mises Stress (MPa)			
	Max; 7.553 × 10^{-2}	Max; 5.557 × 10^{-2}	Max; 19.55 × 10^{-2}
Displacement (m)			
	Max; 1.675 × 10^{-6}	Max; 1.789 × 10^{-6}	Max; 5.859 × 10^{-6}

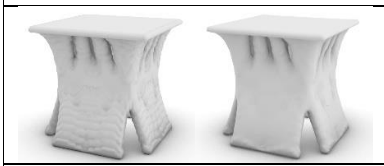
The same box is optimized with the same parameters again but with a small change of providing a 0.3 m thick fixed solid top layer (Table 6.9).

Table 6.9. Box domain and set parameters

	Dimensions (m)	Load (kN)	Target Volume Fraction
	10x10x10	1000	0.20
			

The results from each plugin are shown in Table 6.10. Ameba's result changed the most in comparison to the previous case of the simple box. The slab is filled fully with no holes and only four thick columns carry it opposite to the 6 thinner differently shaped columns. Millipede's result looks bulkier while tOpos's include more of inclined connecting column-like structures rather than the stand-alone ones in the previous trial.

Table 6.10. Comparison between the results in the 3 plugins (Left: original results, right: smoothed results)

Millipede	tOpos	Ameba
		
20 iterations		
		
40 iterations		
		
60 iterations		
		
80 iterations		
		
100 iterations		91 iterations

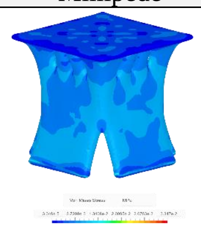
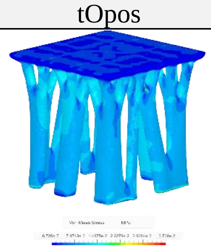
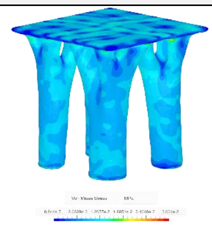
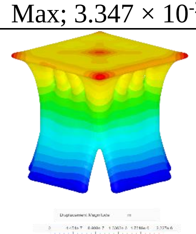
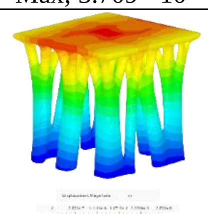
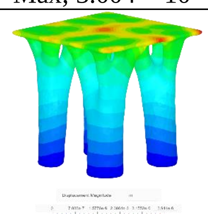
Same as in the previous trials, the computation time and reached iteration and volume fractions are shown in Table 6.11. Looking at the volume fraction, Millipede's result, although seemed bulkier, is actually slightly less volume fraction than the previous trial although it failed to reach the required fraction of 0.2. Ameba and tOpos did reach the 0.2 required volume fraction. The computation time was extremely similar to the earlier simple box case in each plugin.

Table 6.11. Comparison between the reached volume fraction and optimization time of each plugin

	Max iterations:20		
	Reached iterations	Reached volume fraction	Time
Millipede	20	0.30	1 h 58 mins
tOpos	20	0.21	16 sec
Ameba	20	0.58	6 mins
	Max iterations:40		
	Reached iterations	Reached volume fraction	Time
Millipede	40	0.28	3 h 41 mins
tOpos	40	0.21	33 sec
Ameba	40	0.36	9 mins
	Max iterations:60		
	Reached iterations	Reached volume fraction	Time
Millipede	60	0.28	5 h 28 mins
tOpos	60	0.21	45 sec
Ameba	60	0.24	13 mins
	Max iterations:80		
	Reached iterations	Reached volume fraction	Time
Millipede	80	0.27	6 h 50 mins
tOpos	80	0.21	56 sec
Ameba	80	0.21	17 mins
	Max iterations:100		
	Reached iterations	Reached volume fraction	Time
Millipede	100	0.27	9 h
tOpos	100	0.20	1 min 9 sec
Ameba	91	0.20	22 mins

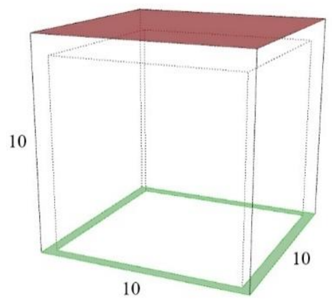
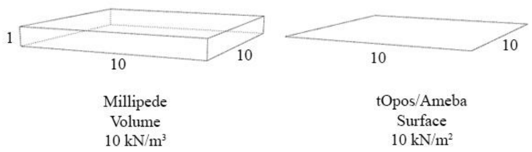
Same as in previous cases, the results analyzed in Simscale are shown in Table 6.12 which show extremely similar maximum values with Ameba having a slightly lower maximum value of stress than the other two programs. On the other side, Ameba had the highest value in displacement.

Table 6.12. Max Von Mises Stress and Displacement of results

	Millipede	tOpos	Ameba
Von Mises Stress (MPa)			
	Max; 3.347×10^{-2}	Max; 3.709×10^{-2}	Max; 3.004×10^{-2}
Displacement (m)			
	Max; 2.227×10^{-6}	Max; 2.786×10^{-6}	Max; 3.944×10^{-6}

The internal part of the box is removed which leaves a domain with sides of 0.75 m and a top of 1 m (Table 6.13). This emptied box is again optimized using the 3 plugins.

Table 6.13. Box domain and set parameters

<table>
<tr><td rowspan="3">
</td><td>Dimensions (m)</td><td>Load (kN)</td><td>Target Volume Fraction</td></tr>
<tr><td>10x10x10</td><td>1000</td><td>0.20</td></tr>
<tr><td colspan="3">Load Type

</td></tr>
</table>

The results of this case study are shown in Table 6.14. Millipede’s result is again the bulkiest with wall-like structures at the center of each side of the box. tOpos on the other hand had the same concept of laying material, especially towards the center of each side. However, the material is distributed in more of a column that thickens and spreads toward the top into smaller branches. Ameba’s included more irregularly sized and placed columns.

Table 6.14. Comparison between the results in the 3 plugins (Left: original results, right: smoothed results)

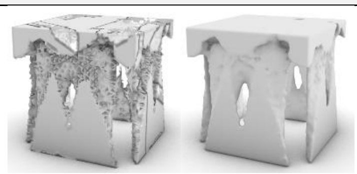
Millipede	tOpos	Ameba
		
20 iterations		
		
40 iterations		
		
60 iterations		

Table 6.14. (Continuation) Comparison between the results in the 3 plugins (Left: original results, right: smoothed results)

		
80 iterations		
		
100 iterations		

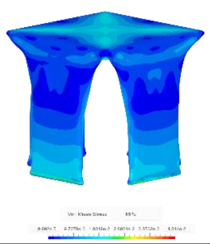
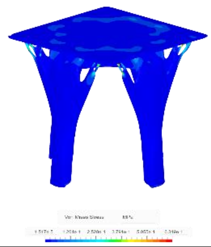
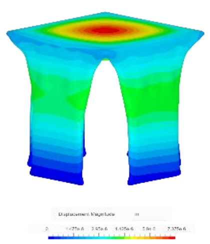
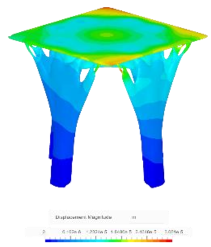
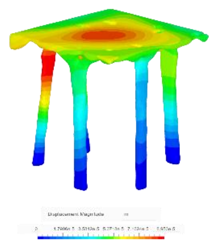
Reached volume fraction, iterations, and computation time are again shown in Table 6.15. While Ameba's result reached the required volume fraction of 0.2, Millipede's result shows the highest volume fraction of 0.62. tOpos also failed to completely fulfill the requirement but was relatively closer with a volume fraction of 0.25.

Table 6.15. Comparison between the reached volume fraction and optimization time of each plugin

	Max iterations:20		
	Reached iterations	Reached volume fraction	Time
Millipede	20	0.64	41 min
tOpos	20	0.30	8 sec
Ameba	20	0.56	5 min
	Max iterations:40		
	Reached iterations	Reached volume fraction	Time
Millipede	40	0.62	1 h 20 min
tOpos	40	0.25	12 sec
Ameba	40	0.34	9 min
	Max iterations:60		
	Reached iterations	Reached volume fraction	Time
Millipede	60	0.62	1 h 53 min
tOpos	60	0.25	16 sec
Ameba	60	0.23	13 min
	Max iterations:80		
	Reached iterations	Reached volume fraction	Time
Millipede	80	0.62	2 h 41 min
tOpos	80	0.25	20 sec
Ameba	80	0.20	18 min
	Max iterations:100		
	Reached iterations	Reached volume fraction	Time
Millipede	100	0.62	3 h 16 min
tOpos	100	0.25	22 sec
Ameba	100	0.20	22 min

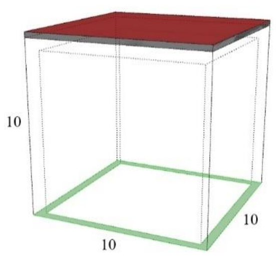
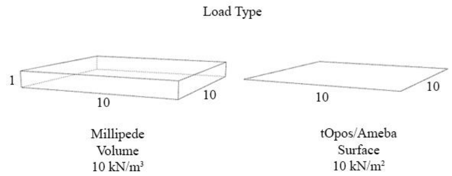
Results are again analyzed using Simscale (Table 6.16). tOpos had the highest maximum stress value while Ameba had the highest displacement value.

Table 6.16. Max Von Mises Stress and Displacement of results

	Millipede	tOpos	Ameba
Von Mises Stress (MPa)			
	Max; 0.481×10^{-1}	Max; 6.319×10^{-1}	Max; 4.229×10^{-1}
Displacement (m)			
	Max; 0.737×10^{-5}	Max; 3.081×10^{-5}	Max; 8.953×10^{-5}

The emptied box is optimized in another case study using the same concept as the second study which is having a fixed 0.3 m thick top (Table 6.17).

Table 6.17. Box domain and set parameters

<table>
<tr><td rowspan="3">
</td><td>Dimensions (m)</td><td>Load (kN)</td><td>Target Volume Fraction</td></tr>
<tr><td>10x10x10</td><td>1000</td><td>0.20</td></tr>
<tr><td colspan="3">
</td></tr>
</table>

The outcomes of this case study are presented in Table 6.18. tOpos and Millipede's results seemed fairly similar to the last case study while Ameba was different in that the slab contained no holes, and the columns were placed in a more symmetrical way in comparison to the last case.

Table 6.18. Comparison between the results in the 3 plugins (Left: original results, right: smoothed results)

Millipede	tOpos	Ameba
		
20 iterations		
		
40 iterations		
		
60 iterations		
		
80 iterations		
		
100 iterations		94 iterations

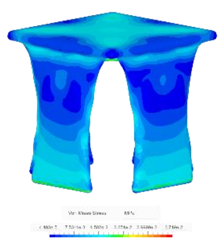
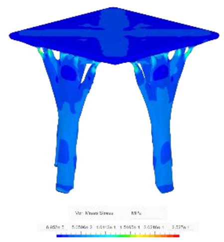
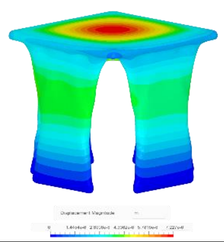
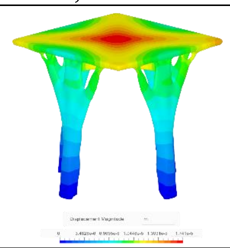
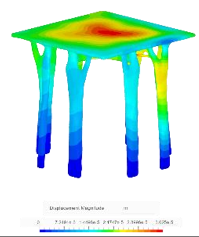
The computation time, iterations, and volume fractions are shown in Table 6.19. Volume fractions are the same as the last case study showing that results were not that different in comparison to the previous case. Computation time shows the same pattern as the previous trials with tOpos taking seconds to compute, Ameba taking few minutes and Millipede taking the longest reaching more than three hours of computation for the 100 iterations case. Ameba also stopped computing a bit earlier when given a maximum iteration number of 100.

Table 6.19. Comparison between the reached volume fraction and optimization time of each plugin

	Max iterations:20		
	Reached iterations	Reached volume fraction	Time
Millipede	20	0.63	40 min
tOpos	20	0.26	15 sec
Ameba	20	0.56	5 min
	Max iterations:40		
	Reached iterations	Reached volume fraction	Time
Millipede	40	0.62	1 h 19 min
tOpos	40	0.26	20 sec
Ameba	40	0.34	9 min
	Max iterations:60		
	Reached iterations	Reached volume fraction	Time
Millipede	60	0.62	1 h 58 min
tOpos	60	0.26	26 sec
Ameba	60	0.23	13 min
	Max iterations:80		
	Reached iterations	Reached volume fraction	Time
Millipede	80	0.62	2 h 47 min
tOpos	80	0.25	39 sec
Ameba	80	0.20	18 min
	Max iterations:100		
	Reached iterations	Reached volume fraction	Time
Millipede	100	0.62	3 h 9 min
tOpos	100	0.25	44 sec
Ameba	94	0.20	22 min

The outcomes of the analysis in Simscale can be seen in Table 6.20. Ameba had the highest maximum stress and displacement values while Millipede had the lowest.

Table 6.20. Max Von Mises Stress and Displacement of results

	Millipede	tOpos	Ameba
Von Mises Stress (MPa)			
	Max; 0.374×10^{-1}	Max; 2.527×10^{-1}	Max; 20.98×10^{-1}
Displacement (m)			
	Max; 0.722×10^{-5}	Max; 1.741×10^{-5}	Max; 3.625×10^{-5}

Based on the different case studies and comparison parameters, it was found that tOpos is the most suitable plugin for conducting the second part of this research regarding tree-like structures. It was the fastest plugin which gives the ability to conduct multiple studies and explore different parameters in a shorter time. The volume fraction was reached or almost reached in all cases. It also resulted in more nature-inspired forms compared to other plugins which fit the topic of tree-like structures.

6.2. Research on The Application of TO for Tree-like Structures

The second part of the case studies includes two stages, an initial study on the load types and a main study on different domain configurations representing treelike structures (Figure 6.2).

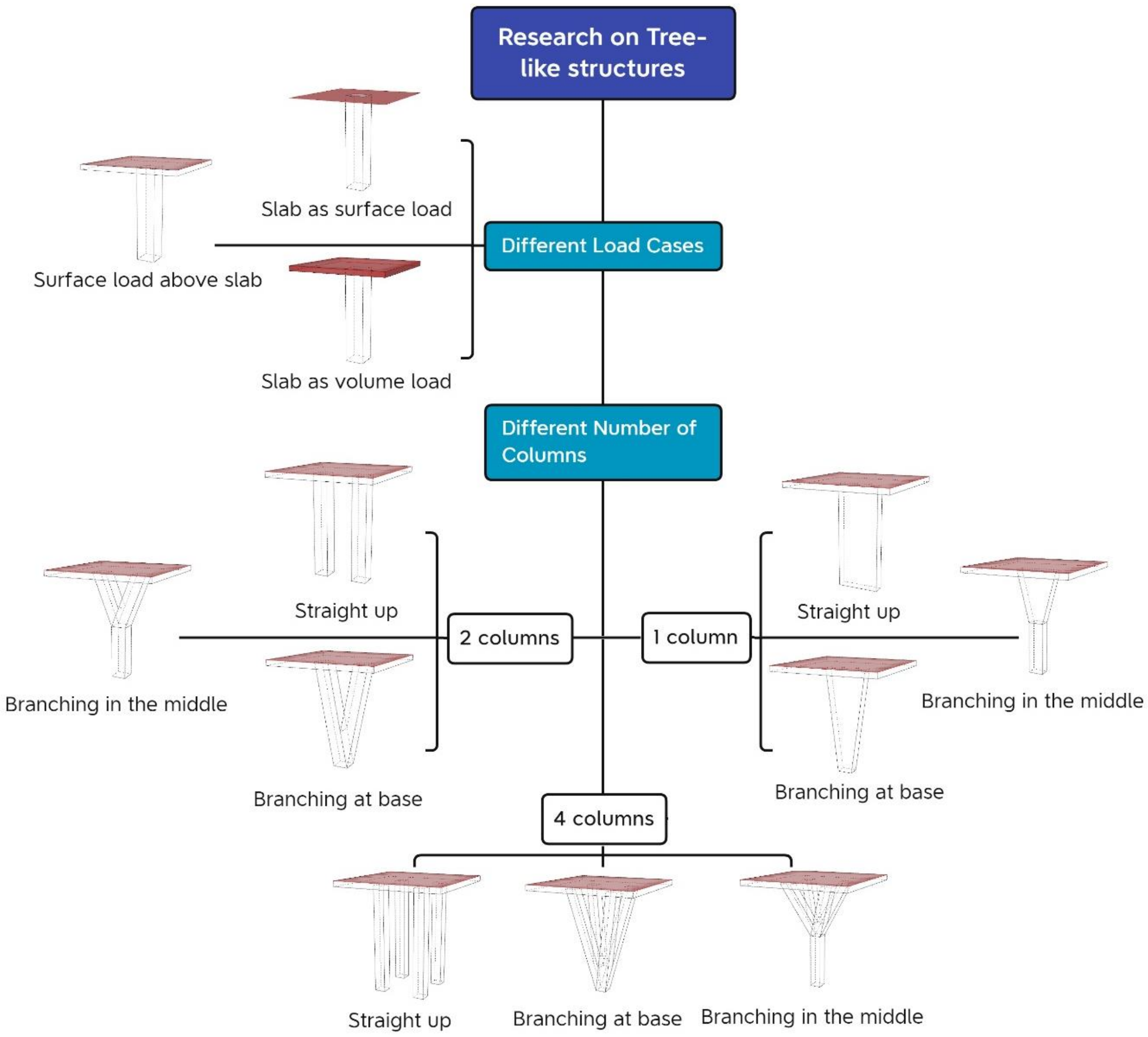


Figure 6.2. Case Studies used for exploring tree-like structures

In the first stage studying load options, a single column with a 2×2 m base is defined as the domain. Three different load variations are explored in this case (Figure 6.3). A 10×10 m surface is defined as the load in the first case. The same surface is extruded into a volume with a 0.3 m thickness and the volume is defined as a load. In the third variation, the top volume is provided as part of the domain representing a slab while a surface load is defined on top of it.

(a) (b) (c)

Figure 6.3. Given load types (a) Surface load (b) Volume load (c) Surface load on top of a slab

The results from this study are shown in Table 6.21. The first two cases result in column structures since no horizontal element is defined as part of the domain. However, the case in which a surface load is defined resulted in a more organic form. Thus, it can be concluded that providing a surface as a load result in a more detailed finer form rather than the bulky forms resulting from providing a volume load. The surface load on top of a slab signifies results more representative of an architectural element and space. Also, instead of the single columns, this case resulted in a horizontal element (slab) connected to a tree-like branching structure. The provided domain and boundary conditions allow the geometry to branch into smaller elements near the slab which gives results much similar to tree-like structures. Accordingly, including a slab as part of the domain gives more suitable results to study different load-bearing structures carrying a slab. Thus, A slab carried by different column configurations is used as the basis for domains to explore the implementation of TO for tree-like structures.

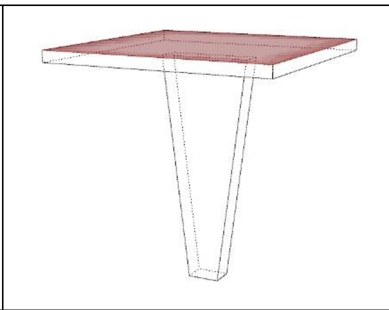
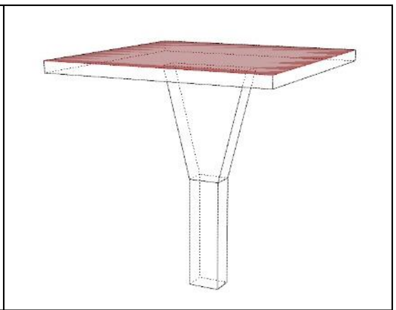
Table 6.21. Case study 1 results

Surface Load	Volume Load	Surface load on top of a slab
		
20 iterations		
		
50 iterations	37 iterations	50 iterations

After deciding on the load type to be used, different column forms are explored at this stage. A single, double, and four-column set is studied. In each case, a straight-up column is defined, a tilted version (branching at the base), and a half-straight half-tilted version (branching in the middle) are defined. The goal of providing these variations is to analyze how the TO algorithm would distribute the material to create the most efficient structure with minimizing the material consumption as much as possible.

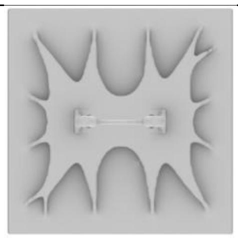
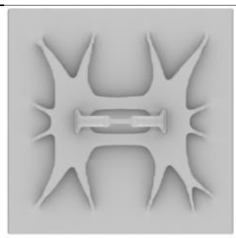
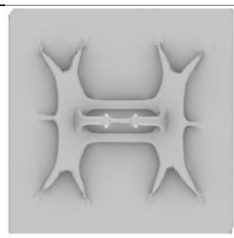
Table 6.22 Shows the three different cases of a single column and its results. In all cases, the material has been distributed to form two thin columns at almost the side edges of the provided domain. A horizontal bar connecting the two columns is introduced at varying heights in all cases. All three cases, including the straight-up column domain, have some sort of inclination in the columns with variation in the angle from one to the other.

Table 6.22. Single-column cases and results

		
		
20 iterations		
		
50 iterations		

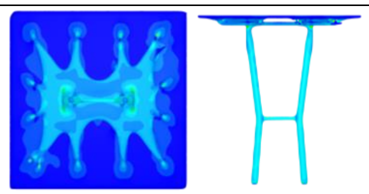
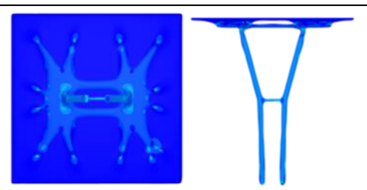
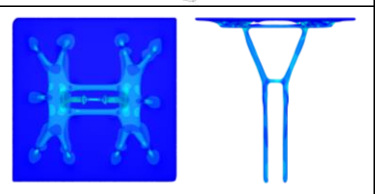
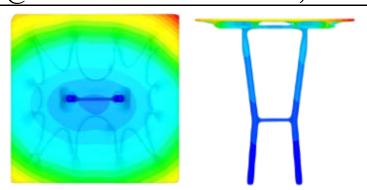
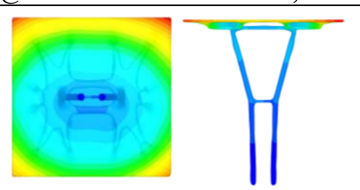
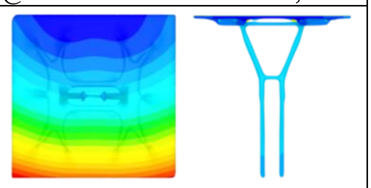
To allow further visual study of the results, the front and bottom views are shown in Table 6.23. The columns become thinner and closer together in the branching cases. The branching at the base case includes a horizontal supporting bar approximately in the middle while the first case's bar is lower and the last is higher. All cases include branching towards the top supporting the slab and allowing it to be thinner. This horizontal supporting panel that is created underneath the slab varies in complexity from one case to the other. The branching of the panel gets simpler and minimal in the last case of the branching in the middle domain while the first case of the straight up domain includes more extensive branching.

Table 6.23. Front and bottom view of the single-column results

	Straight up	Branching at the base	Branching in the middle
			
Front view			
Bottom view			

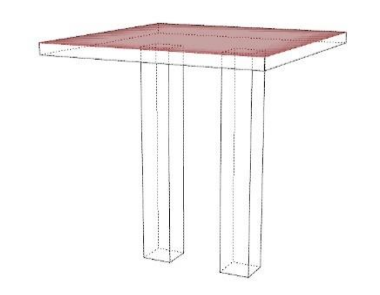
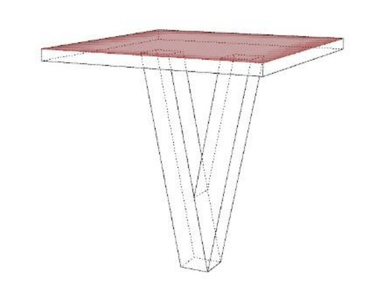
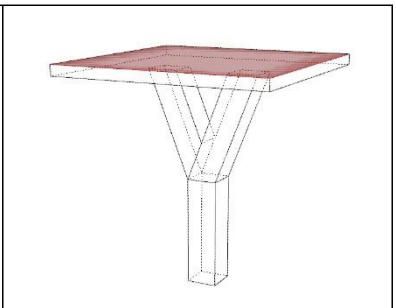
The final results are re-meshed and analyzed in Simscale to calculate displacement and stress values (Table 6.24). The non-branching case had the least stress and displacement max values. Stress values are calculated also at the base and column-slab connection and displacement at the slab edge which equaled the max displacement value in all cases.

Table 6.24. Stress and displacement studies on single-column results

			
Von Mises stress (MPa)			
	Max; 12.6	Max; 36.77	Max; 30.91
	@ Base; 7.68	@ Base; 6.6	@ Base; 7.39
	@ Column-slab connection; 7.87	@ Column-slab connection; 8.01	@ Column-slab connection; 9.41
Displacement Z (m)			
	Max; 6.18×10^{-3}	Max; 7.72×10^{-3}	Max; 18.95×10^{-3}
	@ Slab edge; 6.18×10^{-3}	@ Slab edge; 7.72×10^{-3}	@ Slab edge; 18.95×10^{-3}

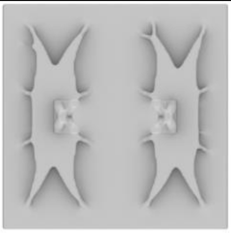
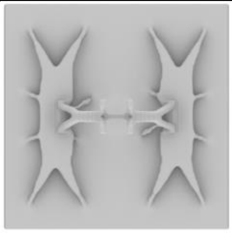
Two columns are introduced next with 3 different cases as shown in Table 6.25. As the straight-up case includes a complete separation between the two columns, its result includes only two inclined columns with no horizontal bar connecting them as in the two other cases. In all cases, the created columns using topology optimization are similar in that the columns branch at both the top and bottom into two segments while connecting and forming single columns in the center.

Table 6.25. Double-column cases and results

		
		
20 iterations		
		
50 iterations		

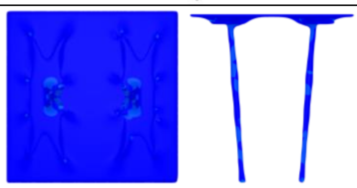
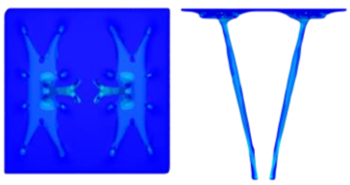
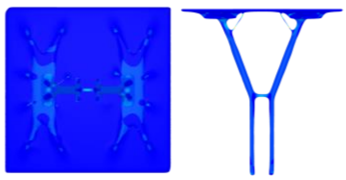
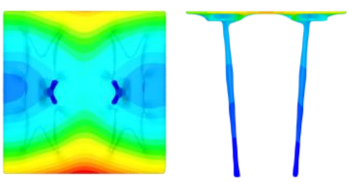
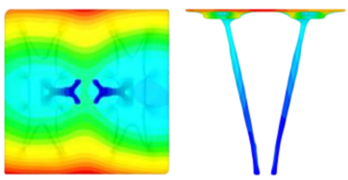
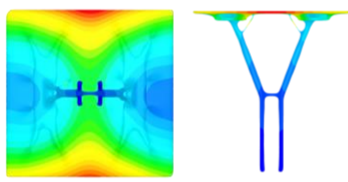
The front and bottom views are shown in Table 6.26. The straight-up case results in slightly inclined columns while the branching at the base case is more inclined following the domain's edges. The last case's columns start straight and branch into inclined parts with a supporting horizontal bar in the middle. It also includes two diagonal supporting bars expanding from the top of the columns to the slab which is the only case including such a supporting element. The branching in the middle case also includes less branching complexity in the supporting horizontal panel compared to the other case studies.

Table 6.26. Front and bottom view of the double-column results

	Straight up	Branching at the base	Branching in the middle
			
Front view			
Bottom view			

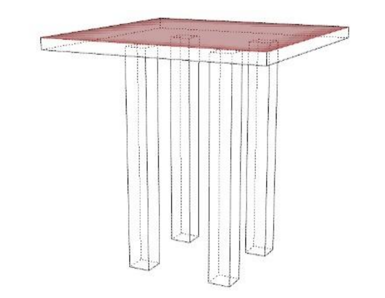
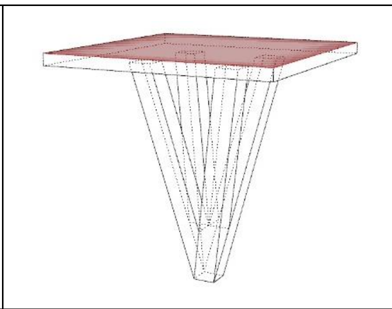
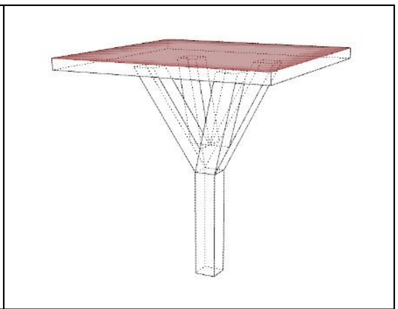
The three results are re-meshed and taken to Simscale for analysis (Table 6.27). The non-branching case has the highest stress value while the branching at the base case had the lowest. When it comes to the displacement the branching at the base case had the lowest max value while the branching in the middle case had the highest value. Stress and displacement values are calculated at other specific points.

Table 6.27. Stress and displacement studies on double-column results

			
Von Mises stress (MPa)			
	Max; 60.6	Max; 30.72	Max; 53.45
	@ Base; 7.07	@ Base; 10.18	@ Base; 7.78
	@ Column-slab connection; 5.91	@ Column-slab connection; 5.09	@ Column-slab connection; 5.69
Displacement Z (m)			
	Max; 4.96×10^{-3}	Max; 4.62×10^{-3}	Max; 5.98×10^{-3}
	@ Slab edge; 4.96×10^{-3}	@ Slab edge; 4.62×10^{-3}	@ Slab edge; 5.98×10^{-3}

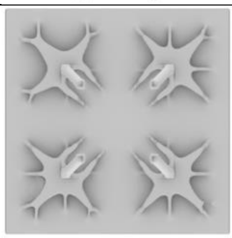
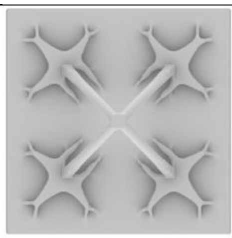
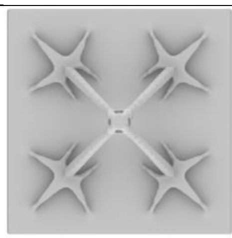
Four columns are defined in the third part of this study. The different configurations of columns and their results are shown in Table 6.28. This case had the most variation between the three results. The straight-up columns resulted in four slightly tilting columns with upper small truss-like elements diagonally aligned giving further support to the structure. The second and third cases were relatively similar with a single thick column at the bottom branching into 4 thinner columns. However, the domain branching at the center results in further complexity in the bottom column as the column first branches into 4 straight-up thinner columns to later expand into inclined branches.

Table 6.28. Four-column cases and results

		
		
20 iterations		
		
50 iterations		

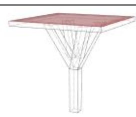
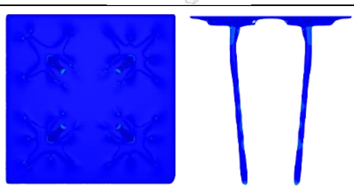
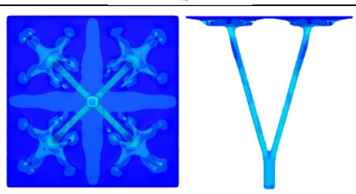
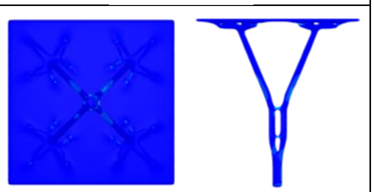
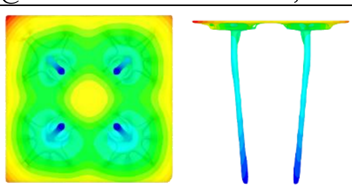
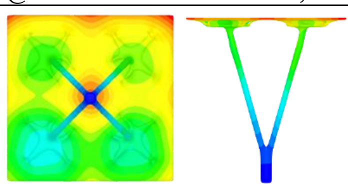
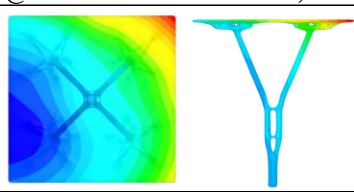
The front and bottom views are provided in Table 6.29. The straight-up case includes columns that inclines mostly outwards but incline inwards near the top. All the results include four separate columns at the top with 4 supporting structures splitting into small branches connecting to the slab. This branching is, similar to previous cases, simpler in the branching in the middle case compared to the other cases.

Table 6.29. Front and bottom view of the four-column results

	Straight up	Branching at the base	Branching in the middle
			
Front view			
Bottom view			

Same as in the last two cases, the results are analyzed in Simscale to find stress and displacement values (Table 6.30). The branching in the middle case had the highest max stress and displacement values compared to the other two. Stress and displacement are also found at specific points. The displacement at the slab edge is, same as in the previous cases, equal to the max values in all cases.

Table 6.30. Stress and displacement studies on four-column results

			
Von Mises stress (MPa)			
	Max; 54.36	Max; 13.02	Max; 86.07
	@ Base; 11.01	@ Base; 5.79	@ Base; 6.84
	@ Column-slab connection; 7.02	@ Column-slab connection; 3.8	@ Column-slab connection; 4.81
Displacement Z (m)			
	Max; 2.1×10^{-3}	Max; 1.95×10^{-3}	Max; 9.09×10^{-3}
	@ Slab edge; 2.1×10^{-3}	@ Slab edge; 1.95×10^{-3}	@ Slab edge; 9.09×10^{-3}

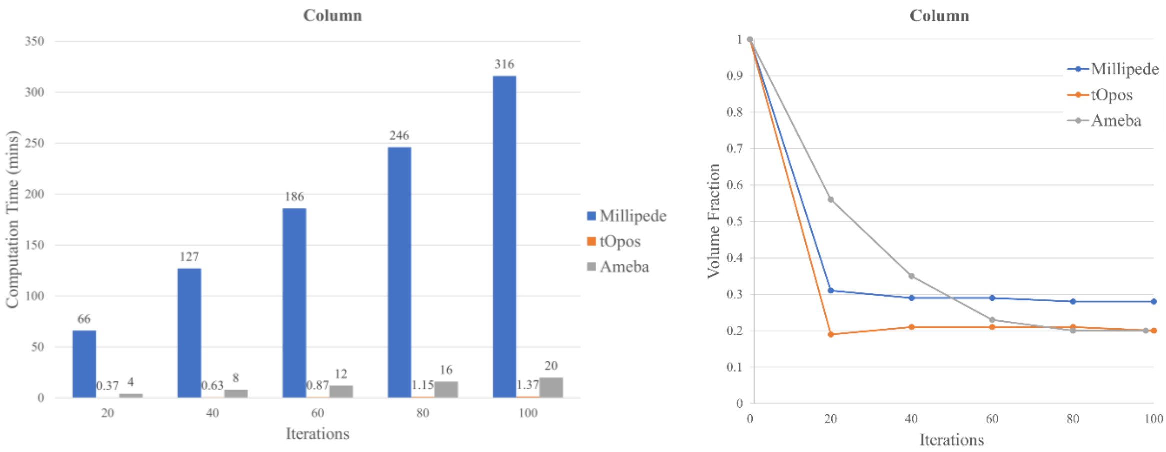
7. RESEARCH FINDINGS AND DISCUSSION

This research has two separate research findings, one being related to the TO grasshopper plugins and the other related to treelike structures.

7.1. Research Findings for TO Tools

Looking at the first part of the study, the three grasshopper plugins (Millipede, tOpos and Ameba) performed differently from each other. Based on multiple comparison parameters, the plugins varied in their performance.

Computation time was a huge distinction factor between the plugins (Figure 7.1.a & Figure 7.2). In all cases, Millipede took the longest, mostly spending hours to compute while tOpos was the fastest among the plugins as it included a GPU computation option.



(a) (b)

Figure 7.1. Column Case study; (a) Optimization time for each trial (b) Volume fraction in relation to the number of iterations

Looking at the volume fraction, Ameba always satisfied the required fraction (Figure 7.1.b & Figure 7.3). tOpos reached the volume fraction in three cases and was relatively close to it in the other two while Millipede failed to reach the required volume fraction in all cases. Also, it is noticed that the volume fraction drops to the target or near to it in tOpos faster than Ameba. The volume fraction is almost reached between the 20^{th} and 40^{th} iterations in tOpos while it takes Ameba between 60 to 80 iterations to approach the target volume fraction.

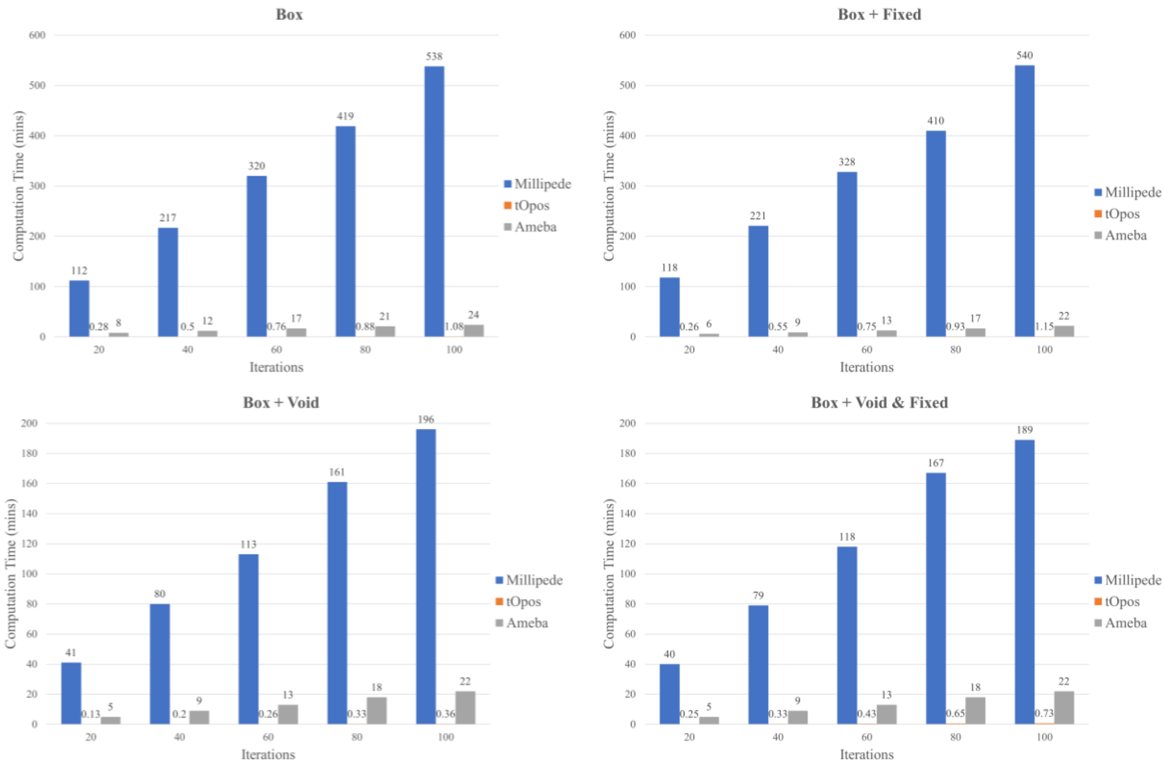


Figure 7.2. Box case studies; Optimization time for each trial

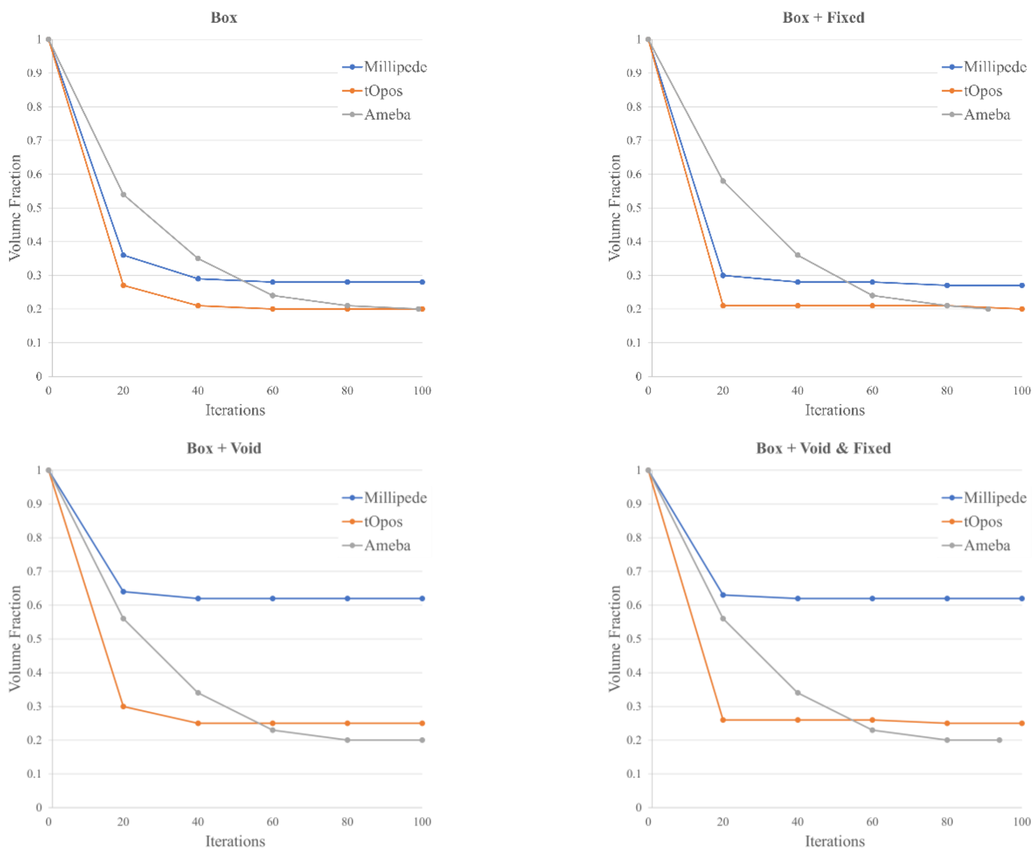


Figure 7.3. Box case studies; Volume fraction in relation to the number of iterations

Taking a look at the structural performance, the maximum stress and displacement values varied from one case study to another (Figure 7.4). Ameba generally had the highest maximum displacement and stress values. Low stress values in Millipede can be associated with its results being mostly solid with no thin elements. tOpos mostly had max values in between the other two plugins. With that being said, the values in almost all cases are extremely low which reflects generally good structural performance.

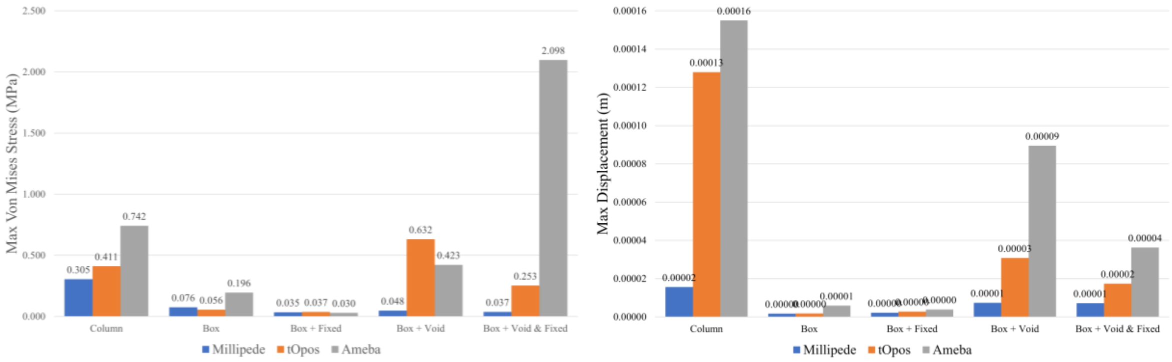


Figure 7.4. Comparison between values of max displacement and stress of the final results for the 3 plugins

In practice, topology optimization results are generally further developed and optimized. However, this research focused on studying and comparing grasshopper plugins for topology optimization and their ability to produce conceptual forms derived from TO. Millipede seemed to be the weakest plugin. It took long hours to conduct each trial and in all cases was not able to achieve the target volume fraction despite the long computation time. Ameba took reasonable time compared to millipede and did reach target volume fractions for all case studies. However, for a commercial plugin, one would expect a finer smoother raw result. Ameba includes a smoothing tool and results can be edited and smoothed using other plugins as done in this research using Dendro. Yet simple smoothing is not enough, and the results need to be redrawn or at least remeshed to produce a logical result. tOpos was the fastest to generate results using GPU. It is worth mentioning however that it is not compatible with all graphic cards. Moreover, when using GPU mode, the plugin sometimes stops during iterations without finishing and reaching the required result and other times stops working after one run. The plugin provides a reset component which solves the issue most of the time. It also includes a CPU mode, but it does take longer time than GPU mode. In case GPU works, tOpos seems to be the most suitable plugin as multiple different options and parameters can be tested in a short amount of time.

7.2. Research Findings for Tree-like structures

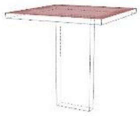
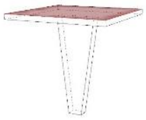
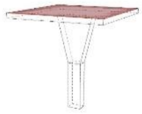
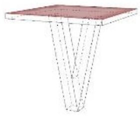
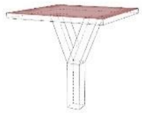
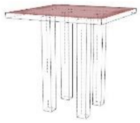
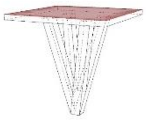
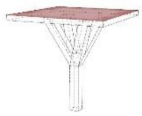
When it comes to treelike structures, topology optimization presents the potential to produce various forms. The user inputs including load types and domains greatly affect the obtained form. Changing the number of columns and their connections and placement impacts the final outcome. It was noticed that the larger the number of columns, the more complex the result gets. Also, straight-up columns result in relatively simpler forms while branching domains result in much more complex yet more analogous to tree-like structures.

When results are analyzed visually, the branching cases result in structures more resembling tree-like ones (Table 7.1). That is fairly expected as the non-branching straight-up cases are, as domains, the furthest in resemblance of tree-like structures. They are, however, included as they can provide valuable insight into how a branching domain could differ from a non-branching one. All the cases of the one-column domain resulted in two column-like vertical elements supporting the slab. These two columns are connected by a single horizontal support. The differences were in the location of the horizontal support, the angle of the columns, and the thickness of the elements. The horizontal support was placed towards the bottom in the straight-up case, almost in the center in the branching at the base case and in the upper part in the branching at the middle case. Despite having inclined edges in the branching at the base domain, the lower part of the columns is almost straight in a similar fashion to the straight lower column part in the branching in the middle case. The straight-up case worked the opposite way in which even the lower part of the columns is inclined.

The two-column domain also resulted in a two-column structure supporting the slab. It differs in that the columns branch both at the base and top into two smaller elements. Also, the branching in the middle case was the only one including a connecting horizontal support. The branching at the base case differs from the straight up one almost only in the inclination angle of the columns.

The four-column domains resulted in forms noticeably different from the other cases. The straight-up domain resulted in four slightly inclined columns with a small truss-like element at the top supporting the connection between the slab and column. The other two cases resulted in a single thicker column eventually branching into four. The branching in the middle case differs in that the single column branches first into four straight vertical columns with connecting horizontal supports between them. These four vertical columns branch later into the four inclined ones. The branching four-column cases are as logically expected the most resemblant of tree structures as they are the only ones starting at the base with one column that branches as it gets to the top. The other cases, however, have some sort of resemblance in one way or another.

Table 7.1. Final results of the tree-like structures; perspectives

	Straight up	Branching at the base	Branching in the middle
1 column			
			
2 columns			
			
4 columns			
			

When front views are analyzed and compared, multiple key marks are noted (Table 7.2). All the one and two-column domains result in two main columns supporting the structure. All three one-column cases and the branching in the middle two-column case resulted in highly similar results that contain two columns with a supporting horizontal bar connecting them. The difference is mostly in the location of this horizontal bar and the inclination of the columns. That is along with an extra diagonal support near the top in the branching in the middle two-column case. From the front view, the straight-up two and four-column cases look similar in that it results in straight-up separate columns. It however differs in that the two-column case slightly inclines outwards from base to top while the four-column case starts inclining outwards from the base to then incline inwards toward the top. In a comparable manner, the branching at the base two and four-column cases appear similar with the four-column case starting at the base as one column before branching while the two-column case results in two completely separate columns. The four-column branching in the middle case appears to be the most complex and distinct from all the other cases.

It is equally valuable to compare the bottom views of the results to get an insight into how the columns branch and connect to the slab in each case (Table 7.3). The area percentage of the horizontal supporting panels compared to the slab area is calculated. The number of these panels and the number of branches connecting them to the slab is calculated. The number of supporting horizontal panels branching from the columns and supporting the slab is equal to the number of columns in each domain. In the one-column cases, for instance, although the results contained two main columns, the supporting horizontal panel is only one. With that being said, the one-column branching cases include a hole in the middle but stay relatively intact as one element. In all cases, the straight-up ones include more complex branching of the horizontal panel into small branches that connect to the solid slabs above it. The branching at the base cases becomes less complex compared to the straight up ones. The simplest branching happens, however, in the branching in the middle cases.

Table 7.2. Final results of the tree-like structures; front views

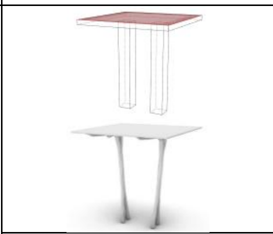
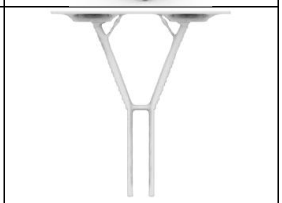
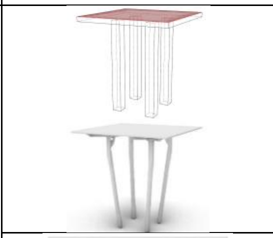
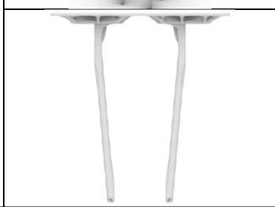
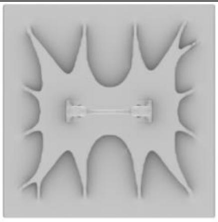
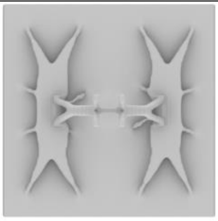
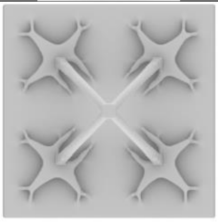
		Straight up	Branching at the base	Branching in the middle
One-column Domain				
				
Number of columns		2	2	2
Inclination	From base to a % of the height	1/3: inwards inclination	1/2: Straight	2/3: Straight
	Rest of column	outwards inclination	outwards inclination	outwards inclination
Connecting horizontal bar		Yes	Yes	Yes
Two-column Domain				
				
Number of columns		2	2	2
Inclination	From base to a % of the height	Slight outwards inclination	outwards inclination	1/2: Straight
	Rest of column			outwards inclination
Connecting horizontal bar		No	No	Yes
Four-column Domain				
				
Number of columns		4	1 to 4	1 to 4
Inclination	From base to a % of the height	3/4: outwards inclination	1/4: Straight	1/2: Straight
	Rest of column	inwards inclination	outwards inclination	outwards inclination
Connecting horizontal bar		No	No	Yes

Table 7.3. Final results of the tree-like structures; bottom views

	Straight up	Branching at the base	Branching in the middle
One-column Domain			
			
Number of panels	1	1	1
Complexity	Most complex	Average complexity	Least complex
Area percentage to slab	32%	23%	18%
Number of branches	14	13	10
Two-column Domain			
			
Number of panels	2	2	2
Complexity	Most complex	Average complexity	Least complex
Area percentage to slab	25%	22%	19%
Number of branches	18	16	14
Four-column Domain			
			
Number of panels	4	4	4
Complexity	Most complex	Average complexity	Least complex
Area percentage to slab	19%	16%	11%
Number of branches	40	32	20

When it comes to structural performance, results varied from one case study to another (Figure 7.5). The one-column straight up case had the lowest max stress value while the four-column branching in the middle case had the highest stress value. The one-column branching in the middle case, however, had the highest displacement while the 4-column straight up and branching at the base had almost the same lowest maximum displacement value.

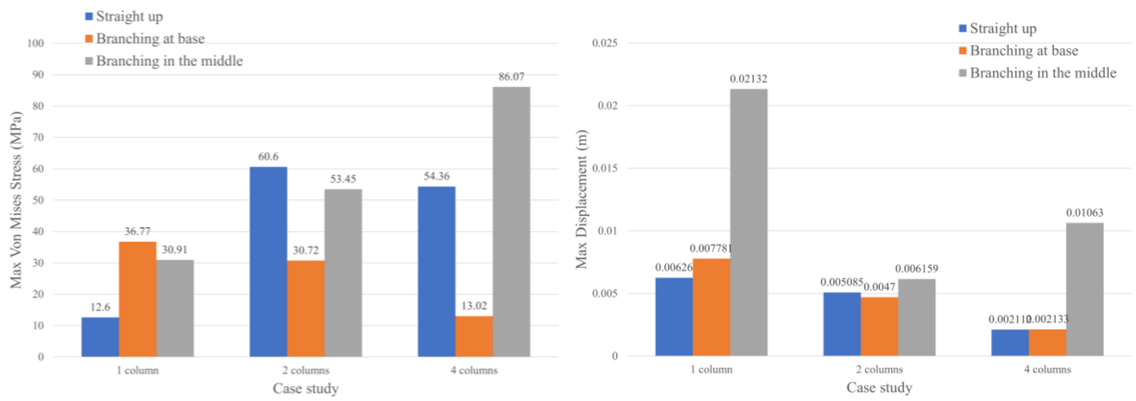
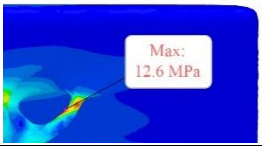
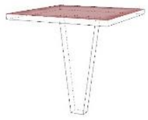
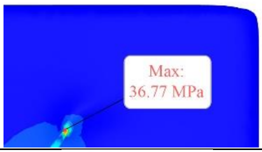
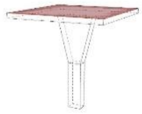
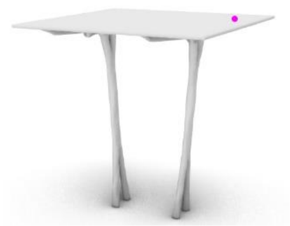
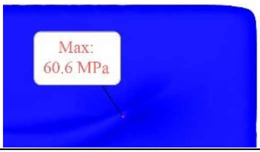
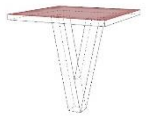
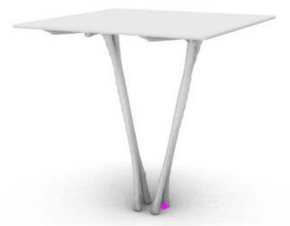
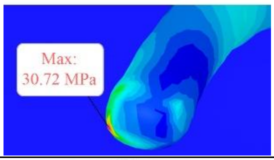
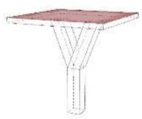
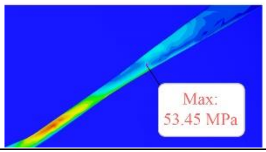
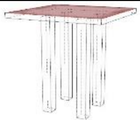
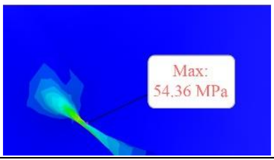
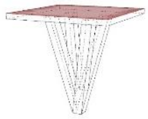
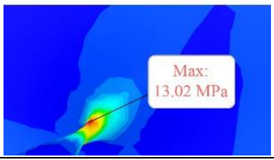
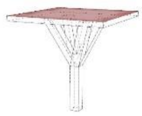
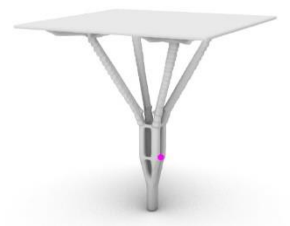
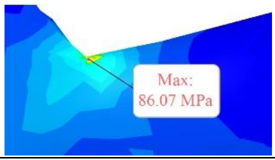


Figure 7.5. Comparison between Max stress and displacement values in all case studies.

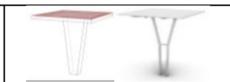
In an effort to find the reasoning behind the high stress values in some cases, the location of the maximum stress in each case is identified (Table 7.4). Except for one case in which the max stress is located at the base of the column and another located in the horizontal supporting element, the results included high stress values mostly in small branches connecting to or supporting the slab. Such an issue can be resolved in the post-processing of topology optimized results in which high stress thin areas can be thickened slightly to provide more stability to the structure while keeping it relatively similar to the results obtained from TO.

Table 7.4. Maximum stress location in each case study

	Straight up	Branching at the base	Branching in the middle
1 column	  	  	  
2 columns	  	  	  
4 columns	  	  	  

In an effort to further analyze the results, the stress values are calculated at the base of the column and in the column slab connection. Displacement at the slab edges is inspected also which in all cases equaled to the max displacement value. Looking at the stress and displacement values of the one-column cases, the straight-up case seemed to perform better than the other two cases (Table 7.5). Except for the max stress, which can be resolved by thickening the small branches at the top, the branching at the base case had relatively similar values.

Table 7.5. Stress and displacement comparison between the one-column cases

One-column cases				
Von Mises stress (MPa)	Max	12.6	36.77	30.91
	@ Base	7.68	6.6	7.39
	@ Column-slab connection	7.87	8.01	9.41
Displacement Z (m) @ Slab edge (Max)		6.18×10^{-3}	7.72×10^{-3}	18.95×10^{-3}

When it comes to the two-column cases, the branching at the base generally resulted in the best values with a slightly higher stress at the base (Table 7.6). This could be associated with the fact that the two columns are relatively highly inclined with no supporting horizontal bar connecting the two columns.

Table 7.6. Stress and displacement comparison between the two-column cases

Two-column cases				
Von Mises stress (MPa)	Max	60.6	30.72	53.45
	@ Base	7.07	10.18	7.78
	@ Column-slab connection	5.91	5.09	5.69
Displacement Z (m) @ Slab edge (Max)		4.96×10^{-3}	4.62×10^{-3}	5.98×10^{-3}

In the four-column cases, it was clear that the branching at the base case was the best out of the three (Table 7.7). From this comparison, it can be concluded that having a thick column branching into four would be structurally better than having four separate thinner columns as the branching cases had less stress at the base and column slab connection compared to the straight-up case.

Table 7.7. Stress and displacement comparison between the four-column cases

Four-column cases				
Von Mises stress (MPa)	Max	54.36	13.02	86.07
	@ Base	11.01	5.79	6.84
	@ Column-slab connection	7.02	3.8	4.81
Displacement Z (m) @ Slab edge (Max)		2.1×10^{-3}	1.95×10^{-3}	9.09×10^{-3}

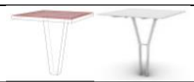
When the straight-up cases are compared, it is hard to specify which was better from a structural point of view (Table 7.8). The two-column case was generally better in stress while the four-column one had the least displacement.

Table 7.8. Stress and displacement comparison between the straight-up cases

Straight up cases				
Von Mises stress (MPa)	Max	12.6	60.6	54.36
	@ Base	7.68	7.07	11.01
	@ Column-slab connection	7.87	5.91	7.02
Displacement Z (m) @ Slab edge (Max)		6.18×10^{-3}	4.96×10^{-3}	2.1×10^{-3}

Same when compared with the four-column cases, the four-column branching at the base seems to be the best when compared with the other branching at the base cases (Table 7.9).

Table 7.9. Stress and displacement comparison between the branching at the base cases

Branching at the base cases				
Von Mises stress (MPa)	Max	36.77	30.72	13.02
	@ Base	6.6	10.18	5.79
	@ Column-slab connection	8.01	5.09	3.8
Displacement Z (m) @ Slab edge (Max)		7.72×10^{-3}	4.62×10^{-3}	1.95×10^{-3}

Same as the straight-up cases, it is hard to clearly select the best case between the three (Table 7.10). The complete opposite shows here as the four-column case was generally better in stress while the two-column one had the least displacement.

Table 7.10. Stress and displacement comparison between the branching in the middle cases

Branching in the middle cases				
Von Mises stress (MPa)	Max	30.91	53.45	86.07
	@ Base	7.39	7.78	6.84
	@ Column-slab connection	9.41	5.69	4.81
Displacement Z (m) @ Slab edge (Max)		18.95×10^{-3}	5.98×10^{-3}	9.09×10^{-3}

In general, the displacement values were higher in the branching in the middle cases which can be linked to the horizontal supporting panel being less complex compared to the other cases. As it branches less and further away from the edges, the displacement at the edges increases due to the shortage of support. Also, stress values at the base and column-slab connection represent a more valid indication of the structural stability as the maximum stress values are in most cases due to extremely thin areas which can be edited in the post-processing. Additionally, in case writing a TO code is feasible, some constraints such as a minimum element cross-section can resolve such an issue.

Final comparison between the nine cases shows that the four-column branching at the base case resulted in the best form both on a structural and visual aspect as it was the most resemblant of tree-like structures while performing the best structurally. This proves the potential of tree-like structures that is expressed in multiple research projects and architects' work. Additionally, it shows the potential topology optimization has in producing forms that are both structurally stable and architecturally innovative.

8. CONCLUSION

Architectural forms have developed and the different methods of obtaining these forms have changed over the years, from basic geometrical forms to more complex and organic structures. The advancement of computer-aided design and the availability and accessibility of digital tools allowed for further exploration of what an architectural form is and how it could be obtained. Topology optimization is one process that works on allocating the material within a design domain based on the specified design objective, parameters, and constraints. Thus, it is able to produce novel forms that could vary depending on these provided parameters and constraints. Initially, most of the results obtained by TO were seen as not constructable. With that being said, the advancement in fabrication methods especially additive manufacturing and robotic fabrication has made it more feasible to construct TO results. Consequently, more studies are exploring the potential and ways of implementing TO in architecture.

Although results can be further developed using other methods, TO is appropriate for implementation in conceptual design to guide the design and help produce novel forms. Different factors affect the results that could be obtained including the domain itself (shape, size), boundary conditions (loads, supports), void and solid fixed regions along with plugins' parameters (volume fraction, number of iterations, resolution, iso value, etc.). The role of the architect revolves around providing these parameters, trying different combinations, and finding which results in the most suitable form for the specific design goals and needs. This research explored some of these parameters and examined their effect on TO results.

The first stage of the research investigated the grasshopper plugins for TO (Millipede, tOpos, and Ameba). Each plugin's features, advantages and disadvantages were explored and discussed based on both literature and various conducted case studies. All plugins produced feasible results that can be further developed to reach a novel final design. Furthermore, depending on the specific design problem, each plugin has advantages that may make it more preferred than the others.

Thereafter, this research used tree-like structures as a case for exploring TO and the effect of making changes to the design domain on the final forms. The nine case studies were evaluated and compared from both geometrical (visual) and structural aspects.

A final comparison between all nine cases can be seen in Figure 8.1. It can be noticed that when the TO algorithm is given the freedom to allocate material within a one-column domain, it always favored distributing the material in a way that results in two main columns supporting the structure instead of having one simple column in the middle. Hence, the one and two-column cases resulted in fairly similar forms. This can be logically associated with the fact that having two columns at the edges of the domain would minimize the span of the cantilever in the slab which would make the structure more stable.

The complexity of the overall structure, including the number of columns and their branching increases as the domain's column number increases. This is logically expected as increasing the complexity of the design domain is reflected in an increasing complexity in the resulted forms. The horizontal panel connecting the main columns to the slab is the most complex in straight up cases while becoming the least complex in the branching in the middle cases. This could be associated with the straight up cases having simpler columns than the branching ones. Thus, the algorithm produced a more complex and larger supporting panel to further help in carrying the slab. In contrast, the complexity of the columns in the branching cases eliminates the need for a large horizontal supporting panel.

When it comes to structural performance, of the nine cases, the four-column branching at the base case seemed to produce the best result. This can further confirm that having a branching structure would be better than a non-branching one. Furthermore, it seemed that having a single column at the base that branches later would be better than having two separate columns at the base even if they connect and branch towards the top. That confirms already found concepts in literature suggesting that branching the column results in a better structure.

Also, the result is better than the branching at the middle case which indicates that branching at a lower point within the column's height could be better in some cases. With that being said, the results of the branching domains do not actually branch at the base or middle. The difference in the domains means that the structure is allowed to branch at a lower point (perhaps the base) in the first case while its branching is restricted to a higher point within the branching in the middle domains. The TO results, however, did not include any forms branching exactly at the base or middle.

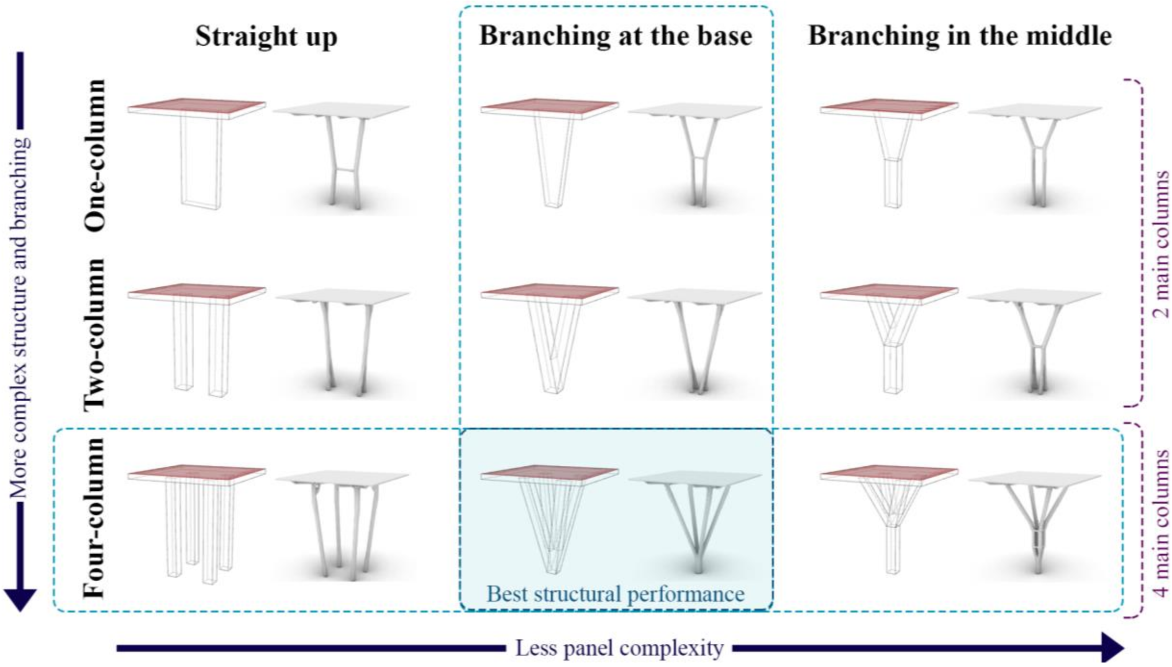


Figure 8.1. Comparison between the nine results

The main conclusion of the study is that it serves as a proof of concept that tree-like structures have the potential in producing more structurally sound structures as having a column branching into smaller ones generated better final forms. Moreover, the results of this study show the effect of the architect's decisions on the ability to produce better results even when an algorithm (in this case TO process) is used. It also proves that TO is a powerful tool that can be used more in the architectural field to produce structurally sound, more economical (using less material), innovative architectural forms.

Future possible research

Further studies can include an exploration of other parameters and design domains. As this research took tree-like structures as a case study to explore the effect that varying domain configurations has on the final result, other structure types can be explored for further understanding of how TO works, the effects of different domains and other user inputs and the potential and possibilities TO has in different structure types. The boundary conditions including different load and support options can also be further explored. Keeping all inputs fixed and changing one parameter at a time to explore its impact on the resulting form can offer huge insight for architects to guide them on how to choose different design parameters depending on their specific design problem.

Tree-like structures can be further explored using TO as a design method. Further studies can include an examination of more design parameters and constraints. This includes including a fixed solid area at the base representing a single column and analyzing how the TO algorithm would distribute the material differently when the base is set up to a single thick column. The observation regarding where the branching starts and its effect on the results can also be further investigated. Different levels of branching can be tested and how this affects the visual and structural aspects of the form can be thoroughly studied. Additionally, the proportions of the design domains could be changed to explore how the results would vary. Having column bases that expand equally on the x and y axis and taking up a larger area would potentially produce extremely different results that could be compared to the ones obtained in this study.

Moreover, studies on the fabrication of treelike structures obtained from TO can be conducted. Different fabrication constraints can be included in the design process to investigate their benefit and influence on the resulting form. The fabrication process itself, its challenges, and how to tackle these issues can also be a valuable study area. Furthermore, Different materials can be studied and the impact of using different material properties on the design process and fabrication of TO results can be explored.

Further studies can also include more complex case studies and explore other details and options provided in the plugins. Also, a comparison between the grasshopper plugins and other tools can be conducted to study their feasibility. More exploration of the available TO tools, in general, can be very beneficial in helping architects choose the most suitable tool for their needs.

REFERENCES

Abdelsabour, I. (2019). Investigating bio-morphism approach to enhance structure's creativity and efficiency. *Journal of Engineering and Applied Science*, 66(5), 491-513.

Agkathidis, A., and Brown, A. (2013). Tree-structure canopy: A case study in design and fabrication of complex steel structures using digital tools. *International Journal of Architectural Computing*, 11, 87-104. doi: 10.1260/1478-0771.11.1.87

Arslan Selçuk, S., Gülle, N. B., and Mutlu Avinç, G. (2022). Tree-like structures in architecture: Revisiting frei otto’s branching columns through parametric tools. *SAGE Open*, 12(3). doi: 10.1177/21582440221119479

Avdić, I. (2019). *Bio-inspired Approach to Early-Stage Structural Form Finding*, Master's thesis, Delft University of Technology, Netherlands, 3-18, 44.

Bao, D. W., Yan, X., Snooks, R., and Xie, Y. (2020). Bioinspired generative architectural design form-finding and advanced robotic fabrication based on structural performance. In *Architectural intelligence, selected papers from the 1st international conference on computational design and robotic fabrication (CDRF 2019)*, 147-170. doi: 10.1007/978-981-15-6568-7_10

Bartels, N., and Houben, T. (2016). *Structural Optimization For 3d Printing*, Master's thesis, Eindhoven University of Technology, Netherlands, 1-23.

Beghini, L., Carrion, J., Beghini, A., Mazurek, A., and Baker, W. (2014). Structural optimization using graphic statics. *Structural and Multidisciplinary Optimization*, 49, 351-366. doi: 10.1007/s00158-013-1002-x

Beghini, L. L., Beghini, A., Katz, N., Baker, W. F., and Paulino, G. H. (2014). Connecting architecture and engineering through structural topology optimization. *Engineering Structures*, 59, 716-726. doi: 10.1016/j.engstruct.2013.10.032

Bendsøe, M. P., and Sigmund, O. (2003). *Topology optimization: theory, methods and applications* (2nd Edition). Germany: Springer, 1-14.

Bhatia, I. S. (2019). *Shaping Transparent Sand in Sand*, Master's thesis, Delft University of Technology, Netherlands, 10-17.

Bialkowski, S. (2016, 24-26 Aug). *Structural optimisation methods as a new toolset for architects*. 34th eCAADe Conference - Complexity & Simplicity, University of Oulu, Finland.

Bialkowski, S. (2017, 20-22 Sep). *tOpos GPGPU accelerated structural optimisation utility for architects*. 35th eCAADe Conference, Sapienza University of Rome, Rome, Italy.

Bialkowski, S. (2018, 19-21 Sep). *Topology optimisation influence on architectural design process - enhancing form finding routine by tOpos toolset utilisation*. 36th eCAADe Conference, Lodz University of Technology, Lodz, Poland.

Burry, J., and Burry, M. (2010). *The new mathematics of architecture* (1st Edition). London: Thames & Hudson, 130-133.

Cuellar Loyola, N. A., Pereira, A., and Menezes, I. (2015, 22-25 Nov). *Robust topology optimization under uncertain loads: A spectral stochastic approach*. CILAMCE 2015, Rio de Janeiro, Brazil.

Damen, W. (2019). *Topologically Optimised Cast Glass Grid Shell Nodes*, Master's thesis, Delft University of Technology, Netherlands, 9-13.

Dapogny, C., Faure, A., Michailidis, G., Allaire, G., Couvelas, A., and Estevez, R. (2017). Geometric constraints for shape and topology optimization in architectural design. *Computational Mechanics*, 59(6), 933–965. doi: 10.1007/s00466-017-1383-6

Deaton, J. D., and Grandhi, R. V. (2014). A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization*, 49(1), 1-38. doi: 10.1007/s00158-013-0956-z

Dijk, R. V. (2020). *Topology Optimization as Architectural Form Finding*, Master's thesis, Delft University of Technology, Netherlands, 6-13.

Dixit, S., Stefańska, A., and Musiuk, A. (2021). Architectural form finding in arboreal supporting structure optimisation. *Ain Shams Engineering Journal*, 12(2), 2321-2329.

Du, W., Zhao, Y., Wang, Y., Wang, H., and Yang, M. (2022). Novel machine learning approach for shape-finding design of tree-like structures. *Computers & Structures*, 261-262. doi: 10.1016/j.compstruc.2021.106731

Eren, O., ve Sezer, H. K. (2019, 20-22 Aralık). *Üretken tasarim ve topoloji optimizasyonu yaklaşimlariyla ürün tasarimi*. Uluslararasi Bilim, Teknoloji Ve Sosyal Bilimlerde Güncel Gelişmeler Sempozyumu, Ankara, Türkiye.

Feringa, J., and Søndergaard, A. (2012, 18-21 Oct). *An integral approach to structural optimization and fabrication*. Proceedings of the 32nd Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA), San Francisco, United States.

Frattari, L., Dagg, J., and Leoni, G. (2013). Conceptual design of a pedestrian bridge by means of topology optimization. In *Structures and Architecture*, 229-236. doi: 10.1201/b15267-30

Freitas, J., and Leitão, A. (2019, 15-18 Apr). *Back to reality - dendritic structures using current construction techniques*. CAADRIA 2019: Intelligent & Informed, Victoria University of Wellington, New Zealand.

Galjaard, S., Hofman, S., Perry, N., and Ren, S. (2015, 17-20 Aug). *Optimizing structural building elements in metal by using additive manufacturing*. International Association for Shell and Spatial Structures (IASS) Symposium, Amsterdam, The Netherlands.

Gaudillière, N., Duballet, R., Bouyssou, C., Mallet, A., Roux, P., Zakeri, M., and Dirrenberger, J. (2019). Chapter 3 - Building applications using lost formworks obtained through large-scale additive manufacturing of ultra-high-performance concrete. In J. G. Sanjayan, A. Nazari, & B. Nematollahi (Eds.), *3D Concrete Printing Technology*, 37-58. doi: 10.1016/B978-0-12-815481-6.00003-8

Gebisa, A., and Lemu, H. (2017, 29 Nov-1Dec). *A case study on topology optimized design for additive manufacturing*. First Conference of Computational Methods in Offshore Technology (COTech2017), Stavanger, Norway.

Guerguis, M., and Principe, S. (2020, 30 Aug-2 Sep). *Biomimetic generative morphologies for 3D-printing*. Advances in Architectural Geometry (AAG) International Conference, Paris, France.

Hassani, B., Tavakkoli, S. M., and Ghasemnejad, H. (2013). Simultaneous shape and topology optimization of shell structures. *Structural and Multidisciplinary Optimization*, 48(1), 221-233. doi: 10.1007/s00158-013-0894-9

Hemmerling, M., and Nether, U. (2014, 12-14 Nov). *Generico: A case study on performance-based design*. XVIII Conference of the Iberoamerican Society of Digital Graphics - SIGraDi: Design in Freedom, Uruguay.

Januszkiewicz, K., and Banachowicz, M. (2017). Nonlinear shaping architecture designed with using evolutionary structural optimization tools. *IOP Conference Series: Materials Science and Engineering*, 245, 082042. doi: 10.1088/1757-899X/245/8/082042

Jipa, A., Bernhard, M., Meibodi, M., and Dillenburger, B. (2016, 3-4 Nov). *3D-printed stay-in-place formwork for topologically optimized concrete slabs*. TxA Emerging Design + Technology, San Antonio, Texas, USA.

Joye, Y. (2007). Architectural lessons from environmental psychology: The case of biophilic architecture. *Review of General Psychology*, 11(4), 305–328. doi: 10.1037/1089-2680.11.4.305

Kingman, J. J., Tsavdaridis, K. D., and Toropov, V. V. (2015). Applications of topolgy optimisation in structural engineering: High-rise buildings & steel components. *Jordan Journal of Civil Engineering*, 9(3), 335-357. doi: 10.14525/jjce.9.3.3076

Kökcü, A. (2020). *Bir zamanlar geometri* (1. Basım). Ankara: Nobel Akademik Yayıncılık, 125-135.

Lauriola, D. (2017). *Combining Efficiency and Aesthetics: Through The Integration Of Structural Topology Optimization in Architecture*, Master's thesis, Technical University of Cartagena, Spain, 1-39.

Lin, K., He, Y., Yang, Y., and Xiong, L. (2021). From topology optimization to complex digital architecture: A new methodology for architectural morphology generation. *Advances in Civil Engineering*, 2021. doi: 10.1155/2021/6839627

Lundgren, J., and Palmqvist, C. (2012). *Structural Form Optimisation, Methods of numerical optimisation and applications on civil engineering structures*, Master's thesis, Chalmers University of Technology, Sweden, 1-12.

Magan, C. (2016). *Topology Optimization of a Concrete Floor Slab Guided by Vacuumatic Formwork Constraints*, Master's thesis, Delft University of Technology, Netherlands, 1-18.

Martens, P. (2018). *Optimising 3D Printed Concrete Structures*, Master's thesis, Delft University of Technology, Netherlands, 1-4, 19-34.

Martins, J. R. R. A., and Ning, A. (2021). *Engineering design optimization* (1st Edition). Cambridge: Cambridge University Press, 1-5.

Meibodi, M. A., Bernhard, M., Jipa, A., Dillenburger, B., Menges, A., Sheil, B. O. B., Glynn, R., and Skavara, M. (2017). The smart takes from the strong, 3d printing stay-in-place formwork for concrete slab construction. In *Fabricate 2017*, 210-217. doi: 10.2307/j.ctt1n7qkg7.33

Melcher, G. (2020). *Topology Optimized Hemispherical Shell under Asymmetric Loads*, Master's thesis, Massachusetts Institute of Technology, United States, 8-11.

Mizobuti, V., and Vieira Junior, L. C. M. (2020). Bioinspired architectural design based on structural topology optimization. *Frontiers of Architectural Research*, 9(2), 264-276. doi: 10.1016/j.foar.2019.12.002

Motro, R. (2011). *Fifty years of progress for shell and spatial structures* (1st Edition). Madrid: International Association for Shell and Spatial Structures, 10.

Naboni, R., Kunic, A., and Breseghello, L. (2020). Computational design, engineering and manufacturing of a material-efficient 3D printed lattice structure. *International Journal of Architectural Computing*, 18(4), 404-423. doi: 10.1177/1478077120947990

Naboni, R., and Paoletti, I. (2018). Architectural morphogenesis through topology optimization. In D. D'Uva (Ed.), *Handbook of research on form and morphogenesis in modern architectural contexts*, 69-92. doi: 10.4018/978-1-5225-3993-3.ch004

Naous, D. (2020). *Topologically Optimised Cast Glass Shell: Topological optimisation and new fabrication methods for compressive free-form glass structures*, Master's thesis, Delft University of Technology, Netherlands, 12-17, 46-65.

Peng, X. (2016). Structural topology optimization method for morphogenesis of dendriforms. *Open Journal of Civil Engineering*, 6(4), 526-536. doi: 10.4236/ojce.2016.64045

Piccioni, V. (2019). *The AM Envelope*, Master's thesis, Delft University of Technology, Netherlands, 9-18.

Prayudhi, B. (2016). *3F3D: Form Follows Force with 3D printing*, Master's thesis, Delft University of Technology, Netherlands, 11-58.

Rian, I., and Sassone, M. (2014). Tree-inspired dendriforms and fractal-like branching structures in architecture: A brief historical overview. *Frontiers of Architectural Research*, 3(3), 298-323. doi: 10.1016/j.foar.2014.03.006

Rodríguez, R., Nágera, M., Toniasso, P., and Slaviero, G. (2019, 11-14 Nov). *Topological optimization software tools: Literature review and real application*. The XL Ibero-Latin-American Congress on Computational Methods in Engineering, ABMEC., Natal/RN, Brazil.

Schittich, C. (2015). *Detail engineering 4: SOM structural engineering* (1st Edition). Germany: Institut für internationale Architektur-Dok, 112-115.

Shao, G. (2020). Comparison of BESO and SIMP to do structural topology optimization in discrete digital design, and then combine them into a hybrid method. In P. F. Yuan, M. Xie, N. Leach, J. Yao, & X. Wang (Eds.), *Architectural intelligence: selected papers from the 1st International conference on computational design and robotic fabrication (CDRF 2019)*, 219-232. doi: 10.1007/978-981-15-6568-7_14

Sigmund, O., and Maute, K. (2013). Topology optimization approaches. *Structural and Multidisciplinary Optimization*, 48(6), 1031-1055. doi: 10.1007/s00158-013-0978-6

Snijder, A. H., van der Linden, L. P. L., Goulas, C., Louter, C., and Nijsse, R. (2020). The glass swing: a vector active structure made of glass struts and 3D-printed steel nodes. *Glass Structures & Engineering*, 5(1), 99-116. doi: 10.1007/s40940-019-00110-9

Søndergaard, A., Amir, O., Eversmann, P., Piskorec, L., Stan, F., Gramazio, F., and Kohler, M. (2016). Topology optimization and robotic fabrication of advanced timber space-frame structures. In D. Reinhardt, R. Saunders, & J. Burry (Eds.), *Robotic fabrication in architecture, art and design 2016*, 190-203. doi: 10.1007/978-3-319-26378-6_14

Søndergaard, A., Amir, O., and Knauss, M. (2013, 24-26 Oct). *Topology optimization and digital assembly of advanced space-frame structures*. ACADIA 2013, Cambridge, Ontario.

Søndergaard, A., and Dombernowsky, P. (2017). Unikabeton prototype: Making digital architecture. In R. Glynn & B. Sheil (Eds.), *Fabricate 2011: Making digital architecture*, 56-61. doi: 10.14324/111.9781787352131

Søndergaard, A., Feringa, J., Stan, F., and Maier, D. (2018). Robotic abrasive wire cutting of polymerized styrene formwork systems for cost-effective realization of topology-optimized concrete structures. *Construction Robotics*, 2(1), 81-92. doi: 10.1007/s41693-018-0016-8

Stefanaki, M. I. (2020). *Glass Giants, Mass-optimized massive cast glass slab*, Master's thesis, Delft University of Technology, Netherlands, 8-17.

Stromberg, L., Beghini, A., Baker, W., and Paulino, G. (2011). Application of layout and topology optimization using pattern gradation for the conceptual design of buildings. *Structural and Multidisciplinary Optimization*, 43, 165-180. doi: 10.1007/s00158-010-0563-1

Stromberg, L. L., Beghini, A., Baker, W. F., and Paulino, G. H. (2012). Topology optimization for braced frames: Combining continuum and beam/column elements. *Engineering Structures*, 37, 106-124. doi: 10.1016/j.engstruct.2011.12.034

Sutradhar, A., Paulino, G. H., Miller, M. J., and Nguyen, T. H. (2010). Topological optimization for designing patient-specific large craniofacial segmental bone replacements. *Proceedings of the National Academy of Sciences*, 107(30), 13222-13227. doi: 10.1073/pnas.1001208107

Tyflopoulos, E., Flem, D., Steinert, M., and Olsen, A. (2018, 14-17 Aug). *State of the art of generative design and topology optimization and potential research needs*. NordDesign 2018, Linköping, Sweden.

Tyflopoulos, E., and Steinert, M. (2022). A comparative study of the application of different commercial software for topology optimization. *Applied Sciences*, 12(2), 611-633. doi: 10.3390/app12020611

Internet: Topology (Cambridge Dictionary), URL-1: https://dictionary.cambridge.org/dictionary/english/topology, Access Date: 20.11.2022

Internet: Topology (Collins Dictionary), URL-2: https://www.collinsdictionary.com/dictionary/english/topology, Access Date: 20.11.2022

Internet: Optimization (Collins Dictionary), URL-3: https://www.collinsdictionary.com/dictionary/english/optimization, Access Date: 20.11.2022

Internet: Optimization (Merriam Webster Dictionary), URL-4: https://www.merriam-webster.com/dictionary/optimization, Access Date: 20.11.2022

Internet: Millipede, URL-5: https://www.creativemutation.com/millipede, Access Date: 15.12.2022

Internet: Mesh or surface, understanding the difference (McNeel), URL-6: https://discourse.mcneel.com/t/mesh-or-surface-understanding-the-difference/75151, Access Date: 15.01.2023

Internet: SOM: 100 Mount Street, URL-7: https://www.som.com/projects/100-mount-street/, Access Date: 29.10.2022

Vantyghem, G., De Corte, W., Shakour, E., and Amir, O. (2020). 3D printing of a post-tensioned concrete girder designed by topology optimization. *Automation in Construction*, 112. doi: 10.1016/j.autcon.2020.103084

Vlah, D., Žavbi, R., and Vukašinović, N. (2020). Evaluation of topology optimization and generative design tools as support for conceptual design. *Proceedings of the Design Society*, 1, 451-460. doi: 10.1017/dsd.2020.165

Wang, H., Du, W., Zhao, Y., Wang, Y., Hao, R., and Yang, M. (2021). Joints for treelike column structures based on generative design and additive manufacturing. *Journal of Constructional Steel Research*, 184. doi: 10.1016/j.jcsr.2021.106794

Wang, L., Du, W. F., He, P., and Yang, M. (2020). Topology optimization and 3D printing of three-branch joints in treelike structures. *Journal of Structural Engineering*, 146(1). doi: 10.1061/(asce)st.1943-541x.0002454

Woo, S. J. (2020). *An Architectural Implementation of Topology Optimization Guided Discrete Structures with Customized Geometric Constraints*, Master's thesis, University of Waterloo, Canada, 1-34.

Xie, Y. M. (2022). Generalized topology optimization for architectural design. *Architectural Intelligence*, 1(1). doi: 10.1007/s44223-022-00003-y

Yan, X., Bao, D., Zhou, Y., Xie, Y., and Cui, T. (2022). Detail control strategies for topology optimization in architectural design and development. *Frontiers of Architectural Research*, 11(2), 340-356. doi: 10.1016/j.foar.2021.11.001

Yoely, Y. M., Amir, O., and Hanniel, I. (2018). Topology and shape optimization with explicit geometric constraints using a spline-based representation and a fixed grid. *Procedia Manufacturing*, 21, 189-196. doi: 10.1016/j.promfg.2018.02.110

Yuksel, O. (2019). An overview on topology optimization methods employed in structural engineering. *Kırklareli University Journal of Engineering and Science*, 5, 159-175. doi: 10.34186/klujes.606666

Zeeman, E. C. (1966). *An introduction to topology: The classification theorem for surfaces* (1st Edition). Coventry: Mathematics Institute, University of Warwick, 1-26.

Zegard, T., Hartz, C., Mazurek, A., and Baker, W. F. (2020). Advancing building engineering through structural and topology optimization. *Structural and Multidisciplinary Optimization*, 62(2), 915-935. doi: 10.1007/s00158-020-02506-6

Zhou, Q., Shen, W., Wang, J., Zhou, Y. Y., and Xie, Y. M. (2018, 16-20 Jul). *Ameba: A new topology optimization tool for architectural design*. IASS Annual Symposium, Boston, United States.

Zhu, J., Zhou, H., Wang, C., Zhou, L., Yuan, S., and Zhang, W. (2021). A review of topology optimization for additive manufacturing: Status and challenges. *Chinese Journal of Aeronautics*, 34(1), 91-110. doi: 10.1016/j.cja.2020.09.020

Gazili olmak ayrıcalıktır