/Y.U.S.F./

-Your Urban Structural Forestry-

An algorithmic approach to tailor-fit cutting pattern generation for timber loadbearing construction elements, using urban trees.

Frank de Zwart

January.2026

TU Delft Delft University of Technology
Faculty of Architecture
and the Built Environment

Abstract

The research question this paper answers is: ‘’How can felled urban trees be processed into tailor-made load-bearing architectural elements using computational optimization?’’. This question is relevant because urban trees are currently an under-utilized material. The built environment accounts for up to 40% of the energy demand. Timber structures can have a Global Warming Potential ten times lower then steel structures, when taking the embodied carbon into account. This paper has found that a best-fit heuristic for 3D bin-packing could result in a 30-40% utilization of roundwood timber for lead-bearing elements. Metaheuristics can improve the utilization by several percent. The trees marked for felling in Rotterdam could supply enough wood in half a year to supply for both small scale residential dwellings as well as for large scale commercial buildings. These findings are a prove of concept for a framework that utilizes urban trees into tailor made load bearing elements.

Key words – Structural timber, urban forestry, material allocation, Stock utilization, Rotterdam Netherlands, 3D-Bin Packing, Timber cutting pattern generation

/Y.U.S.F./

-Your Urban Structural Forestry-

An algorithmic approach to tailor-fit cutting pattern generation for timber loadbearing construction elements, using urban trees.

TU DELFT
MSC ARCHITECTURE, URBANISM & BUILDING SCIENCES

BUILDING TECHNOLOGY GRADUATION STUDIO

<u>Mentor Team:</u>

M1: Dr. Stijn Brancart

M2: Dr. Gabriele Mirra

DE: Ir. Henri van Bennekom

<u>Student:</u>

Frank de Zwart
4860683

<u>Date:</u>

January.2026

Delft University of Technology
Faculty of Architecture
and the Built Environment

Preface

This master's thesis report concludes my Masters Building Technology at the TU-Delft. This thesis marks the end of my time as a master student. I have really enjoyed the last 7 years I spend in Delft as a student, friend and roommate. I will look back on joyous memories and valuable lessons; both academically as personally. My time in Delft has been an enriching experience. Last year has been in theme of this master's thesis. The topic of which I really enjoy. This thesis has combined my interest in timber engineering, circular design and computational design. Last year has also been challenging, since I have never worked a project of this scale before. Luckily I did not have to go through it alone.

I would like to thank my mentors Stijn Brancart & Gabriele Mirra for their time, expertise and energy to help guide me through this project.

Stijn Brancart, thank you for putting the topic of urban trees to my attention and for your expertise in timber engineering. Meeting you weekly during Q3 has been of great help to me to get my thesis going.

Gabriele Mirra – with your vast knowledge of computational design you have helped me be critical about my approach and the way I present my findings. When I was struggling to put my computational framework into the right words you always stayed sharp and have helped me build a clear vocabulary to describe my work more accurately and precisely. You have given me insight and guidance translating my computational workflow to explanatory figures and text.

I would also like to thank my fellow students, friends and roommates for the discussions we have had and feedback they have given me. Together you have greatly enhanced my time as a student by pushing me, and creating cherished memories. I am lucky to have met amazing people. In particular I would like to thank my roommate Joris Hoogeweegen , for being involved with my thesis and brainstorming with me in every phase of this project. Joris, it has been a pleasure bouncing my ideas off you.

I am grateful to my partner for her love, encouragement, and understanding during the completion of this thesis. Finally, I would like to thank my family for supporting me throughout my endeavors as a student, both emotionally as in practical ways.

It has been a privilege to be able to enjoy such a great education.

Delft, January 2026
Frank de Zwart

Table of contents


01
Introduction

Chapter 1: Introduction

1.1 Context

Every month around 100 trees are being cut down in Rotterdam because they are unwanted, sick or pose safety risks. These trees are broken down or being shredded. This is a big waste a large portion of this wood could see use in timber construction. The disposal of this wood is especially wasteful since most of these trees are hardwood and can take a long time to grow.

At the same time the build environment could use more timber building to contribute to reaching the Paris Emissions Agreement. Global warming is a big problem and it is becoming more eminent to change the way we build. The built environment currently accounts for up to 40% of the energy demand (Küpfer et al., 2021) and 50% of the total material consumption in Europe. (Brütting et al., 2019) Timber building practices can help cut these numbers down. A study by Hemmati et al. (2024) show that the global warming potential (GWP) contribution of a steel structure can be more then ten-fold that of a timber structure if you consider the embodied carbon of wood.

By combining the under-utilized trees in Rotterdam with the timber construction there is benefit to be made in the carbon emissions of local building practices. In the current architectural design workflow material usually follow design. In the case of European timber construction this usually leads to commercially available softwood from Scandinavia or engineered timber. By creating a tool that can automatically link a design to a dataset of urban trees a new framework could arise that would allow for a hybrid designing pipeline utilizing city wood. This paper will research such a framework and ideally design a tool that would allow for this hybrid designing process.

1.2 Knowledge gap

The supply of timber is dynamic and changes for every month. To couple this with a design workflow that will automatically assist an architect / engineer it is required that such a tool is parametric. Within parametric design there is substantial research has been carried out in regards to stock-constrained design. As the name suggests stock-constrained design focuses on constraining a design to a certain set of predefined element (stock). The research of stock-constrained design can usually be subdivided into three categories. One where the stock can modified to fit the design, one where the structural outcome design is the result of the limited stock availability and in the last category the design is constrained by the geometric shape of the available stock elements. The first category, where the stock is modified to fit the design comes closest to the framework that will be proposed in this paper. The existing research however primarily focuses on the one-dimensional modification of the stock (Brütting et al.,2019). This is relevant for materials like steel where the length and width of the material is predefined. This paper would like to address the knowledge gap regarding the three-dimensional cutting of stock without focusing on complex geometries.

1.3 Research aim and objectives

The aim of this research is to explore a framework that would allow for a structural design to be automatically linked to a underutilized urban timber stock and generate possible material allocations along with 3D cutting patterns. The objective is to create a tool that would allow for 3D allocation of load-bearing elements in the wood stock of Rotterdam's urban trees. This tool will be tested with different parameters & goals to test the flexibility and limits of such program.

1.4 Research questions

To reach these objectives the primary question of this thesis should be answered. The main research question is: *How can felled urban trees be processed into tailor-made load-bearing architectural elements using computational optimization?'*

To answer this question, it is divided into the following three sub-questions:

1. What could the urban felled wood in Rotterdam supply and what are the structural qualities of these trees?
2. How to parameterize the dimensions of loadbearing elements in a structural design to suit the framework?
3. How to allocate the loadbearing elements into the available timber stock?

Answering these sub-questions is expected to provide sufficient insight to advice on, or even create a framework like discussed in the objectives.

1.5 Scope and limitations

The scope of the research is restricted to ensure that the work can be carried out within a reasonable time period and to refrain from straying too far from the objectives of a master thesis. The study is limited to:

- Felled urban trees located within the municipality of Rotterdam.
- Structural analysis based on Euler-buckling, Compression Strength, Bending Moment, Longitudinal Shear & Fire-safety.
- Sawn-wood timber for load bearing elements.
- Dimensioning based on required strength and length, disregarding joining techniques.
- A simplification of trees into a homogeneous cylinder.

These limitations mean that the results should be interpreted as an exploratory assessment of the structural potential of felled urban trees within an optimistic framework, rather than as a fully validated design tool ready for direct implementation in practice.

1.6 Significance

The findings in this paper could prove useful for architectural designer and the wood industry. By creating a methodology by which city timber can be utilized, further development could lead to marketable possibilities for urban wood. This paper can be used as a proof of concept for a pipeline by which architects can choose specific trees for their design before they are felled.

The results of this paper could motivate municipalities to enhance the cataloguing of their trees to improve commercial use of city timber. By including set of dimensions for every tree that is marked for felling, a more applicable and marketable tool could arise.

From an academic perspective this paper touches a new area of stock-constrained design in which the 3D milling of wood is integrated into the use of a timber stock. The findings in this paper stimulate further research and contribute to the development of more comprehensive frameworks.

1.7 Thesis outline

To answer the research questions this thesis is set up in five parts. At first a literature research will be carried out. The literature research will start off with a background on wood to get familiar with the main material this paper is about. This literature will touch subjects like the biological behavior of trees, this will be neglected in the primary pipeline, but will see some implication a parameter research. In order to justify the limitation to mere sawn-timber the literature research will compare sawn-timber to round-timber and engineered timber. This will be followed up by a study into cutting techniques and timber conversion yield. The second part of the literature research consists of reference projects. Discussing previous work regarding stock-utilization and bin-packing solutions.

The literature review will be follow up with the methodology by which this thesis will produce its results. The methodology introduces the reader to the overall framework proposal. It explains how the stock of city trees will be integrated. The second chapter of the methodology shows by which structural calculations the timber elements will be dimensioned and how the required formula's are parameterized. The third part of the methodology explains the algorithmic heuristic by which the initial timber allocation is performed. This is followed by an explanation of the workings of the custom genetic algorithm used in the framework along with the scoring and hyperparameters used for genetic optimization. The methodology ends with an introduction to the parameter exploration that is part of the results. For the parameter exploration seven different situations will be tested, along with two alternative approaches for genetic optimization. These seven different scenario's are: optimizing by inclusion of weighted timber species scoring, exploring a more extreme reference building, the inclusion of biological behavior of wood, the inclusion of non loadbearing elements into the element database, rounded element dimensions, the preservation of tall trees and experimentation with different hyperparameters.

Chapter four will introduce the case-study of the wood supply as seen in Rotterdam. This chapter shows an overview of the structural characteristics for the eight wood species that will be used in this research.

The results of performing the methodology in combination with the case-study of Rotterdam's will be shown in the fifth chapter. The chapter will start of with a short introduction to the structures make up the required element database. The structures will be timber-frames increasing in complexity, eventually leading to an impressions of the loadbearing structure of a realized dwelling. First the results regarding element dimensioning are laid out. This is followed by the solutions the heuristic program gives for each structure. These results are then compared with the findings of the meta-heuristic. The results chapter ends with the parameter exploration, where the results for all eight different parameter situations are shown.

The results are discussed in the next chapter. Here the findings of the results are evaluated and explained. The discussion will go over the findings in the order as they are presented in the results chapter, while also cross-referencing results to draw conclusions from them.

This thesis finishes up with a conclusion where the main research question is answered and suggestions for further research are given. The conclusion goes over the main findings and limitations of the research and rates them in relation to real world applicability.

02 Background

Chapter 2: Background & State of the Art

/C2.1: Background on Wood & Timber

/2.1.1: Background on wood biologics

What are the different wood categories, what determines these categories and how do they behave?

Wood is commonly categorized in two categories: coniferous wood and deciduous wood. The seed type of a tree determines its category. Coniferous wood and deciduous wood are industrially also called softwood and hardwood, respectively. Since it is the seed that determines whether it is softwood or hardwood and not their structural properties, these names can be misleading. There is hardwood that is more soft then certain softwood types. (Ramage et al., 2016)

Other than primary growth, growth in length, trees also undergo secondary growth, in width, which sets them apart from most plants. The cells a tree makes in secondary growth changes characteristics with changes in its environment; commonly the change of seasons. In spring the tree creates large cells with thin walls, making it easy to transport water and boost photosynthesis. These cell layers are called 'earlywood' and are less dense than the layers of cells from colder seasons. In these layers the cells have thicker layers and are more densely packed. This change in cell density is what you see in the annual rings and can show the age of a tree. The outer layers of a tree are responsible for water conductivity. This part of the tree is called sapwood. When a tree continues growing the oldest sapwood tissue dies and becomes heartwood. Heartwood has different technical properties than sapwood. (Ramage et al., 2016) The heartwood fills its pores with biochemicals making it more stiff and more durable. (Bekin, 2019)

The structure of the cell walls is one of the major determinants that make up the of strength and mechanical properties for wood. There are some factors that impact and change the structure of the cell walls. One of these factors it the formation of knots. A knot forms when the trunk grows around a branch. As the tree grows in width, it envelops the base of the branch. If the branch is alive, it becomes a tight knot integrated into the wood. If the branch dies before being fully enclosed, it forms a loose knot, and it may decay or fall out. The structure of the cells around the knot are no longer linear but have a distorted grain direction.

Another factor is the formation of 'reaction wood'. Reaction wood form as a response to non-vertical loads, like strong winds or an asymmetric crown shape. The load leads to compression on one side of the tree and tension on the other. The cell structure of the tree follows this compression and tension and creates denser and less dense wood accordingly. Reaction wood alters the uniform structural properties of timber and can twist, cup or warp dramatically during machining. A similar but less intense situation is seen in the twisted growth of trees. Wood cells, especially in soft woods, are rarely truly vertical. They usually twist in one direction. This spiraling grain results in twisted sawn timber after drying. (Ramage et al., 2016)

Depending on species and factors like stress direction and moisture content the technical performance and strength of timber can very over a hundred-fold. (Arup, 2019)

/2.1.2: Round .vs. Sawn .vs. Engineered timber

To go from wooden logs to construction material these are several ways timber can be used. The least modified way is to use the wood as ‘Round timber’, also known as whole timber. In this form the logs are primarily stripped of their branches and debarked but are still used almost as a whole. It is common for round timber to undergo some form processing, like rounding. By rounding the logs are being spun around and milled, in order to create a more linear, round homogenous product. (Bukauskas et al., 2019)

When the logs are cut into pieces we are dealing with ‘Sawn timber’, also called solid sawn timber. Sawn timber is generally processed using longitudinal saw cuts parallel to the trunk axis. (Woodard & Milner, 2016) Sawn timber can result in e.g. large square columns but is most commonly known for rectangular planks. Sawn timber usually consists mostly of heartwood.

Engineered timber is created by joining separate timber pieces together to form structurally optimized building materials. Engineered timber is the most processed material of the three. There different ways to make engineered timber but the most common ones are Parallel Stand Lumber (PSL), Laminated Veneer Lumber (LVL), I-joist, Glue laminated (GluLam), Cross Laminated Timber (CLT) and Brettstappel / Dowel Laminated Timber (DLT). In figure 1 the six engineered timber types are shown along with their application. (Ramage et al., 2016)

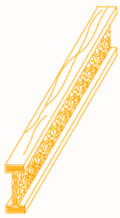
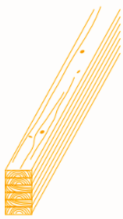
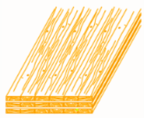
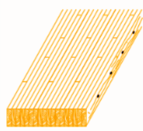
Engineered Timber Product	Parallel Strand Lumber (PSL)	Laminated Veneer Lumber (LVL)	I-Joist	Glulam	Cross Laminated Timber (CLT)	Brettstappel
Typical Detail						
Application	• Beams • Columns	• Beam • Columns • Cord	• Joist • Beam	• Beam (Long span) • High Loading	• Roof • Wall • Floor	• Roof • Wall • Floor
Usage	Interior	Interior	Interior	Interior / Exterior	Interior/ Exterior	Interior/ Exterior

Figure 1: Different types of engineered timber and their application. (Ramage et al., 2016)

All three timber usages have their own pro’s and cons. According to Ramage et al. (2016) various studies have demonstrated that unsawn timbers have higher and less variable bending strengths than sawn timbers. Round timber is more dimensionally stable than sawn timber. (Bukauskas et al., 2017) Another claim is that round timber is less vulnerable for warping since the tree keeps both their sapwood and heartwood structure. A primary reason for using unsawn timber in this project is that it preserves the wood as high up in the reuse chain as possible. By keeping as much of the log untouched the timber keeps most of its value and flexibility in processing options, while it can always be downcycled into lower grade timber products in a new reuse stage. One of the big downsides to using round timber is the irregularity in shape. Without the process of sawing the timber in uniform shapes a set of round timber becomes a library of unique materials making it challenging to create a design and allocating the logs.

This is not a problem in sawn timber, where the logs can be cut into desired dimensions (within the restriction of the tee size). Sawn timber can be less strong then whole timber. This can be partially explained by the fact that a linear longitudinal cut does not always follow the exact flow of the grains in the log. (Woodard & Milner, 2016) When the logs are cut this creates waste and the timber is not used optimally. It could be one of the goals for a framework to minimize the amount of cutoffs and maximize wood usage. Another problem with sawn timber is that it is susceptible for warping during the drying process.

For engineered timber the pro's and con's depend on engineering type and on different factors like wood species and size. But in general some statements can be made regarding engineered wood. First of all it is the most processed timber type of the three meaning it is lower on the wood utilization cascade. It also required more machining and thereby more labor plus all these engineering types except for DLT require also use some sort of glue which decreases their ecological performance. Engineered timber however can be made to be very strong, even from wood species that would otherwise under perform in loadbearing capacities. When considering the option of hybrid engineered timber, combining higher and lower performing timber, the possibilities are even bigger. For the scope of this project LVL and GluLam would be most relevant. CLT and DLT can be eliminated from the fact that the opted design consists only out of beams and columns, and no load bearing walls will be taken into consideration. I-joist and PSL require to much processing and stray to far from the utilization cascade.

Round Timber

+ High utilization

+ Better loadbearing

- Adds complexity from the irregular shapes

Sawn Timber

+ More uniformity in shapes

- Utilizes less of the log

Engineered Timber

+ Can be shaped to liking

+ High & 'custom' loadbearing capacities

- Endless possibilities

- Lower in the utilization cascade

Figure 2: Overview pro's & con's Round, Sawn and Engineered timber

For this research the choice will be made to exclusively focus on sawn timber. Sawn timber would allow for a more uniform sizes then round timber, which makes it is use more realistic in the current market. Sawn timber is also high up in the wood cascade meaning it could see more recycle steps in its lifecycle. Sawn timber also has clear limitations regarding size. This makes designing a framework for the utilization of such timber more manageable.

/2.1.3: Timber conversion yield and cutting patterns

In the production of construction-grade sawn timber, typically only half of the log's original volume is recovered as usable lumber. For instance, Brandstetter et al. (2020) gathered data from sawmills in 21 countries and found that on average 53% of a log's volume could be converted to sawn wood. Approximately 30% of the log results into chips, 12% becomes sawdust and the remaining 5% are shavings or shrinkage loss. The sawmills in this data set handle coniferous wood. Another study by Negeo et al. (2024) had similar findings for Eucalyptus hardwood. In this paper a lumber recovery rate of 49% is recorded. These figures align with the findings by Clark et al. (1974), who reported an average yield of 54% from the milling of 47 yellow poplar trees.

When milling logs into small, non loadbearing elements, higher average yields are achievable since smaller pieces allow for more flexibility in cutting patterns. As can be seen in Klement et al. (2023), who reported conversion rates of 65% to 72% when processing Beech logs into 5x5cm blanks.

There are numerous standard cutting techniques for timber processing. The placement of a cut can heavily influence the quality of the timber. The primary cause for difference in quality is the grain of the wood. Timber elements that have radial graining are more prone to warping and have a lower structural performance. Depending on preference and the required timber elements common cutting techniques for planks are: plain sawn, quarter sawn & rift sawn, as shown in figure 3.

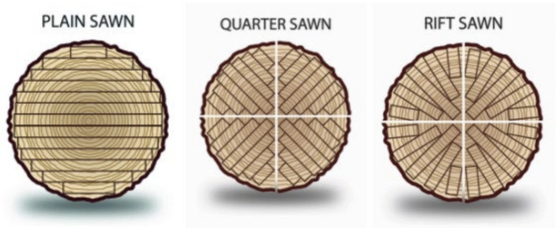


Figure 3: a visualization of plain, quarter and rift sawing techniques
Retrieved from: https://shorturl.at/VSoB4 ***2**

Bigger elements like columns and beams require different cutting techniques. Figure 4 shows methods like cant sawing (1), grade sawing (4), asymmetrical sawing (6) and component sawing (7).

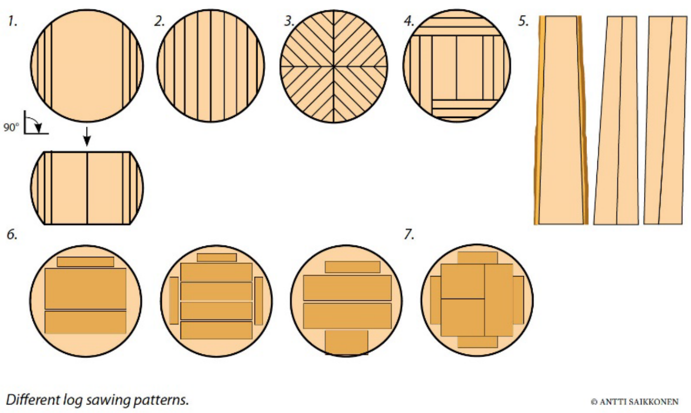


Figure 4: Different log sawing patterns for bigger elements
Retrieved from: https://shorturl.at/VSoB4 ***3**

For the highest grading timber the outer sapwood ring and the core pith should be avoided. The grade sawing method (4) in figure 4 shows this utilization. By cutting trough the middle the pith is avoided and the columns in the center do not overlap with the sapwood. Another approach in regard to the pith would be to box it in. If the pith is centered in the middle of a rectangle the warping on all four sides would cancel each other out.

Looking at component sawing (7) in figure 4 it can be seen that most of the edges of the bigger rectangles near the edge of the tree. This would mean that some sapwood is included in the final beam. This does not have big impact on the structural performance of such beam. Some sawmills even cut up to the bark, meaning the post would be slightly rounded off. This is called wane. A paper by Martitegui et al. (2007) shows that the bending strength of pieces with wanes is no lower than the strength of members without this flaw.

Software exists where splits in wood are detected, so new cutting patterns can be made to increase the yield. 3D scans are made in the sawmill for each tree in order to detect splitting. Figure 5 shows such a scan, including an updated cutting pattern to increase the yield. The imagine from figure 5 is 'USNR optimizer' software.

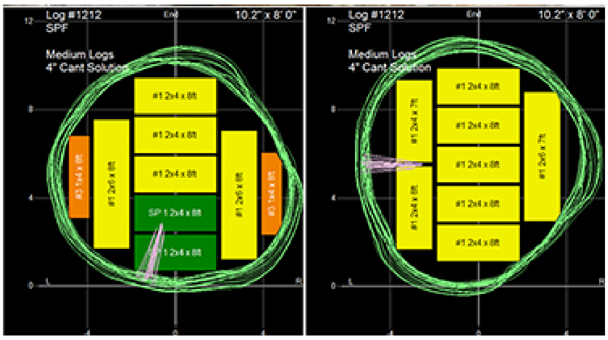


Figure 5: USNR's split detection system, including a new cutting pattern to increase yield.
Retrieved from: https://shorturl.at/QLrK9 ***4**

/C2.2: Reference projects

/2.2.1: Literature review: Stock Optimalisation

Considerable research has been conducted regarding strategies for stock constrained design. A categorization could be made to subdivide this research into three group:

- Stock constrained design where stock is modified to fit the design.
- Stock constrained design where the design follows the length and strength performance of the available stock
- Stock constrained design where the design follows the complex geometry of the stock

For the first two categories it is commonly seen that the stock consists of a set of linear one-dimensional elements with different properties. As seen in Brütting et al., 2020. The most important properties being length and strength. The input is a desired structural framework, with specific loads. The main criterium is to allocate the linear elements to the framework in a way that the strengths of the reused material would withstand the loads from the desired structure. The strategies for the allocation can depend on different goals; like minimizing cut-off waste or the weight of the structure. Warmuth et al. (2021) present a computational tool that creates solutions for the 1D element allocation by using Multiple Integer Linear Programming (MILP). MILP is a type of optimization by which a solution can be calculated by a series of linear mathematical equations. Solvers explore many combinations of variables to minimize or maximize the objective. This can require heavy computation but it guarantees a globally optimal solution. A heuristic approach is tested against the MILP. The findings of this paper show that the heuristic best-fit approach allows for near real-time implementation with its quick computation. The trade-off for solution quality to computation time suggests the heuristic is best for a initial design tool, where the MILP would prove more useful in calculating solutions for more final designs.

Warmuth et al. (2024) look into stock constrained design and prioritize the research into shape optimization. With shape optimization their computational framework is able to give a design a rework that uses 99% of the available stock whereas the original would reassign 86%.

Van Marcke et al. (2024) add a new aspect to one-dimensional framework from above. In their research the elements from a reused truss are taken as a complete triangular shape. Adding a new level of dimension to its framework. By using predetermined shapes the allocation solutions could be simplified.

The utilization of complex geometry in stock is addressed in the ‘Tree fork truss’ paper by Zachary and Martin (2016). The scope of this research is to generate one arching truss from a set of tree forks. The tree forks used in the truss are made from trees that were not cut yet. By making photos of trees a 2D database could be made of tree forks with potential for the truss. In order to get the input for a complete computational model 3D-scans were made of each tree. This shows a big bottleneck for the implementation of shape into the framework. The 3D scans required a hands-on approach and in this case for the trees the be cut down. Meaning this workflow would be unpractical for the case-study of this paper, unless new developments would be made where the municipality would include 3D scans within its GIS database.

A more automated method is discussed by Bukauskas et al. (2017). They present the idea of 'Gate constraints' see figure 6**.** The gate constraints allow for some geometric diversity in the structural elements but rule out elements that bend to much for a linear approach to be valid. This paper uses the geometry with several different first-fit and best-fit heuristics to test for basic form fitting. The formfitting could be simplified into a 1D bin-packing problem. The results were rated based wasted length, items remaining and bins used. In this research the bins are considered to be the roundwood logs are determined by the gate constrained in figure 6**.** The best performing heuristic on bin use was a best-fit approach with items pre-sorted on decreasing compression / tensile strength and minimizing remaining length.

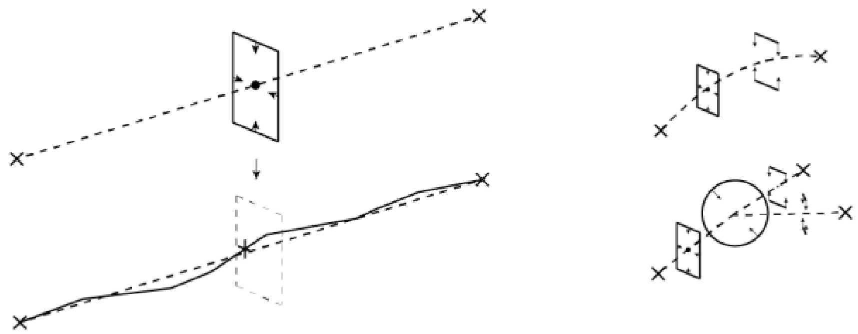
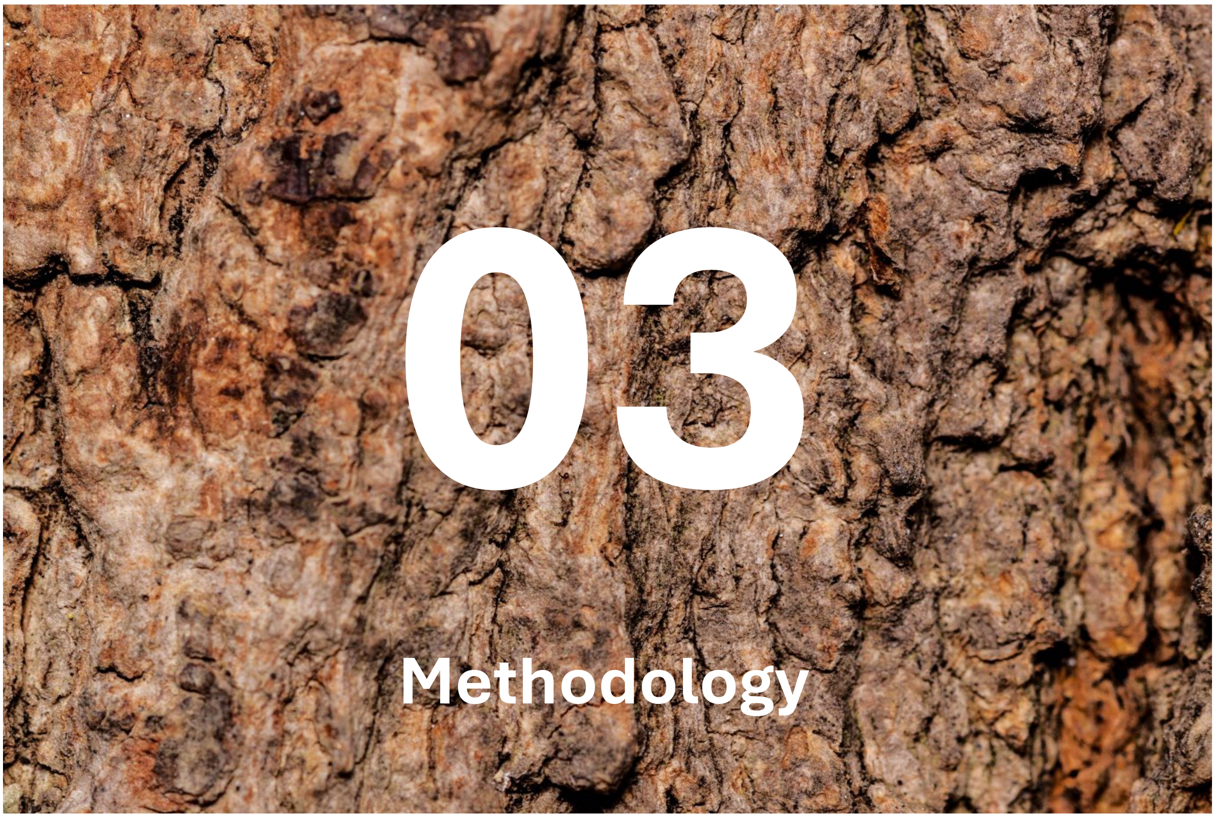


Figure 6: 'Gate' constraints. As presented by Bukauskas et al. (2017).

/2.2.1: Literature review: 2D/3D Bin packing

The problems in the aforementioned papers can primarily be reduced to 1D bin-packing, since the stock was reduced to simple linear elements. This thesis is set out to explore bin-packing within a 3D representation of a log. This means that the heuristic as proposed by Jan Brütting et al. (2019) and Warmuth et al. (2021) is not applicable for this research.

The framework for this research would come closer to the problem as proposed by Hinostroza et al. (2013). This paper argues that nonlinear mixed integer programming could be used to tackle small scale 2D bin-packing, but for larger problems the use of heuristics is most viable. The heuristic is combined with a simulated annealing (SA) metaheuristic. The heuristic delivers a packing yield of 91.3%. The optimization with SA increases this yield to 93.6%. Another SA optimization for bin-packing is seen in the research by Tole et al. (2023). In this paper a First-fit Grid Search algorithm is used. This grid search algorithm would allow for tightly packed bins but would not comply to guillotine packing logic. Which is a requirement for the case study in this thesis in order to allow for traditional sawmilling.


03
Methodology

Chapter 3: Methodology

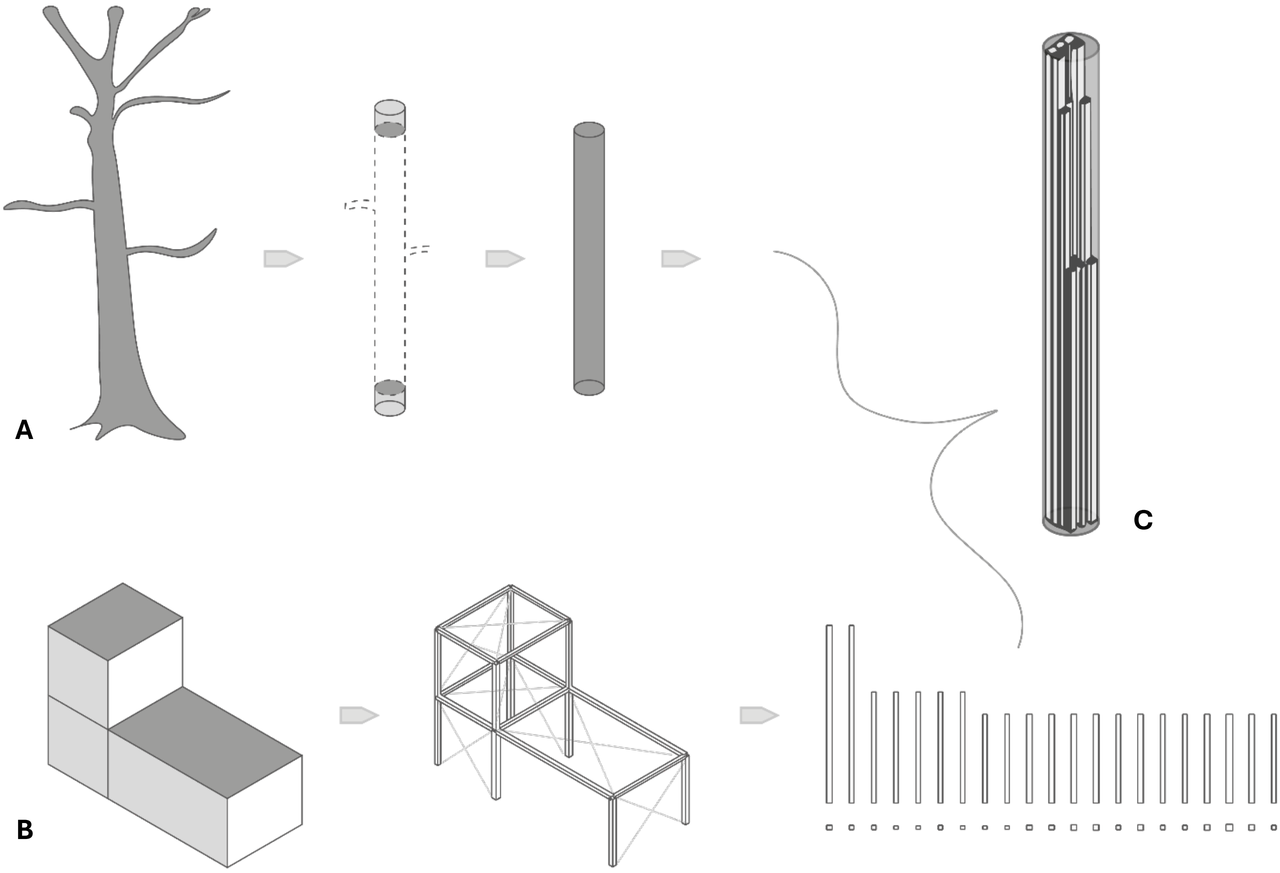


Figure 7: Schematic overview of the methodology.

The method by which the research will be conducted can be divided into 3 segments, see figure 7. Part A consists of setting together the stock of city trees that represent the upcoming felling. The second part of the framework (B) calculates the dimensions the timber elements require to realize the design. This creates a set of elements regarding each timber species. In the 3rd phase (C) the elements from B are to packed into the logs from the database in part A. By running a heuristic program combined with metaheuristic a prediction can be made on the required amount of trees, of which species these can be, and how to handle them in milling.

/C3.1: Stock of city trees

A database representing city trees will be the starting point for the algorithm. The trees will be geometrically simplified to cylinders with a height and diameter. The height of such cylinder will represent the trunk of the tree before branching off into its crown. The diameter of the cylinder is to match the smallest diameter part of the tree trunk.

The amount of trees for each wood species should be readily available in the GIS database of the municipality. This information is to be downloaded as a JSON file to allow for simple integration within the algorithmic software.

/C3.2: Dimensioning timber elements

/3.2.1 Structural design formulas

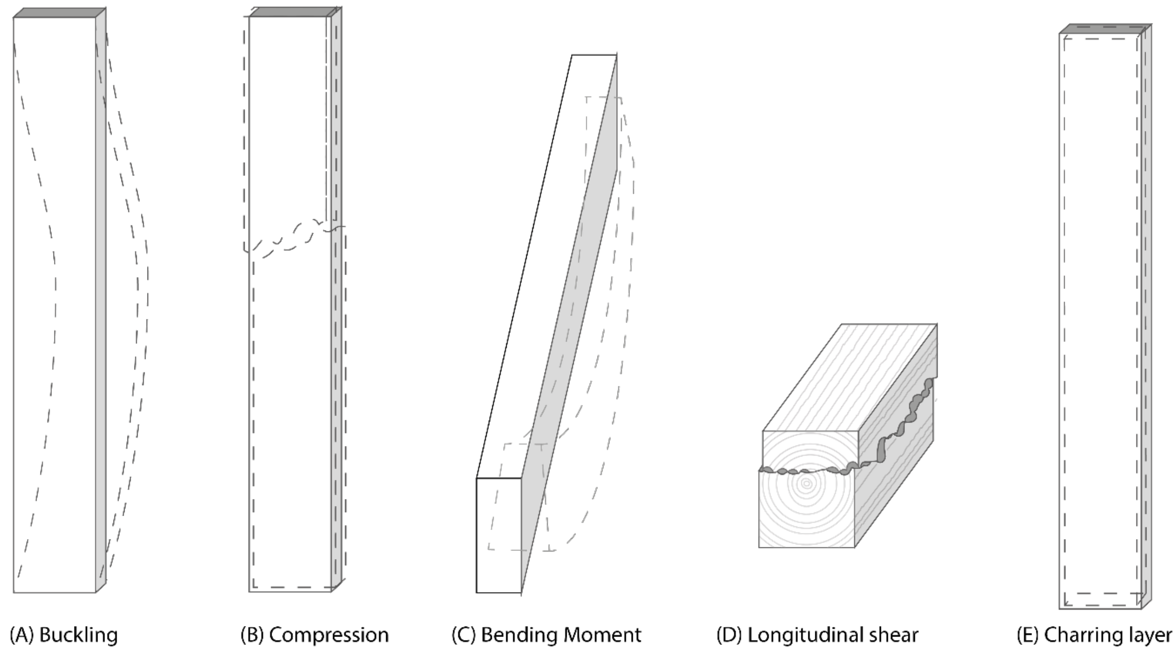


Figure 8**:** A visual representation of the five structural design formulas that will be used to govern the element dimensions.

To suit the unpredictability and the differences of an irregular city wood stock the formula's for structural design will be rewritten and modified to allow for parametric design. In order to do this the formula's will be rewritten to a variant where the width of an element is a product of the characteristic load bearing abilities of a wood species together with a factor x. The factor x is meant to parameterize the ratio of width to height. If x is 2 that would mean the ratio of height to width would be 1:2. This parametrization means that for each element there would be:

$$N_{element\ variants} = N_{species} \times N_{factors}$$

In the paragraph below the rewritten and modified version for five structural design formulas are given in a short fashion. In the **appendix** the full conversion can be found for each 'traditional' formula to the parametric version used in this research.

/3.2.2: Column dimensioning

3.2.2.1 Buckling (A)

For the buckling calculations all column elements are regarded as pin ended struts meaning L_e=L. The standard Euler Buckling formula for perfect axial loading is:

$$Pcrit = \frac{(\pi^2 EI)}{(Le)^2}$$

Since loads are rarely perfectly axial a factor of 5 is added to the P*crit*.

$$Pcrit \ = \ \frac{(\pi^2 EI)}{5(Le)^2}$$

Required column width as a product of non-perfect axial buckling with a ratio-factor x:

$$b = \sqrt[4]{\frac{60Pcrit\ (Le)^2}{xE\pi^2}}$$

3.2.2.2 Compression strength (B)

The compression in the columns must not exceed the characteristic compressive strength.

$$\sigma_c = \frac{F}{A} \leq f_{c,0,d}$$

Required column width as a product of compression strength with a ratio-factor x:

$$b = \sqrt[2]{\frac{FA}{xf_{c,0,d}}}$$

/3.2.3: Beam dimensioning

The beam calculations are done under the assumption that the beams are laterally restrained against tipping, and therefore are not vulnerable to lateral torsion buckling.

3.2.3.1 Bending Moment (C)

When calculating for bending moment, the bending stress (σ_m) must be lower then the design bending strength of a material $f_{m,d}$.

$$\sigma_m = \frac{M}{W} \ \leq \ f_{m,d}$$

The minimum height a beam must be to withstand the bending stress according to a height-width ratio x is:

$$h = \sqrt[3]{\frac{6xM}{\sigma_m}}$$

3.2.3.2 Longitudinal shear equation

The standard formula to calculate shear stress in a beam is:

$$\tau = \frac{S\, A_c\, \bar{y}}{I\, b}$$

At the neutral axis the shear force is the highest meaning τ_{max} is at y = 0. Substituting A_c and $\bar{y}$ at level y = 0 together with I gives:

$$\tau_{\max} = \frac{3S}{2hb}$$

With $f_{v,k} \geq \tau_{\max}$ and ratio x the formula can be rewritten as:

$$b \geq \sqrt[2]{\frac{3S}{2f_{v,k}x}}$$

/3.2.4: Fire safety-Charring Layer

The main idea behind the fire safety calculations is similar to the calculations from above. The big difference is the assumption that a different safety factor can be used. The building needs to withstand the existing dead loads when on fire but doesn't need to sustain forces for a long period of time. Meaning that the safety factors change to 1.0 for dead loads, 0.5 for snow loads and 0.0 for wind loads. With these new safety factors the elements can be designed in a manner to withstand the loads at their minimum dimensions and a charring layer with a certain depth is added so that this volume could burn down and the elements would still hold. The formula for an effective charring layer is:

$$a_{\text{eff}} = \beta_n\, t + k_0$$

Eurocode 5 provides a charring rate β_n of 0.8 mm/min for solid timber (Table 3.1, EN 1995-1-2). This value is based on typical softwood properties. The website of *Designing Buildings (2023)* suggests that sawn hardwoods have charring rates closer to 0.5 mm/min. For now 60 minutes (R60) meaning *t* = 60. The value for k_0 is commonly 7 mm, as used in the Eurocode 5, but new research suggests that the value in practice is much higher. A research by Huč et al. (2020) had resulted into k_0 values of 8.4mm up to 30.4mm. Regardless of these new findings however, for this research model a k_0 of 7 mm is used.

$$a_{\text{eff}} = 0.5 * 60 + 7 \;=\; 37\; mm$$

This means that for every exposed side a charring layer of 37mm is desired in order to withstand accidental fire load combination. Assuming that for the timber frame the columns have four exposed sides and beams are exposed on three sides, the effective size for columns becomes:

$$b = b_{\text{eff}} + 2a_{\text{eff}}$$

$$h = h_{\text{eff}} + 2a_{\text{eff}}$$

For beams:

$$b = b_{\text{eff}} + 2a_{\text{eff}}$$

$$h = h_{\text{eff}} + a_{\text{eff}}$$

/3.2.5 Loads

In order to start assigning stock it is important to dimension the timber elements required for the loadbearing structure. To dimension the elements an estimation has to be made of the self-weight of the structure and the applying forces. The self-weight of the structure is unknown since the dimensions and the stock are not assigned yet. In the research model the self-weight is approximated by the C30 timber in a rectangular 20×20cm profile per running meter.

Furthermore characteristic load values are taken from the Eurocodes values for flat-roofed, commercial building in the Netherlands to determine the load in this research model (EN 1991-1-1:2002; EN 1991-1-3:2003; EN 1991-1-4:2005). These values can be easily adjusted in the model to suit different design cases.

./Loads used in the research model

___*Permanent Actions*___

-Self-weight roof:	3.0	kN/m^2
-Self-weight floor:	4.0	kN/m^2
-Self-weight structure:	0.162	kN/m

All forces are multiplied by 1,35 for the ULS

___*Variable Loads*___

-Snow load:	1.5	kN/m^2
-Wind load:	0.38	kN/m^2

All forces are multiplied by 1.5 for the ULS.

/3.2.6: Resulting forces on structure

To properly use the formulas as mentioned above it is important to have an overview of the resulting forces in the structure for each element. In order to get the correct right force distribution along every element the software program Karamba3D will be used. Karamba3D is a parametric structural engineering tool that simulates structural behavior under defined load cases. Karamba3D is embedded in the Grasshopper / Rhino 3D environment. By combining Grasshopper logic with Karamba3D a framework can be made where a structural frame can be automatically subdivided into beams and columns and be measured accordingly. This allows resulting forces and moments to be easily for traceable per element. The output per element include; Normal Force [kN], Shear Force Vz & Vy [kN], Torsional moment [kNm], Bending moment My & Mz [kNm].

By using the Grasshopper environment the Karamba3D outputs can directly be fed into the aforementioned design equations. This creates a dataset of possible dimensions for each timber species, indexed by each element. The dataset can be converted into a JSON for easy cross-handling with more advanced programming software like VSCode.

/C3.3: Cutting pattern generation - 3D bin packing

One of the biggest differences between a timber stock as reviewed in this paper compared with stock assigning solutions of other materials is that the logs can be cut in three dimensions. In other stock-constrained research the stock usually allows for 1D cutting. This is logical since most elements, for example steel beams, are only to be cut in length, as for their height and width is fixed. The 3 dimensional cutting of the wood stock allows for more options and thereby more complexity.

3.3.1: Problem formulation & restrictions

To generate cutting patterns the log will be populated with the required beam-column elements. This resembles the bin-packing problem. The logs are 3 dimensional cylindrical bins and to goal is to optimize the distribution (packing) of elements in these bins. Bin-packing is a well documented NP-Hard problem. A problem in the NP-class (Nondeterministic Polynomial Time) is a problem where the a solution is easy to validate but finding a solution could be computationally challenging. For NP-Hard problems there is no known algorithm that can guarantee finding a solution. In practice for NP-Hard problems as the problem size grow, the number of possible solutions grow exponentially. This makes it so brute-force solutions become impractical very quickly. For this reason a heuristic approach combined with metaheuristics will be used and tested.

In order to tie the bin packing algorithm to a realistic scenario several constraints are imposed. The main constraint is 'guillotine cutting'. Guillotine Cutting is the bin-packing terminology for cuts that go from edge to edge. This method of cutting is meant to represent how wood is processed in practice by sawmills. Another restriction is that each solution should enforce a certain kerf size in between packed elements. The kerf should mimic the thickness of the saw blade. Without enforcing kerf the cuts in the sawmill would chip away some thickness from each element and make it sawdust. Figure 9 is a visualization of these restrictions.

The problem can be summarized by saying:

- Logs are treated as 3D cylindrical bins defined by diameter and length
- Beam & Columns are rectangular elements defined by width, height and length.
- Restrictions:
  -Guillotine cutting
  -Enforce a kerf size

The primary objective is to maximize utilization, defined as:

$$Utilization = \frac{Total\ placed\ volume\ rectangular\ elements}{Total\ \ log\ volume}$$

A more diverse objective will be explored in *Chapter 3.4 – Parameter Exploration*, utilizing the lowest scoring logs first.

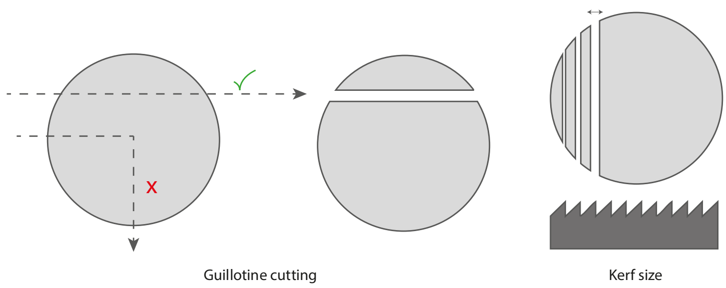


Figure 9: Visualization of the Guillotine cutting & Kerf size restrictions

3.3.2: Algorithmic approaches (3D Bin-packing)

3.3.2.1 Greedy Heuristic

The greedy heuristic will be the core component in the proposed bin packing algorithm. The heuristic will set the base mathematical rules by which the bins will be packed. In figure 10 the primary logic behind the greedy heuristic is made visual in a 2D plane. The idea is to create a rectangle at the center of the circle, essentially dividing the area into one main square and four circular segments. This division resembles the cutting behavior as seen in Cant sawing and component sawing. After this division the biggest element in the list is placed in the bottom left corner. In order to enforce guillotine cutting this element divides the center square into two new rectangles (1 & 2 in Fig. 10 C). This new division can be made in two ways; with a guillotine cut going up vertically or going across horizontally. In this heuristic the choice is made to always cut in a manner that would create the largest remaining rectangle. Then by the same logic new elements are placed inside the remaining space, again dividing the area into two by each placement. The four circular segments are also treated as four bins for small elements. Each segment is reduced to a rectangle with a height equal to 1/10th the length of the original diameter, and filled accordingly.

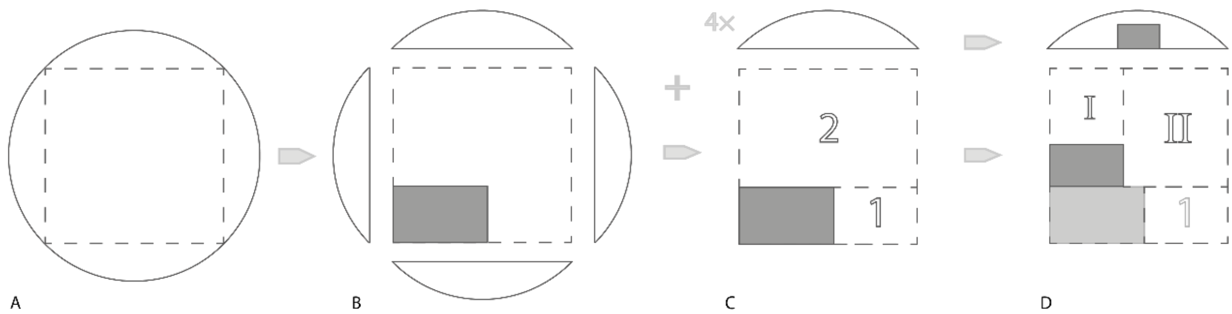


Figure 10: A visual representation of the greedy packing heuristic on a 2D level.

Since the bin packing in this framework is not a 2D but a 3D problem it is important to include vertical logic. For the vertical aspect the same logic applies as on a 2D level. Whenever an element is placed it divides the space into three new boxes; one in the x-axis, one in the y-axis and one in the z-axis. Essentially splitting the empty box into three new smaller boxes for each placement. Figure 11 shows this 3D splitting logic after one element is placed and after six elements.

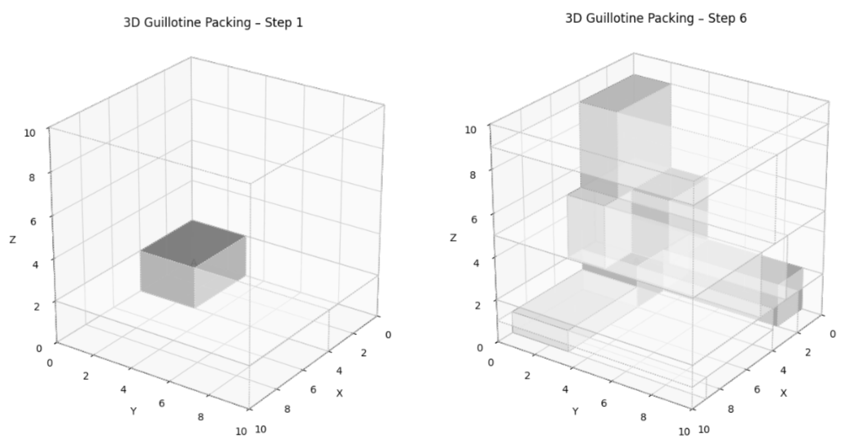


Figure 11: Visual representation of the guillotine bin packing heuristic logic in 3D.

Figure 11 shows small elements in a placement order to visually explain the heuristic logic. The elements in the final results will likely all be tall rectangles packed vertically. In the initial heuristic run the elements are placed in a descending order, placing the tallest element first. For every new element the heuristic first tries to place it in the already opened bins.

When placing an element the ratio with the smallest area is tried first, if this configuration does not fit, the different possible length × width ratio's are tried. Rotations of 90 degrees are allowed.

If the opened bins do not suffice, a new bin is opened. The heuristic runs through the bin order until the first bin that has the minimum required height for the element. In the first run the bins are ordered in a height-ascending manner, meaning that initially the bin with the closest matching height is chosen for the unpacked element. At this stage there is no preference for any specific wood species a bin might have. Since the heuristic only considers the logs in the given order and choosing the first bin that fits, the heuristic can be classified as a greedy heuristic.

In the appendix is pseudocode for the packing and placement logic. Algorithm 1 in the appendix shows that for every element in the permutation, first the list of open bin candidates is cycled through before opening a new bin. In this algorithm the function 'TryPlaceElement' is mentioned. This function is shown in algorithm 2 (appendix); and shows the element placement logic as shown in figure 10 and 11.

3.3.2.2 Genetic Algorithm (GA)

Combining the heuristic approach with metaheuristics would allow for optimization. Metaheuristics are a form of optimization through changing parameters by which the heuristic algorithm solves. In this research a genetic algorithm (GA) will be used. Genetic algorithms are a commonly used metaheuristic. A genetic algorithm will change variables (gene) to create a solution (chromosome). By making multiple possible solutions (a population) the GA scores every solution based on a scoring formula (fitness). Genes from the best solutions according to this formula will be used in new iterations (a generation). Genetic algorithms use a large population size each with different alterations. This allows for many different solutions to be tried out simultaneously.

The variables (genes) that the metaheuristic will change are: the element the order and the bin order. This means that the process is permutation based. Figure 12 shows how different permutations can change; and improve the outcome of the packing problem. The figure shows a simplified stock of 4 bins and 8 elements. If you pack the elements in a height-descending order using the heuristic logic you would get the result **order 1**. The heuristic starts by placing the first element in the order, element A in this figure, and chooses the bin that best fits it's length. For the next elements in the order, B & C, the heuristic tries to place these in the bin that is already open; and succeeds. Element D can no longer fit in the opened bin, and therefore requires to open a bin that fits within the required length range. Element E & F also fit this bin and for element G a new bin is opened. For the last element, H, there are now bins that fit resulting in opening the 4th bin to supply for element H. In **order 2** the G element is shuffled to the 3rd different place in the series, changing the placement permutation. When the elements are placed using the same heuristic logic in this order, only 3 bins are needed for the final placements rather than 4. This improvement can be explained by the fact that the earlier placement of element G allows for the element H to be stacked on top of it later on in the process, creating a synergy between the two elements.

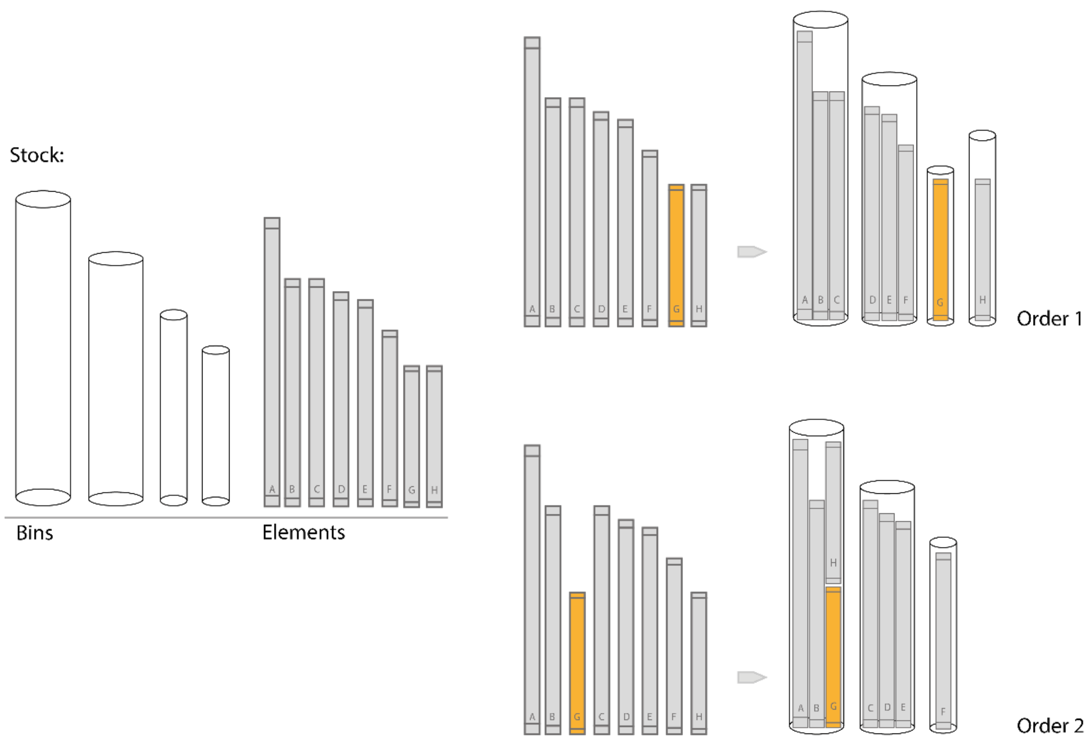


Figure 12: Visual example of a different permutation scoring better using the heuristic rules.

In order to manipulate the permutation the GA will perform order-crossover (OX). Order crossover is a recombination operator where offsprings are created by combining sections of the permutation from two parents. Figure 13 shows a simplification to explain this process. In this figure, the elements that are to be packed are numbered 1 to 270. The heuristic goes through this order and gives the solution. Then, for the first generation the order is randomized. The number of randomized permutations is called the population. In this example the population size is 3. Every possible order in this population is called a chromosome. With these chromosomes the heuristic program runs using the newly created permutations. These are then rated by a fitness-score. For the next generations the best performing chromosomes are combined into offsprings; new possible permutations. In the example of figure 13 the best scoring chromosome has the order 18,62,113,203...221. The second best chromosome is 7,20,43,173...90. For the cross-over operation a chunk of a random length is taken from the best scoring chromosome. The offspring will inherent this chunk directly, in the same relative place as it's parent. So in the case of chromosome 2 in generation 2 it could be said that the chunk of 62,113,203,... is chosen. This leaves the offspring with a gap in front and behind of the sequence: x,62,113,203,...,x,x,x. The empty spots are filled with the genes of the second best chromosome in the order of occurrence, skipping over the duplicate genes. This way it preserves a block from parent 1 and preserves the relative ordering of the remaining genes of parent 2.

In order to explore more diversity two other operations are included in the genetic algorithm. These two functions are 'mutations'. The first is a swap mutation, where random individual genes are swapped from location within the series. The second mutation operator is based on inversion; the order of a randomly sized chuck is inverted. These mutations adds a randomness to some of the solutions and allows the algorithm to explore different options. The order of occurrence for mutations are controlled by a hyperparameter.

Other hyperparameters are tournament size and elite count. Tournament size is what determines the parents for each offspring. If only the best performing two parents are chosen for a whole generation it strongly limits the diversity. For this reason a group of random parents are chosen and compete together within this pool. This allows for some overall lower performing parents to still participate in the generation of offsprings.

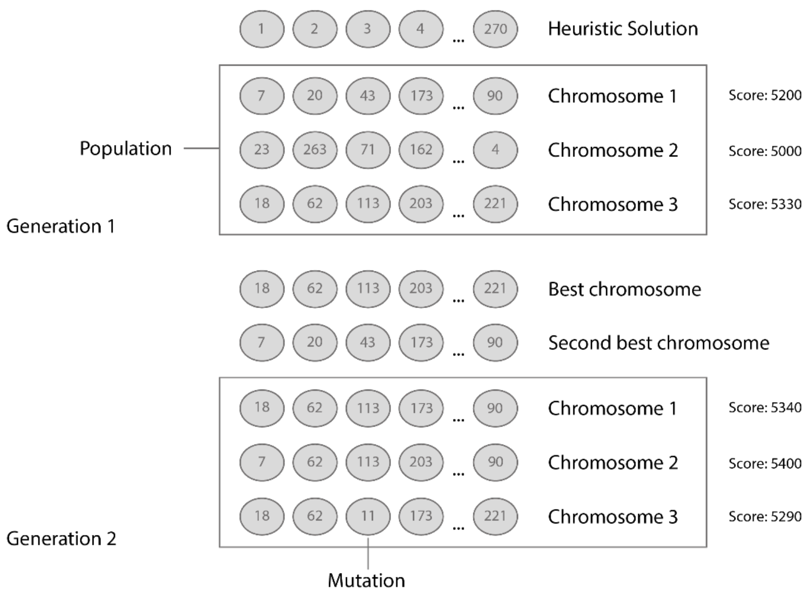


Figure 13: Visual example of the GA with cross-over.

Elite-count is the number of best-scoring solutions that get directly transferred to the next generation as is. This guarantees that the permutation that scores the highest is not lost. Whenever the scoring in the GA stagnates it means that the offsprings are not able to score higher than the elite parent. For an elite count of i.e. 2, this means that for a population size of i.e. 50, there are 2 directly transferred chromosomes and the remaining 48 permutations are generated through cross-over.

The fitness-score for the GA is defined as:

$$score = (100 \times util\_sum) - (10 \times bins\_used)$$

Util_sum = sum of the utilization over all opened bins

Bins_used = the total number of bins.

Like mentioned in chapter 3.3.1, the primary goal is to maximize bin utilization. Utilization is calculated as the total volume of elements in a bin divided by the total volume of the bin. Utilization is calculated as a percentage meaning that the utilization value per bin ranges from 0 to 1. A bin with 50% would be have a util score of 0.5. In the fitness score this is multiplied by a factor of 100, meaning a 50% bin would add 50 score points. In order to penalize the algorithm for opening a new bin a score reduction of 10 points is subtracted per bin. This prevents the algorithm from opening up unnecessary bins while still keeping the penalty low enough to allow for broader exploration.

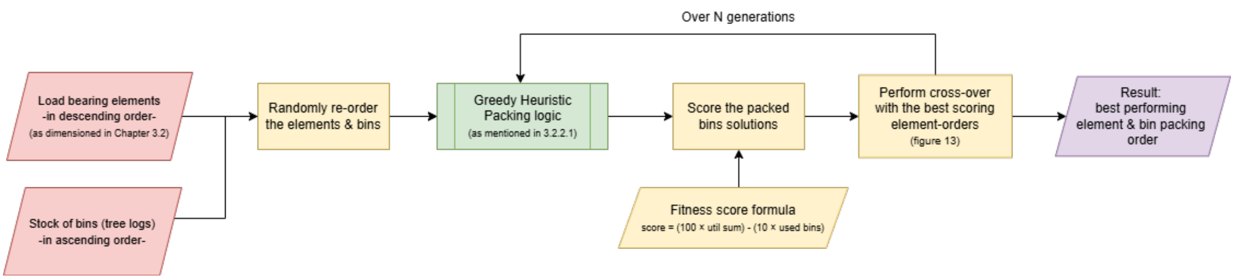


Figure 14: Flowchart simplifying the genetic algorithm approach.

In the appendix is pseudocode for the working of the genetic algorithm as discussed above. It is sectioned under 'Algorithm 3: GeneticAlgorithm'.

GA hyperparameters:

For the experiments in this research the hyperparameters of the GA, unless mentioned otherwise, will be set to:

- pop_size = 50
- generations = 30
- mutation_rate = 0.1
- elite_count = 2
- tournament_size = 4

3.3.2.3 Formula based alternative GA approach

The algorithmic approach as mentioned above does not use an existing Python library for the genetic optimization. This means that the optimization is not benchmarked. In this paragraph two different formula-based approaches will be discussed, that can be coupled with the existing GA Python Library: **PyGAD**. The logic for the packing heuristic will remain the same.

Approach 1, formula to directly generate a series

4-parameters, 5 input variables

The first approach is based around a formula with variables that generate a series. By changing the variables the formula creates a new series, this series is used to move around the items in the permutation.

The formula by which the series is calculated is:

$$s_i = as_{i-1} + bs_{i-2} + c + di$$

In this formula, the four parameters $a, b, c,$and d are the decision variables optimized by the genetic algorithm. s_i is the number at position i in the sequence, s_{i-1} the previous number and s_{i-2} the one before the previous.

The series ($\hat{s}_i$) that follows from the formula for s_i is used to nudge the indices of the original permutation. This can be written as:

$$\text{key}_i = i + \text{strength} \cdot \hat{s}_i$$

The key is the new index for every element in the permutation. i is the original index. The value of the $\hat{s}_i$ series is multiplied by a strength value. The strength value allows the algorithm to explore more diverging broader solutions, with a high value, or generate more local solutions; when the strength value lower. Strength is the 5th decision variable in this GA.

8 parameters, 9 input variables

Similar to the 4-parameter approach, a formula is used by which the indices will be moved around using the key_i . But to explore a more flexible alternative with more input variables, this formula uses 8-parameters. Some of the parameters will have a stronger drift because of their mathematical operator; like the square-root for parameter *e*.

$$s_i = as_{i-1} + bs_{i-2} + (c + di + ei^2) + f\sin(gi + h)$$

This formula based approach enables the PyGAD library to explore the solution space by optimizing 5 or 9 parameters that generate packing orders, including a variable that controls the perturbation strength. A possible caveat with this approach is that it does not directly keep specific element orders in sequence; like how the cross-over does. If there is a very specific element order that works it is harder to keep this niche combination with the formula approach as opposed to the complete sequence sampling seen in cross-over.

Approach 2, formula based on characteristics

For the second formula approach the permutation will be based on 7 variables. Each variable will represent a weight-value for a certain characteristic of an element/bin.

The 7 variables are: wH, wV, wA, uD, uH, uV, uS.

wH, wV & wA are the weights for the loadbearing element characteristics; where:

H = element height
V = element volume
A = element cross section-area

uD, uH, uV and uS are weights for bin characteristics; where:

D = bin diameter
H = bin height
V = bin volume
S = bin slenderness

The characteristic value for each item in the stock will be multiplied by these variable weight-values. This will create a new ranking for every item. The permutation can then be based around these scores, making it so the order goes from highest scoring to lowest. By changing these weights the GA can explore different element & bin permutations, with a direct correlation to their properties. This approach can be seen as an enhanced version of the initial heuristic approach where the elements are sorted on descending length and the bin on ascending height. Instead of basing it only on one metric this version of the genetic algorithm tries to define the best working set of metrics by which this specific stock of items can be ordered.

This approach has more narrow causality than the series based approach, because with this approach small changes of the GA-controlled weights generate only gradual changes in the priority scores. Thereby only limited reordering of elements and bins rather than a completely different permutation. A con of this approach is that the permutations will overall be less divergent and can not easily explore orderings where items with very different properties are next to each other. This could lead to skipping over synergies as shown in figure 12.

The results of the alternative GA approaches are presented under chapter 5.3.8.
The fitness function and hyperparameters remain the same, unless mentioned otherwise.

3.3.3: Implementation

The heuristic and metaheuristic were implemented in Python 3.12 and run from Visual Studio Code.

- **Data handling:** Inputs and outputs were handled in JSON files to allow for easy integration with Grasshopper. Each JSON contained logs (circles) and their corresponding range of elements (rectangles), identified by slot IDs.
- **Visualization:** Solutions are visualized using matplotlib. The solutions are presented as 3D cylindrical bins with the responding elements packed inside. 2D cross-sections at three different levels are created parallel to the 3D model, to allow for easier reading.
- **Progress monitoring:** Iterative algorithms included progress output during the running of the algorithm to track time and to discover possible optimization pitfalls (current score, utilization rate, amount of logs used, elapsed time, ETA).
- **Evaluation metrics:** For each run, utilization percentage, number of logs used, placed vs unplaced items, and runtime were recorded.

Figure 15 shows how the 2D cross-sections are to be interpreted.

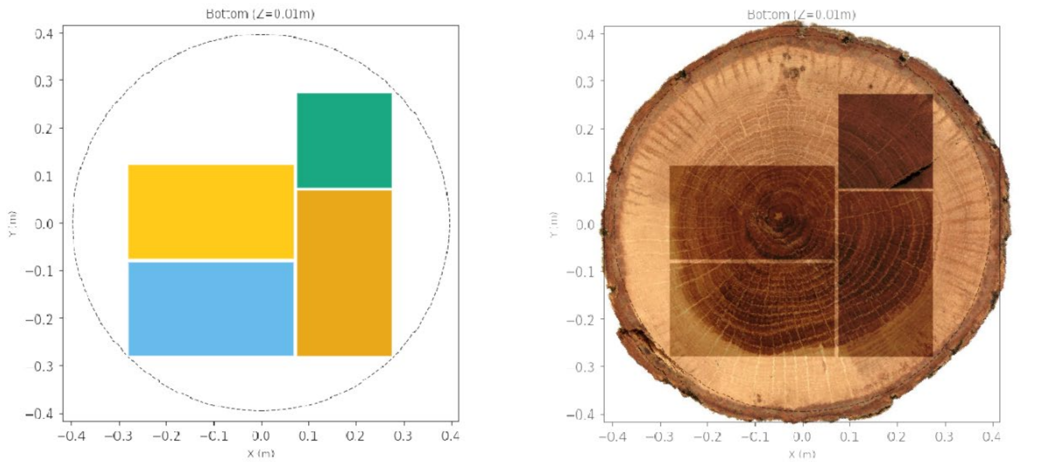


Figure 15: Visual interpretation of cross-section, by overlaying the cutting pattern on an image of an oak.

/C3.4: Parameter exploration

With minor adjustments to the primary framework or databases different circumstances and goals can be explored. In this section five different situations will be experimented with.

The parameters that will be explored are:

- 3.4.1 Weighted timber species
- 3.4.2 Exploring a more extreme reference building
- 3.4.3 Inclusion of biological behavior
- 3.4.4 Inclusion of non-loadbearing elements
- 3.4.5 Rounded dimensioning for load-bearing elements
- 3.4.6 Exploring different GA Hyperparameters & fitness-score
- 3.4.7 Preservation of tallest trees

3.4.1 Weighted timber species

The only objective of the genetic optimizer with the current scoring is to strive for the highest utilization with a small penalty for opening new bins. However these objectives can be changed in order to pursue different results. One of these objectives could be to prioritize certain tree species over others. This could prove useful in several different situations. One of which could be whenever there is a large supply of one specific tree specie. I could be preferable to design using primarily these wood types, to allow for the best use of the available stock. By changing the formula by which the GA calculates it's optimization score a species preference could be integrated within the current framework. The GA formula is changed to:

$$\begin{aligned} \text{score} = {} & (100 \times \text{util\_sum}) - (10 \times \text{bins\_used}) \\ & + (\textbf{weight\_GA\_species} \times \textbf{species\_score}) \end{aligned}$$

The new addition to this formula is the weight_GA_species × species_score. Weight_species is a parameter value that is set by the user. It is meant to represent the overall importance of weighing species; similar to the values 100 and 10.

The other parameter is species_pref_score. The species_pref_score is a formula on itself. The species preference score can be best explained in two parts. The first part is:

$$\left(\text{value}(\text{species}_\text{b}) \cdot \text{utilization}(\text{b})\right)$$

Value(species) is a parameter that is beforehand. It is the theoretical value that the user assigns to each tree species. The value is on a scale of 0 to 1. Zero being of no value and one being the most valuable. This value is multiplied by the utilization of a bin of that species.

For the second part; the sum of those values is taken and divided by the overall number of bins. This avoids that the algorithm could score points by simple opening up bins of priority wood without utilizing them properly.

$$\text{species\_pref\_score} = \frac{1}{\text{N}_\text{used}} \sum_{\text{b} \in \text{used bins}} \left(\text{value}(\text{species}_\text{b}) \cdot \text{utilization}(\text{b})\right)$$

The GA hyperparameters remain be the same.

3.4.2 Exploring a more extreme reference building

This section will explore whether the Rotterdam city wood stock could supply for a more extreme structure. The structure that will be used for this experiment is a simplified version of the *Floating Office Rotterdam* (FOR) (see figure 16 & 17). FOR is a design by PowerHouse Company. The structure is 65.88 meters long and 18.25 meters wide, with 3 stories totaling a 11.5 meters height. The original building has a slanted roof with long engineered timber beams spanning from the center to the side. The choice has been made to simplify this roof structure into a flat roof allowing for the grid like column-beam frame to continue.

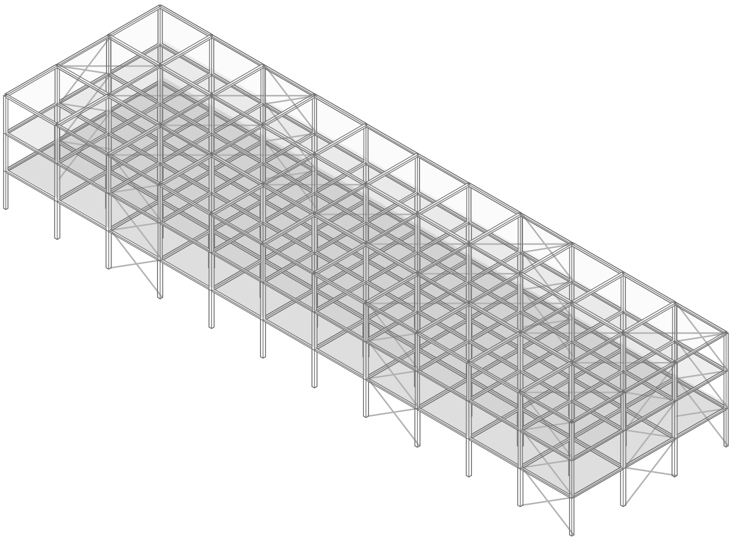


Figure 16**:** Impression of a simplified version of the *Floating Office Rotterdam* structure.

Figure 17**:** *Floating Office Rotterdam (FOR)*
Photograph of Floating Office Rotterdam (FOR) by Powerhouse Company. Photo by Marcel IJzerman, retrieved from https://shorturl.at/bQZn2 ***2**

3.3.3 Inclusion of biological behavior

In the current heuristic approach the log is assumed to be a homogeneous cylinder. However with real wood the cross-section of a tree can be subdivided into three segments as mentioned in chapter one. The outer layer of a tree; the sapwood, is of considerable less structural quality. Also the center of a trunk; the pith, is unfavorable. Including the pith in your cuts can result in strong warping of your timber. In practice there are two main approaches in regard to the pith. Either leaving it out, or completely boxing it in. By boxing in the pith in a rectangle of relative big size, the idea is that since warping occurs on all sides evenly it cancels each other out. As for the sapwood, in most sawmills the sapwood is included in the cuts. For the sake of this parameter exploration however the sapwood will be avoided. The pith will follow the logic of either avoiding it or boxing it in.

In order to explore these parameters some changes had to be to the heuristic. The heuristic will assume that the diameter of the trunk is equal to the diameter of the heartwood. Inside the heartwood the heuristic will treat the center as if there was already an element placed that spans to the top of the trunk. This way the heuristic will avoid placing elements overlapping with the pith. If the heuristic is trying to place an overlapping element however it will run a check on whether that element is a rectangle and of considerable size. If so, the heuristic will choose to 'box in the pith' instead of avoiding it.

3.4.4 Inclusion of non-loadbearing elements

The framework for this research primarily focuses on loadbearing elements. The bin-packing algorithm however should work for any type of rectangle. By adding new elements to the dataset the algorithm could be used to include non-loadbearing elements, like the wooden shades that are on both sides of the *Muiden* building (see figure 18). The shades used for the model below are 3x10cm.

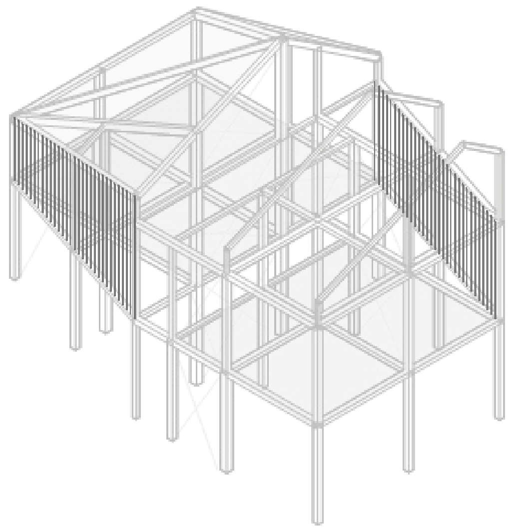


Figure 18: An impression of the loadbearing structure of *WoonHuis Muiden* including wooden shades at the sides of the building.

3.4.5 Rounded load-bearing element dimensions

The dimensions for the elements are calculated as discussed in chapter 3.2. The order of magnitude of these are calculations give dimensions to the mm. This makes it so every element is narrowly tailormade. the quality In order to create a certain level of uniformity the dimensions and rounded to 5 centimeter. This creates a certain standardization and could allow for easier reuse after the lifespan of the building.

3.4.6 Exploring different GA Hyperparameters & fitness-score

In chapter 3.3.2.3 the hyperparameters and fitness-score are set. In order to justify the choice for these fitness-score and hyperparameters, there will be experimentation with a larger population and different fitness-score. For the results in 5.3.6 the population size will be set to 500, and there will be experimentation with a new fitness score of:

$$\text{score} = (100 \times \text{util\_sum}) - (40 \times \text{bins\_used})$$

The expectation is that a larger bin penalty will lead to less bins. In order to achieve this the algorithm could choose to stack elements into tall bins, where it could otherwise prefer to fit the elements in a smaller bin and not stack them on top of each other. This could mean that the new fitness-score would lead to less bins, but increased usage of tall trees.

3.4.7 Preserve tallest trees

In order to counteract the expected increase of tall tree usage, when enforcing a -40 bin penalty, there will be experiments with a change to the heuristic approach. The goal of the new heuristic approach is to preserve tall trees were possible. The way this will be achieved is by changing the logic by which a bin is chosen. In the heuristic as proposed above the bin permutations is the only leading factor for bin choice; given that the element must fit in the bin. The reasoning behind this approach is so that shuffling the bin order would give the heuristic new solutions. If the heuristic always chooses the best fitting bin for an element, shuffling the permutation would not matter, thereby reducing the exploration space of the GA. The new heuristic approach that will be experimented with in this section tackles this problem by setting a height-difference limit from which the bin choice can deviate. The height-difference parameter will be set to 20cm, meaning that a bin is allowed to be 20 centimeters taller than the element that is to be placed. This allows for the shuffling of bins, while still preserving tall trees. If no bin exists within the stock that satisfied the 20cm barrier, the first best bin in the permutation that fits is taken.


04
Case study

Chapter 4: Case Study: Rotterdam City Wood

/C4.1: Introduction on Rotterdam's City Wood

To test the framework as proposed in the methodology a dataset for urban trees is required. This thesis will look into the urban trees of Rotterdam. In Rotterdam, at the time of March 2025 a total of 605 trees have been requested or approved to be taken down. Figure 19 shows an impression of these trees. In this figure the green circles represent trees that have been approved to be felled, where trees with a blue circles have their request pending. For the sake of this research however the trees with their request pending are treated the same in the database as the approved ones.

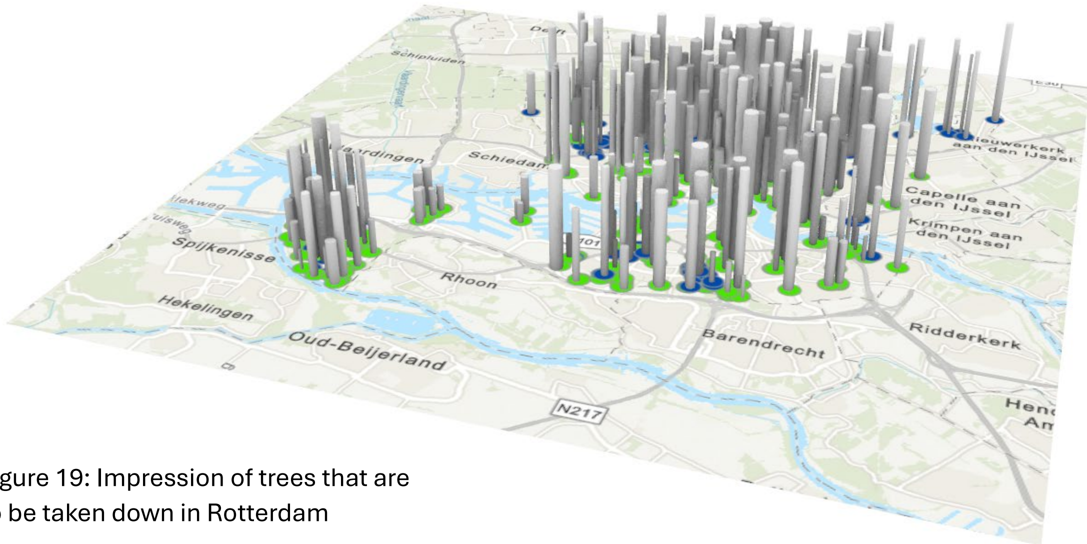


Figure 19: Impression of trees that are to be taken down in Rotterdam

For each tree it is catalogued what species it is and where it is located. The dimensional information of these trees are not consistently readily available. In order to create a 3D database dimensional estimations need to be made. Based on the species of each tree, a range for length and diameter is defined. For each tree, a value within this range is then sampled according to a normal (bell-shaped) distribution, so that values near the mean are more likely than values near the extremes.

According to RotterdamsKrom there are over 700 different species of trees in the city of Rotterdam. Based on the wood flow in 2023 and 2024 they have made a top ten most relevant wood species in Rotterdam. These ten wood species are: Poplar, Plane, Oak, Willow, Elm, Robinia, Beech, Ash, Maple, Chestnut. All ten species are hardwood. Of these ten species, eight will be taken into consideration for this research. Willow & Plane are excluded from this list, because their structural performance was not readily available and these species rarely see structural use in architecture.

Table 11 in the appendix shows the amount of species and their assigned dimensions that will be used as the dataset for the result section.

/C4.2: Wood data description

In this paragraph each species will be quickly discussed regarding their properties and potentials. All 8 species will be addressed in order of highest density to lowest.

.Robinia

Lat. Robinia

The wood species with the highest average density is Robinia.
They have a maximum height of 25-30m and an average trunk diameter of 0,6-0,9m. Robinia trees tend to branch off low above ground making it so that have a relatively short branchless trunk. Robinia wood should be dried slowly because it has a strong tendency to warp. Robinia is the most durable tree that grow in the Netherlands and is sometimes comparable to exotic hardwood. (*Houtinfo - Houtsoorten*, 2019)

.Density	720	[kg/m3]
.E-modulus	16.000	[MPa]
.Hardness	71 (//) – 48 (⊥)	[N/mm2]
.Bending strength	126	[MPa]
.Compression strength	70	[MPa]
.Shear strength	17	[MPa]

(*Tabel Loofhout*, n.d.)

Robinia trees have an average *form stability*. They are known to be used in load bearing structures and engineered timber. Furthermore have application as facade finishing, interior-, exterior- and roof carpentry.

.Oak

Lat. Quercus

The wood species with the second highest average density is Oak.
They have a maximum height of 18-30m and an average trunk diameter of 1,2-1,8m. Oak trees can have a straight branchless trunk up to 15 meter, and can live up to 400 years old. Oak wood should be dried slowly because it tends to warp. (*Houtinfo - Houtsoorten*, 2019)

.Density	710	[kg/m3]
.E-modulus	12.500	[MPa]
.Hardness	57 (//) – 32 (⊥)	[N/mm2]
.Bending strength	97	[MPa]
.Compression strength	50	[MPa]
.Shear strength	10	[MPa]

(*Tabel Loofhout*, n.d.)

Oak trees have a low to average *stability*. They are known to be used in load bearing structures and can be used in engineered timber. (Llana et al., 2022) Furthermore oak has application as facade finishing, interior-, exterior- and roof carpentry, veneers and furniture. (*Houtinfo - Houtsoorten*, 2019)

Figure 20: Example of Robinia tree

Figure 21: Example of Oak tree

.Ash

Lat. Fraxinus

The third wood species in this list is Ash.
Ash trees have an average height of 20-30m with a maximum up to 40m. The average trunk diameter is 0,4-0,9m. Ash trees can have a straight branchless trunk up to 20 meter. Ash wood can be dried reasonably quick and already dries in the outdoor air.

.Density	700	[kg/m3]
.E-modulus	12.500	[MPa]
.Hardness	64 (//) – 34 (⊥)	[N/mm2]
.Bending strength	110	[MPa]
.Compression strength	54	[MPa]
.Shear strength	12	[MPa]

(*Tabel Loofhout*, n.d.)
Ash wood has an average *form stability*. They are known to be used in load bearing structures and have seen some use in engineered timber. Furthermore ash has application as interior and exterior carpentry. (*Houtinfo - Houtsoorten*, 2019)

.Beech

Lat. Fagus

Beech trees have a estimated height of 30m with a whopping maximum up to 45 meter. The average trunk diameter is 1-1,5m. Beech trees can have a straight branchless trunk from around 9 to 15 meter. Beech wood dries slow and can be prone to crack and warp. Despite this beech is one of the most used and known hardwood species in the European industry.

.Density	700	[kg/m3]
.E-modulus	13.500	[MPa]
.Hardness	71 (//) – 28 (⊥)	[N/mm2]
.Bending strength	113	[MPa]
.Compression strength	54	[MPa]
.Shear strength	10	[MPa]

(*Tabel Loofhout*, n.d.)
Beech wood has an low *form stability*. Beech is commonly seen in engineered timber, also n combination with Poplar wood. (Hematabadi et al., 2021) Furthermore beech has application as interior carpentry, veneers and furniture. (*Houtinfo - Houtsoorten*, 2019)

Figure 22: Example of Ash tree

Figure 23: Example of Beech tree

.Elm
Lat. Ulmus

Elm trees have an average height of 35-40m. The average trunk diameter is 0,9-1,4m, but can be as big as 2,5 meters. Elm trees have a straight branchless trunk around 10 to 18 meters. Ash wood can be dried reasonably quick but can not be dried to fast or it will crack internally.

.Density	640	[kg/m3]
.E-modulus	10.800	[MPa]
.Hardness	60 (//) – 37 (⊥)	[N/mm2]
.Bending strength	*88*	[MPa]
.Compression strength	50	[MPa]
.Shear strength	*6,8*	[MPa]

(*Tabel Loofhout*, n.d.)

Elm wood has a low *form stability*. Timber elm elements can be used for small, not to heavy structural purposes. Elm is also used for furniture and in interior carpentry. (*Houtinfo - Houtsoorten*, 2019)

.Maple
Lat. Acer

Maple trees have a estimated height of 20 to 25m. The average trunk diameter is 1,5m. Maple trees can have a straight branchless trunk from around 15 meter. Some species of Maple can reach up to 40 meter in height with a branchless trunk of 20m. Maplewood dries slow but can dry really good to the outdoor air.

.Density	630	[kg/m3]
.E-modulus	12.500	
.Hardness	*54* (//) – 30 (⊥)	[N/mm2]
.Bending strength	113	[MPa]
.Compression strength	58	[MPa]
.Shear strength	14	[MPa]

(*Tabel Loofhout*, n.d.)

Maple wood has an low *form stability*. Maple doesn't commonly see use as load bearing or engineered timber. It is more known for applications in furniture, veneers and flooring. (*Houtinfo - Houtsoorten*, 2019)

Figure 24: Example of Elm tree

Figure 25: Example of Maple tree

.Chestnut

Lat. Castanea

The average chestnut tree height is 15-25. With a trunk diameter of 0,6 to 1 meter. Chestnut trees tend to branch off 6 to 15 meters above ground. Chestnut wood should be dried slowly and is known to warp and crack both externally as internally when drying. Chestnut is easy to work with but can cause irritations to the skin.

.Density	620	[kg/m3]
.E-modulus	9.000	[MPa]
.Hardness	34 (//) – 19 (⊥)	[N/mm2]
.Bending strength	*71*	[MPa]
.Compression strength	46	[MPa]
.Shear strength	9	[MPa]

(*Tabel Loofhout*, n.d.)

Chestnut has an average *form stability*. They can to be used in small load bearing structures and as engineered laminated. Furthermore chestnut has indoor and outdoor applications as facade finishing, interior-, exterior- and roof carpentry. (*Houtinfo - Houtsoorten*, 2019)

.Poplar

Lat. Populus

The wood specie with the lowest of the ten densities is Poplar wood. Poplar trees can grow fast and are sometimes ready for harvest within 25 years. The height of a poplar tree depends on its kind within the species but range from 18 to 35 meter. A trunk diameter of 0,9 to 1,2 meter. Poplar trees branch off relatively low above the ground. Poplar dries moderately quick but can be subject to deformations because of reaction wood (see chapter 1). (*Houtinfo - Houtsoorten*, 2019)

.Density	400	[kg/m3]
.E-modulus	9.000	[MPa]
.Hardness	29 (//) – 12 (⊥)	[N/mm2]
.Bending strength	*65*	[MPa]
.Compression strength	33	[MPa]
.Shear strength	6	[MPa]

(*Tabel Loofhout*, n.d.)

Poplar has an average *form stability*. Poplar wood is used for indoor carpentry, but is also known to be hydrothermally modified to be used for outdoor purposes such as façade cladding. Poplar timber is not seen in load bearing structures but has a big share in engineered timber. Also together as a hybrid with Beech in laminated timber. (Hematabadi et al., 2021) (*Houtinfo - Houtsoorten*, 2019)

Figure 26: Example of Chestnut tree

Figure 27: Example of Poplar tree

Table 1 shows an overview of the technical performance for each wood species, as mentioned above.

Name	Density [kg/m3]:	Modulus of elasticity [MPa]:	Hardness [N/mm2]:	S. Bending [MPa]:	S. Compression [MPa]:	S. Shear [MPa]:	Form stability	Shrinkage (90-60%) [%]:	Shrinkage (60-30%) [%]
Robinia	720	16000	71 (//) – 48 (⊥)	126	70	17	average	$\varnothing_r$: 1,2 $\varnothing_t$: 1,7	$\varnothing_r$: 0,8 $\varnothing_t$: 0,9
Oak	710	12 500	57 (//) – 32 (⊥)	97	50	10	low-average	$\varnothing_r$: 1,2 $\varnothing_t$: 2,1	$\varnothing_r$: 0,8 $\varnothing_t$: 1,2
Ash	700	12 500	64 (//) – 34 (⊥)	110	54	12	average	$\varnothing_r$: 0,6 $\varnothing_t$: 1,3	$\varnothing_r$: 0,7 $\varnothing_t$ 1,2
Beech	700	13 500	71 (//) – 28 (⊥)	113	54	10	low	$\varnothing_r$: 1,2 $\varnothing_t$: 2,5	$\varnothing_r$: 0,9 $\varnothing_t$: 1,5
Elm	640	10800	60 (//) – 37 (⊥)	88	50	6,8	low	$\varnothing_r$: 1,6 $\varnothing_t$: 2,8	$\varnothing_r$: 1,5 $\varnothing_t$: 1,6
Maple	630	12 500	54 (//) – 30 (⊥)	113	58	14	low	$\varnothing_r$: 1,2 $\varnothing_t$: 2,0	$\varnothing_r$: 0,8 $\varnothing_t$: 1,4
Chestnut	620	9000	34 (//) – 19 (⊥)	71	46	9	average	$\varnothing_r$: 0,7 $\varnothing_t$: 1,3	$\varnothing_r$: 0,6 $\varnothing_t$: 0,9
Poplar	400	9000	29 (//) – 12 (⊥)	65	33	6	average	$\varnothing_r$: 0,7 $\varnothing_t$: 1,9	$\varnothing_r$: 0,6 $\varnothing_t$: 1,4

Table 1: Overview of the technical values for each tree species.

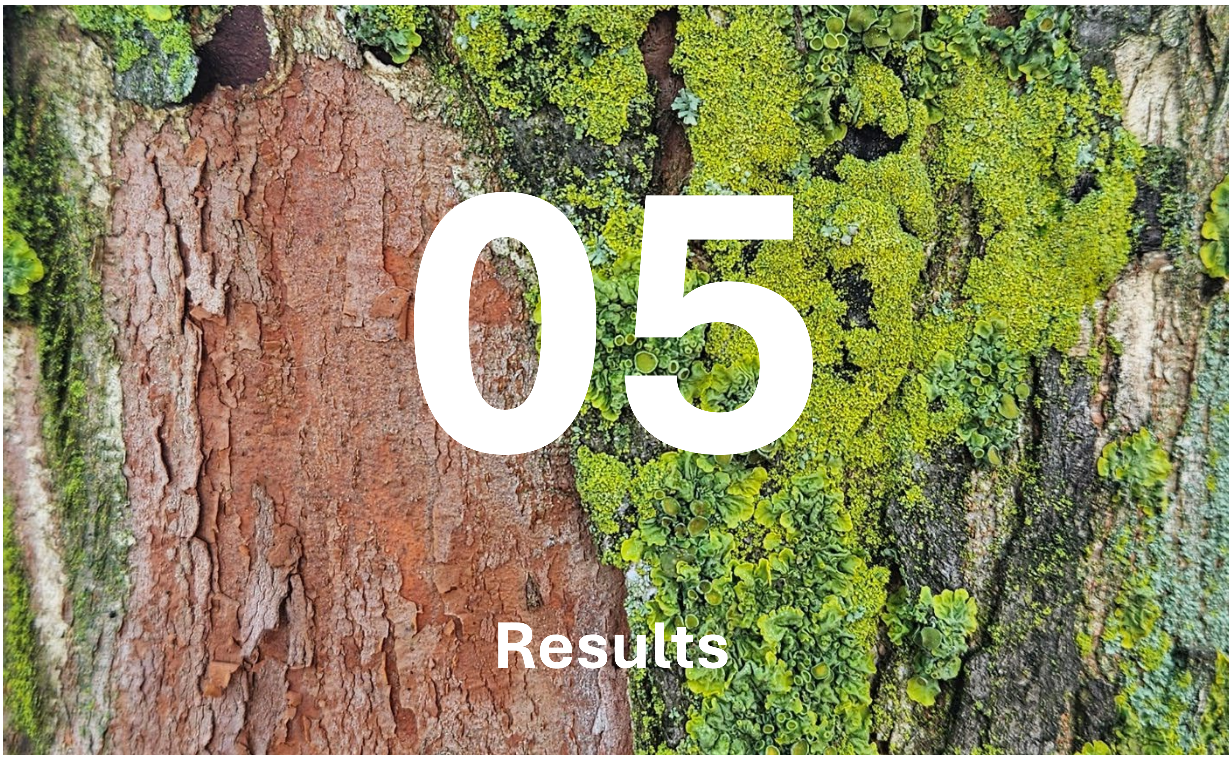
In table 2 the estimated / common height for the eight wood types are summarized. For most tree types their height lies around 20 to 30 meters. The effective branchless size of the trunk fluctuates, and lies more between 5 to 15 meter. Although this table gives an insight into the expectations of tree sizes, in the case of Rotterdam's urban forestry these numbers can vary a lot since they are often grown in different circumstances then the trees in this table. For this reason these numbers are significantly reduced for the sake of this research. For most trees a range between 5 to 10 meters of branchless trunk will be the distribution scale, as can be seen in table 3.

Specie	Height [m]	Diameter [m]	Branchless trunk [m]
Robinia	25 - 30	0,6 - 0,9	Low
Oak	18 - 30	1,2 - 1,8	15
Ash	20 - 30	0,4 - 0,9	15 - 20
Beech	30 - 45	1 - 1,5	9 - 15
Elm	35 - 40	0,9 - 1,4	10 - 18
Maple	20 - 25	1,5	15
Chestnut	15 - 25	0,6 - 1	6 - 15
Poplar	18 - 35	0,9 - 1,2	Low

Table 2: Overview of diameter and branchless trunk size as proposed by Houtinfo - Houtsoorten, 2019

Proposed range for branchless trunk size	
Species	Branchless trunk [m]
1. Robini	2 To 5
2. Quercus	5.8 To 9.9
3. Fraxinus	7 To 11.3
4. Fagus	5.1 To 9.6
5. Ulmus	5.4 To 11.6
6. Acer	6.2 To 9.9
7. Castanea	4.6 To 9
8. Populus	2 To 5

Table 3: The range of proposed branchless trunk sizes for the normal distribution for each wood species.



05

Results

Chapter 5: Results

This chapter presents the results from experimenting with the proposed computational framework to the case study of Rotterdam's urban timber stock. The outcomes are presented according to the main steps of the methodology: element dimensioning, cutting pattern generation through bin packing, and parametric exploration. Each section reports the numerical results and illustrates the implications for stock utilization.

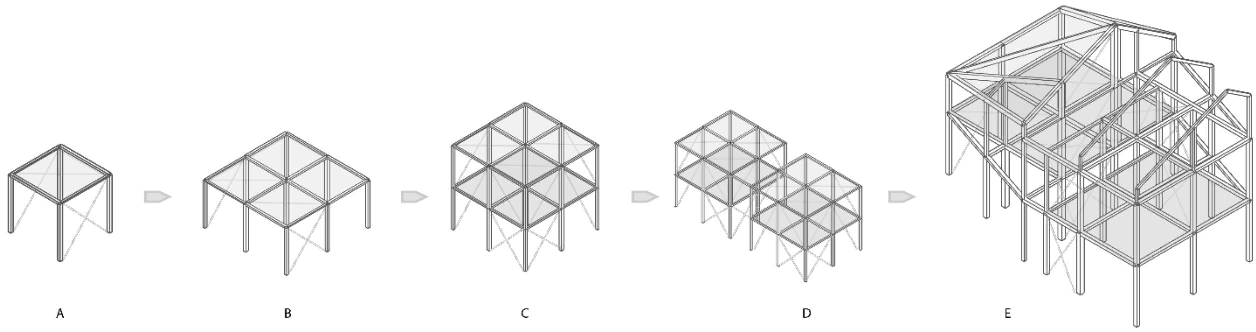


Figure 28: Levels of complexity with intermediate steps from a 4x4x4 frame to an impression of 'Woonhuis Muiden'.

For the primary experiments an impression of the loadbearing structure of the 'Woonhuis Muiden' building was used. *Woonhuis Muiden* is a building designed by MokeArchitects using engineered timber (Fig. 29). This result section will explore whether it could have been possible to realize this building using timber from the case study of Rotterdam. In order to show that the program works at different levels of complexity four steps have been taken before computing *Woonhuis Muiden*. Figure 28 starts with a simple timber frame of 4 by 4 by 4 meters. Timber frame **B** repeats this 4x4 frame, creating a grid with 4 modules. Level **C** is meant to show that the program can handle multiple building layers. At step **D** the frame of its predecessor is duplicated and placed next to one another. These timber frames behave separately from each other essentially representing two individual structures. Level **E** is an impression of the loadbearing structure of the aforementioned *Woonhuis Muiden*, which operates similar to the double structures of level **D.**

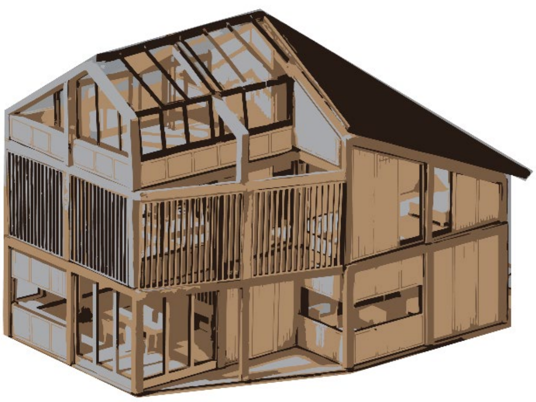


Figure 29: Model design of the 'Woonhuis Muiden' by MokeArchitects, https://www.mokearchitecten.nl/portfolio/woonhuis-muiden

/C5.1: Results element dimensioning

The formula for Euler Buckling is governing for columns and for beams that is the bending moment. The a_{eff} for a charring layer is 37mm. The fire resistance dimension requirements are bigger in size when compared to the ULS dimensions. Making fire-safety the main determinant. On average, the required element sizes increase by approximately 31% when designed for fire resistance (charring layers) than for the ultimate limit state.

In table 4 the average width x height for each element is laid out for every wood type. Robinia; the strongest option, gives an average of 8.8% lower in size when compared to Poplar; the structurally lowest performing wood type. Table 4 shows the results of the 'Woonhuis Muiden' reference; the average dimensions for the other timber frames are in the Appendix.

	Average Width x Height [cm]	Average Width x Height [cm]
Wood type	1:1 Column	1:3 Beam
Robinia	16.3x16.3	17.4x12.0
Oak	16.9x16.9	18.7x12.4
Ash	16.9x16.9	18.0x12.2
Beech	16.7x16.7	17.9x12.1
Elm	17.3x17.3	19.1x12.5
Maple	16.9x16.9	17.9x12.1
Chestnut	17.7x17.7	20.3x12.9
Poplar	17.7x17.7	20.8x13.1

Table 4: Average Width x Height per tree species for 1:1 Column's and 1:3 Beams in the *'Woonhuis Muiden'* inspired structure.

In table 5 the results are shown for different ratio's in a range of 1 to 5 for element made from Poplar trees. Results show that the required average area of columns increases strongly by increases in the aspect ratio. Also the maximum value for the required height exceeds 50cm from a ratio of 1:3 onwards. This is unrealistic in practical scenario's making these results unfavorable. For that reason, and to limit sample size, only column elements from a 1:1 to 1:2 ratio will be taken into account in the bin packing algorithms.

For beams the cross-sectional area decreases as the ratio increases, as could be expected from its formula. Across all ratios the beam elements stay within a reasonable size for practical use. In this study, aspect ratios of 1:1, 1:3, and 1:5 are used to generate cutting patterns. Although the 1:5 configuration appears most promising due to its lower area, the 1:1 and 1:3 variants are also included to explore potential synergies from their geometric diversity.

Ratio w:h	min. Value (c	max. Value (cm)	Average (cm)	Average Area (cm2)
Column 1:1	18.1x18.1	29.3x29.3	22.6x22.6	521
Column 1:2	16.4x25.5	25.8x44.3	20.2x33.0	680
Column 1:3	15.5x31.8	24.1x57.5	19.0x42.2	815
Column 1:4	15.0x37.8	22.9x69.5	18.2x50.5	935
Column 1:5	14.6x43.2	22.0x80.8	18.0x58.4	1045
Beam 1:1	17.7x21.4	21.4x23.9	20.1x23.8	561
Beam 1:2	16.2x21.4	20.1x29.2	17.7x24.4	502
Beam 1:3	14.2x24.0	17.1x32.9	15.3x27.4	484
Beam 1:4	13.0x26.0	15.4x35.8	13.9x29.8	477
Beam 1:5	12.2x27.7	14.3x38.3	13.0x31.7	475

Table 5: Column & beam sizes for Poplar trees as determined by fire-safe Euler Buckling & Bending-moment

Upon closer inspection it could look confusing as to why the values of Width x Length do not always align with the ratio. This can be explained by the charring layer being a fixed number, added to the sides of the original ratio-conform rectangles.

Having established the required timber dimensions, the next step is to allocate these elements within the Rotterdam city wood stock.

/C5.2: Cutting pattern generation

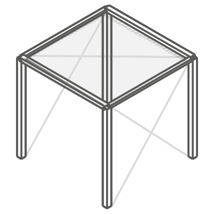
5.2.2 3D Bin packing algorithm

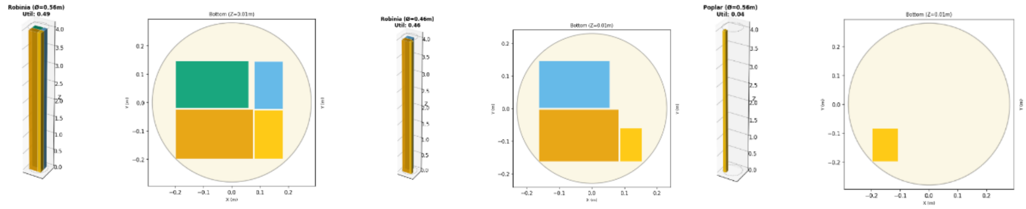
5.2.2.1 Greedy Heuristic: initial run

Before going into optimization it is important to have an efficient heuristic. The heuristic forms the foundation for the optimization. The heuristic as proposed in the methodology is a greedy heuristic. To recap the steps the heuristics takes:

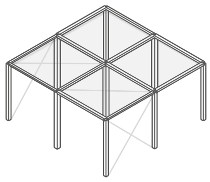
- The first bin that matches the height of an element is opened
- The next element in the list is tried to be placed inside the same bin, using guillotine logic
- Repeat until the element does not fit the opened bins
- Open a new fitting bin from the bin order and fit the unplaced element
- Repeat until all elements are placed

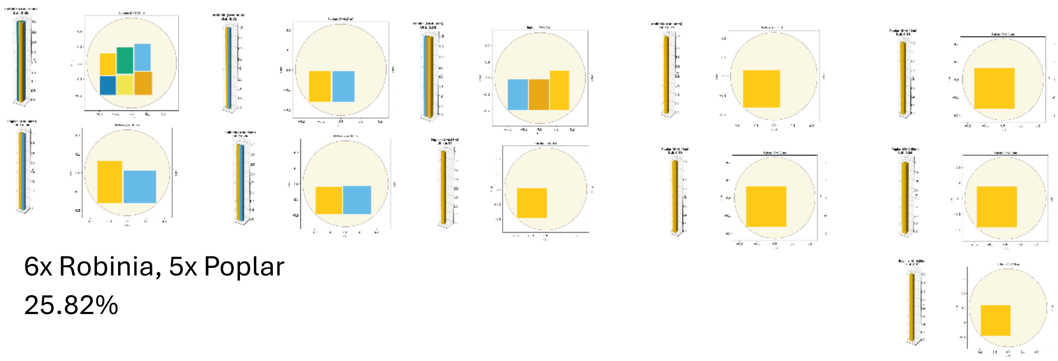
Figure 30, on the next page, shows the results of the initial heuristic, with an ascending bin order and descending element order. For each complexity level as mentioned in figure 30 the heuristic has generated cutting patterns. The visuals however do not fit the page nicely so for structures that use 10+ trees the cutting patterns are in the appendix.



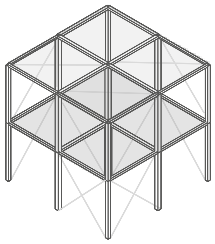


3 trees: 2x Robinia, 1x Poplar
Average utilization: 33.05%

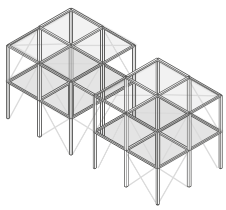




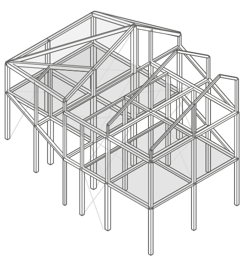
11 trees: 6x Robinia, 5x Poplar
Average utilization: 25.82%



See appendix for cross sections.
20 trees: 10x Robinia, 10x Poplar
Average utilization: 29.10%



See appendix for cross sections.
41 trees: 19x Robinia, 18x Poplar, 1x Chestnut, 2x Elm, 1x Maple
Average utilization: 26.72%



See appendix for cross sections.
39 trees: 16x Robinia, 11x Poplar, 7x Ash, 5x Maple
Average utilization: 29.77%

Figure 30: Overview of the cutting patterns and placement results of the greedy best-fit heuristic.

5.2.2.2 Metaheuristic: Genetic Algorithm

After the initial placement of the heuristic, the genetic algorithm scores that placement according to:

$$score = (100 \times util\_sum) - (10 \times bins\_used)$$

The objective of the genetic algorithm is to improve this score by changing parameters like the order in which the elements are placed & which bins they are assigned to. In figure 31 the progression of the GA is shown for the *Muiden* building. In the appendix the progression graph for the other structures are shown.

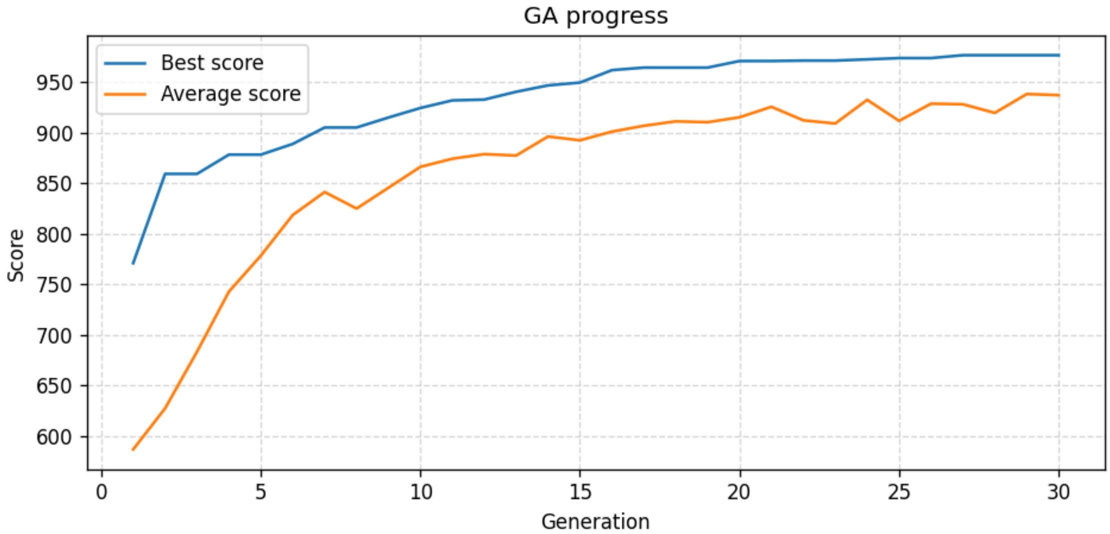


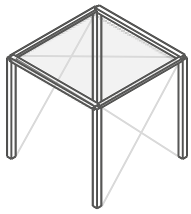
Figure 31: Genetic Algorithm progression graph of the bin-packing optimization for the *Muiden* structure.

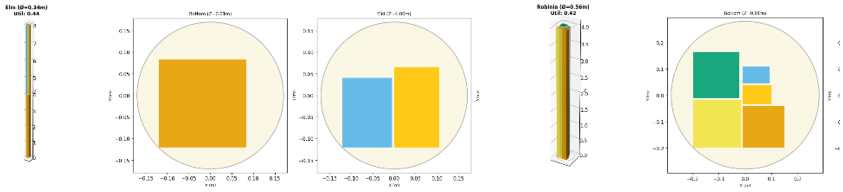
Table 6 is a summary of the gain made by the GA for each structure.

Heuristic VS Metaheuristic						
	Heuristic		MetaHeuristic		Difference	
Structure	Trees #	Utilization [%]	Trees #	Utilization [%]	Trees #	+ Utilization
4x4	3	33.05	2	42.76	1	9,71
8x8	11	25.82	5	47.70	6	21,88
8x8x2	20	29.10	14	44.48	6	15,38
2x 8x8x2	41	26.72	28	43.49	13	16,77
Muiden	39	29.77	34	38.72	5	8,95

Table 6: Number of trees and overall utilization for each structure, Heuristic vs Metaheurstic.

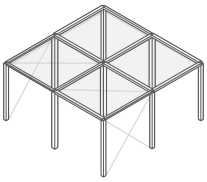
Figure 32, on the next page, shows the results of the metaheuristic. For each complexity level the metaheuristic has generated cutting patterns. The visuals however do not fit the page nicely so for structures that use 10+ trees the cutting patterns are in the appendix.

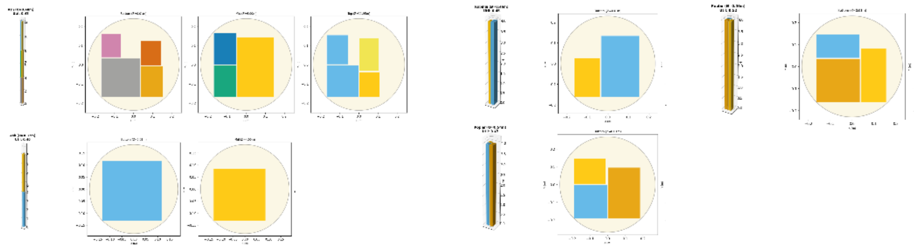




2 trees: 1x Elm, 1x Robinia

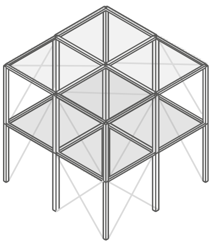
Average utilization: 42.76%





5 trees: 2x Ash, 2x Poplar, 1x Robinia

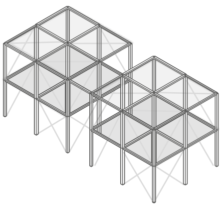
Average utilization: 47.70%



See appendix for cross sections.

14 trees: 7x Robinia, 7x Poplar

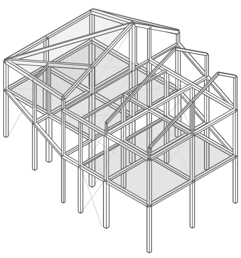
Average utilization: 44.48%



See appendix for cross sections.

28 trees: 13x Robinia, 11x Poplar, 4 Elm

Average utilization: 43.49%



See appendix for cross sections.

34 trees: 11x Robinia, 8x Maple, 7x Ash, 6x Poplar, 2x Elm

Average utilization: 38.72%

Figure 32: Overview of the cutting patterns and placement results after 30 generations of optimisation with a GA.

/C5.3: Results parameter exploration

Since the heuristic- and metaheuristic-approach have proved useful this section will explore the effect of changing different parameters in regard to the framework. The results of these explorations will be based on the element dataset inspired by the *Woonhhuis Muiden* structure, unless mentioned otherwise.

The parameters that will be explored are:

- 5.3.1 Weighted timber species
- 5.3.2 Exploring a more extreme reference building
- 5.3.3 Inclusion of biological behavior
- 5.3.4 Inclusion of non-loadbearing elements
- 5.3.5 Rounded load-bearing element dimensions
- 5.3.6 Exploring different GA Hyperparameter & fitness-score
- 5.3.7 Preservation of tallest trees
- 5.3.8 Alternative GA approaches, PyGAD based

5.3.1 Weighted timber species

Results weighted timber species			value_species
Required logs		Total: 47	Robinia: 0.01
	23x Poplar		Oak: 0.10
	15x Maple		Ash: 0.20
	4x Robinia		Beech: 0.30
	3x Chestnut		Elm: 0.50
	2x Oak		Maple: 0.55
			Chestnut: 0.70
Average utilisation		34.83%	Poplar: 0.80

Table 8: The distribution and utilization of logs, with the objective to prioritize lower performing timber species (value).

Table 8 shows the results of a test-run that prioritizes timber species that perform lower on structural strength. The right side of the table shows the value each species got assigned to for this optimization.

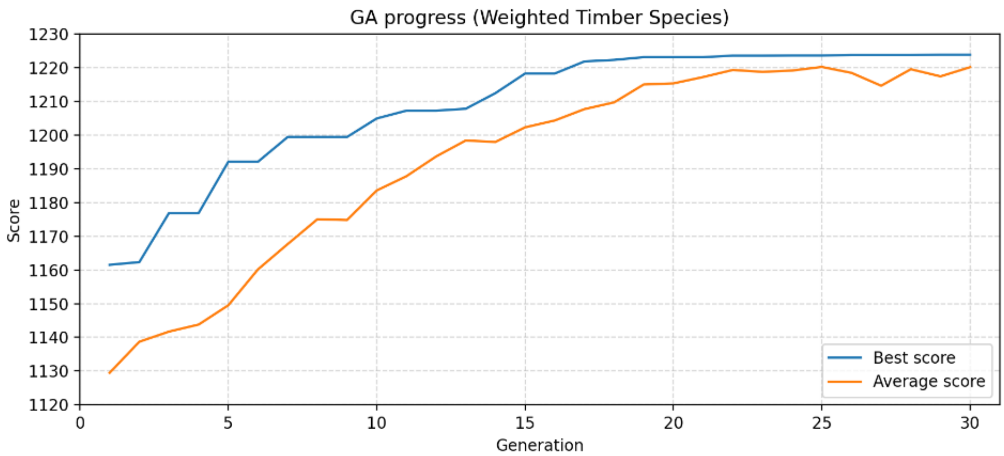


Figure 33: Graph displaying the GA progress over 30 generations with weighed timber species.

5.3.2 Exploring a more extreme reference building

Table 9 shows the required amount of logs for the FOR building. In figure 34 are 10 of the 167 cross-sections that are generated for the element allocation. This selection of 10 elements are chosen because they give a good overview of the way the elements are distributed throughout all 167 logs.

Results binpacking FOR	
Required logs	Total: 167
	63x Maple
	41x Ash
	22x Robinia
	19x Poplar
	9x Elm
	8x Oak
	4x Chestnut
	1x Beech
Average utilisation	34.11%

Table 9: Required Logs & Average Utilization for a simplified *FOR structure.* According to the framework.

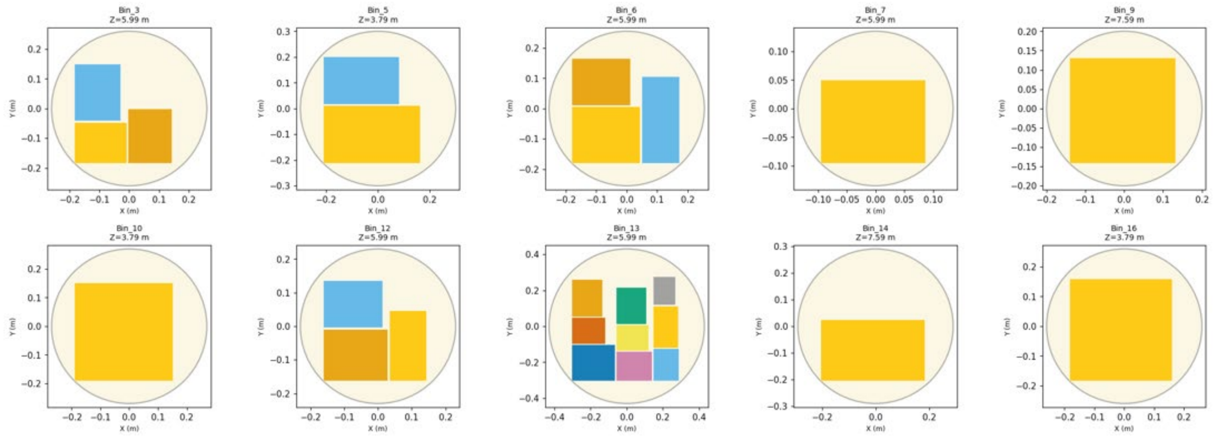


Figure 34: 10 of the 167 Cross-sections generated for the element allocation for the FOR-like structure.

A condensed collection of all 167 cross-sections can be found in the appendix.

5.3.3 Inclusion of biological behavior

In figure 35 are two cross-sections shown that represent the distribution of elements along the 34 trees. Smaller elements are packed around the pith, where large rectangular columns box the pith in. All 34 cross-sections of biological behavior can be found in the appendix. Table 10 sums up the overall required trees to supply for all load bearing elements including the biological behavior.

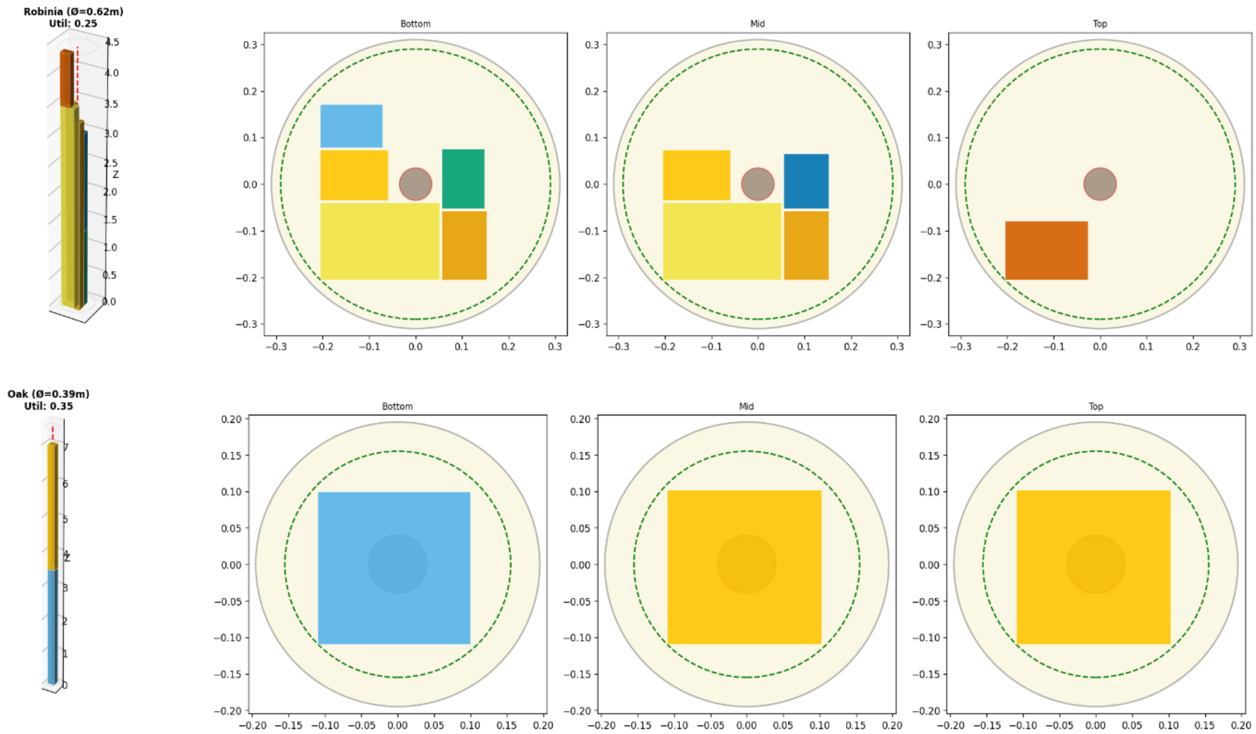


Figure 35: Two of the cross-sections as a result of the inclusion of biological behavior.

Results binpacking with biological behavior	
Required logs	Total: 34
	11x Robinia
	7x Oak
	6x Ash
	1x Beech
	7x Maple
	2x Chestnut
Average utilisation	22.97%

Table 10: Bin packing performance after including the biological restrictions.

5.3.4 Inclusion of non-loadbearing elements

Figure 36 shows two cross-sections that display a 60%+ utilization as a result of the inclusion of non-loadbearing elements. These two cross-sections are chosen because they represent the way the 60%+ utilization bins are filled. Figure 37 (next page) shows the cross-sections where there are no longer small elements to pack, resulting in lower utilized bins.

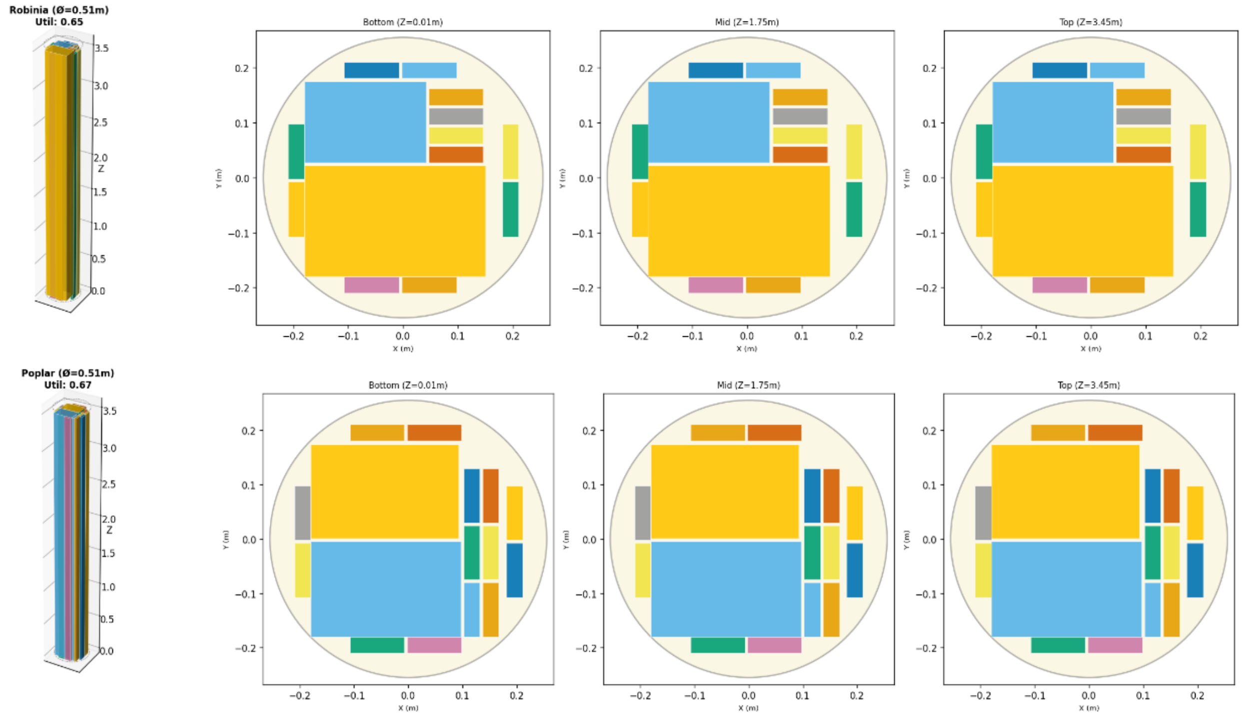


Figure 36: Crosssections with a utilisation above 60% as a result of the inclusion of non-load bearing elements.

Results binpacking with non loadbearing elements	
Required logs	Total: 49
	13x Robinia
	13x Poplar
	12x Maple
	5x Ash
	3x Oak
	2x Chestnut
	1x Elm
Average utilisation	39.62%

Table 11: Utilisation table for all bins in the situation of *Woonhuis Muiden* with the addition of non - loadbearing elements.

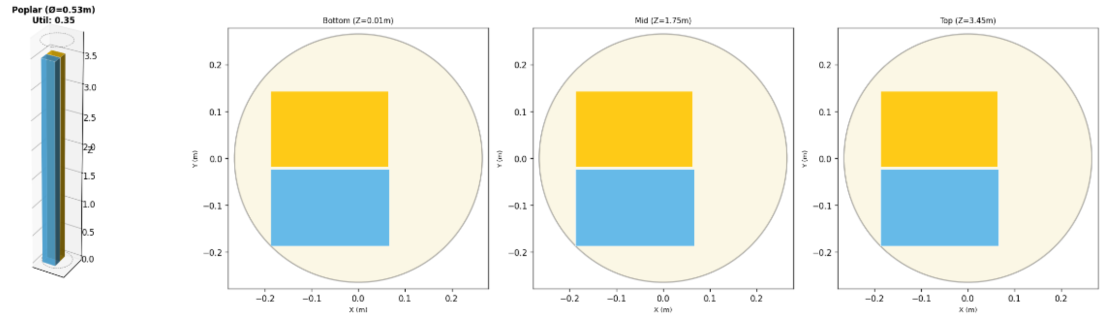


Figure 37: A crossection of a log scoring below 40% utilisation; packed in the scenario of included non-loadbearing elements.

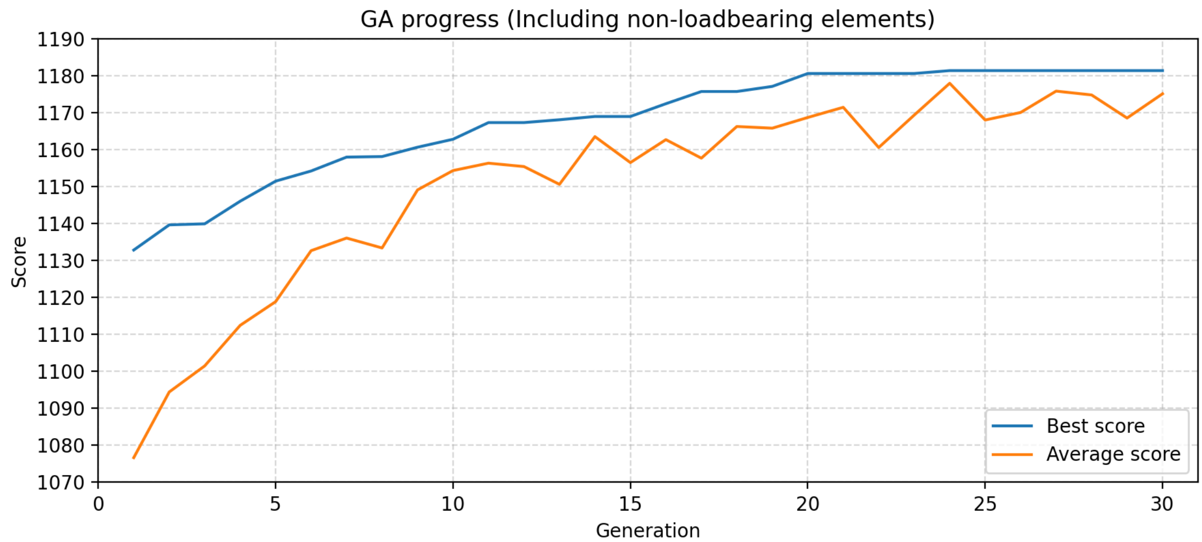


Figure 38: Graph displaying the GA progress over 30 generations with the inclusion of shade elements.

5.3.5 Rounded load-bearing element dimensions

Figure 39 shows a selection of 3 cross-section of bins that are packed with load-bearing elements that have their dimensions rounded up to a significance of 5 cm. Table 12 shows the required amount of bins for this distribution and the overall average utilization of said bins. The graph in figure 40 (next page) shows the progress of the genetic algorithm.

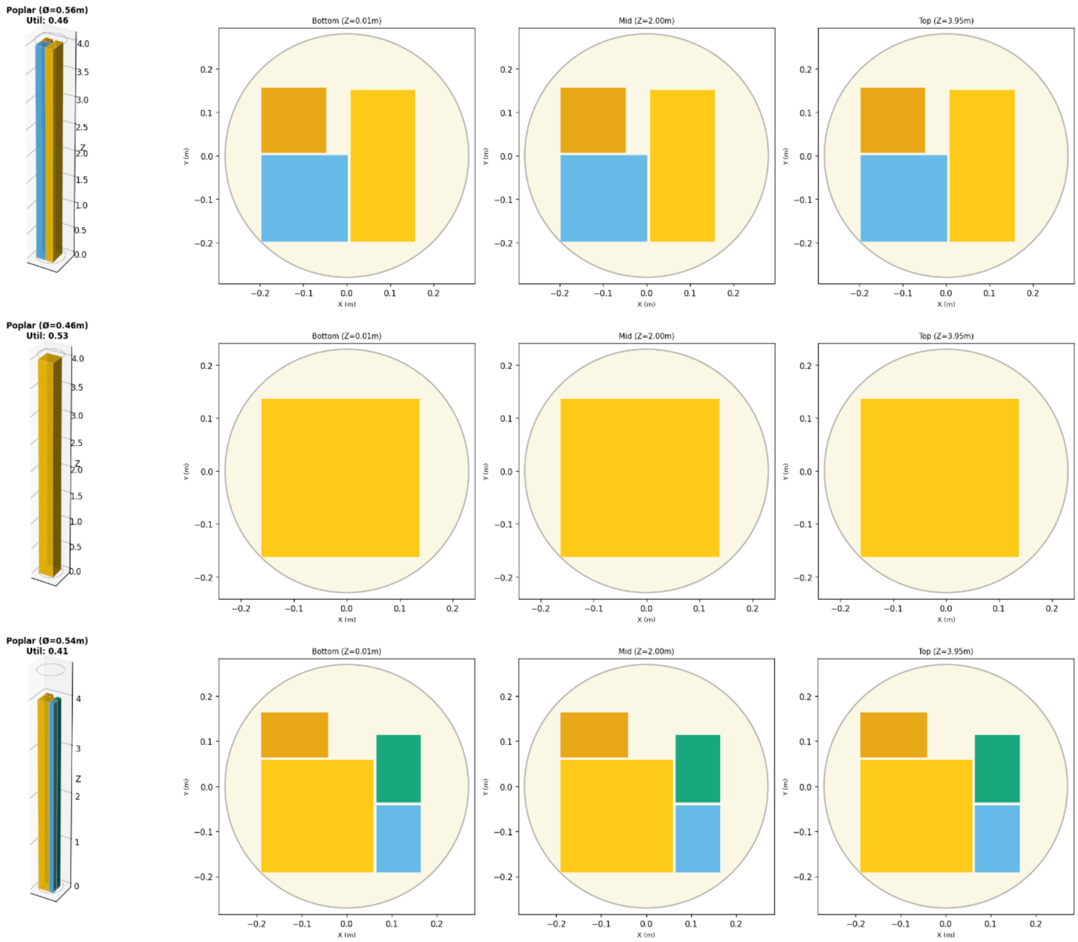


Figure 39: A selection of 3 crossections of loadbearing elements with their dimensions rounded to a 5 cm significance.

Results bin packing with rounded elements	
Required logs	Total: 36
	14x Robinia
	17x Oak
	3x Ash
	8x Maple
	1x Chestnut
	10x Poplar
Average utilisation	33.63%

Table 12: Utilisation table for all bins in the situation of *Woonhuis Muiden* with all loadbearing elements rounded to a 5 cm significance.

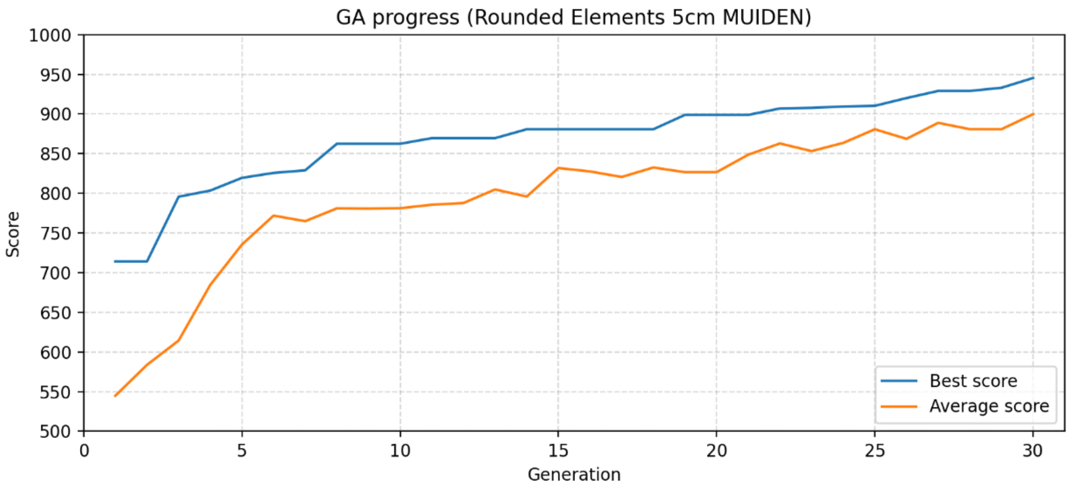


Figure 40: Graph displaying the GA progress over 30 generations of elements with their dimensions rounded to 5 cm significance.

5.3.6 Exploring different GA Hyperparameters & fitness-score

Population size of 500

Figure 41 shows the GA progress graph with the population size set to 500 instead of the population size of 50 from previous results. Table 13 sums up the total stock usage and the average utilization across the opened bins, as a result of the solution with **500 population.**

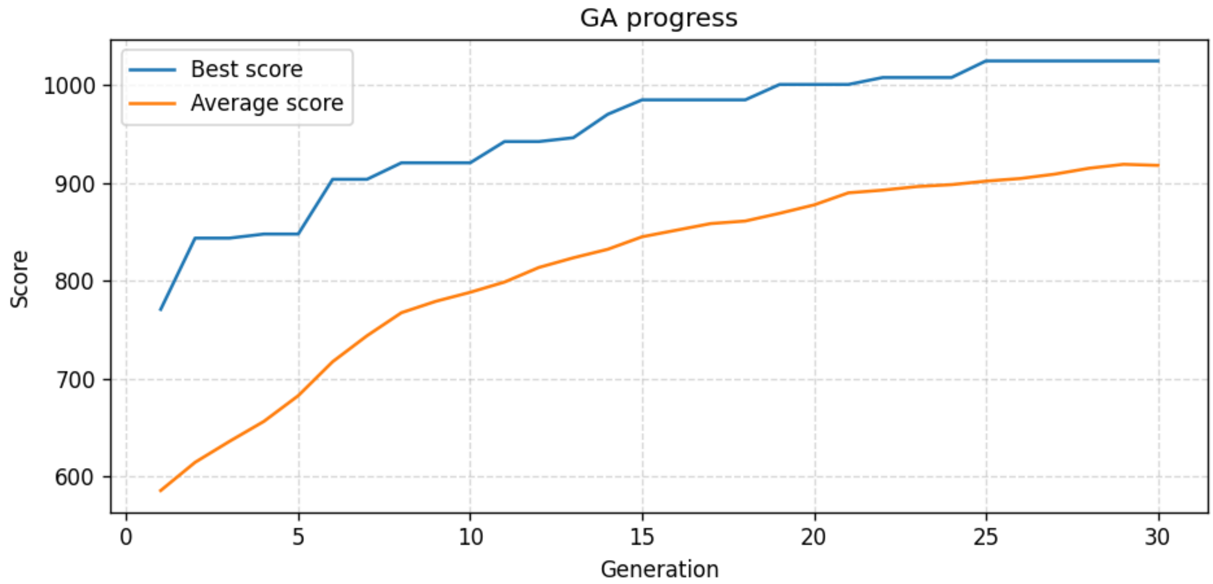


Figure 41: Graph displaying the GA progress of 500 population over 30 generations.

Results bin packing with population size 500	
Required logs	Total: 35
	10x Ash
	7x Elm
	7x Maple
	6x Poplar
	4x Robinia
	1x Oak
Average utilisation	39.28%

Table 13: Utilisation table as a result of a population size of 500.

New fitness-score with a bin-opening penalty of -40

Table 14 shows the results of the element distribution over the stock, as a result of a higher **bin-opening penalty (-40)** in the fitness-score. Figure 41 shows the GA progress graph with the newly set fitness-score. Figure 42 (next page) shows 3 cross-sections of two of the packed bins with the -40 bin opening penalty. These 3 bins are chosen out of the 22 bins because they best show the element stacking that is happening as a result of the -40 penalty.

Results bin packing with -40 bin penalty	
Required logs	Total: 22
	7x Maple
	11x Ash
	1x Robinia
	3x Chestnut
Average utilisation	38.39%

Table 14: Utilisation table as a result of a new fitnessscore with a -40 bin opening penalty.

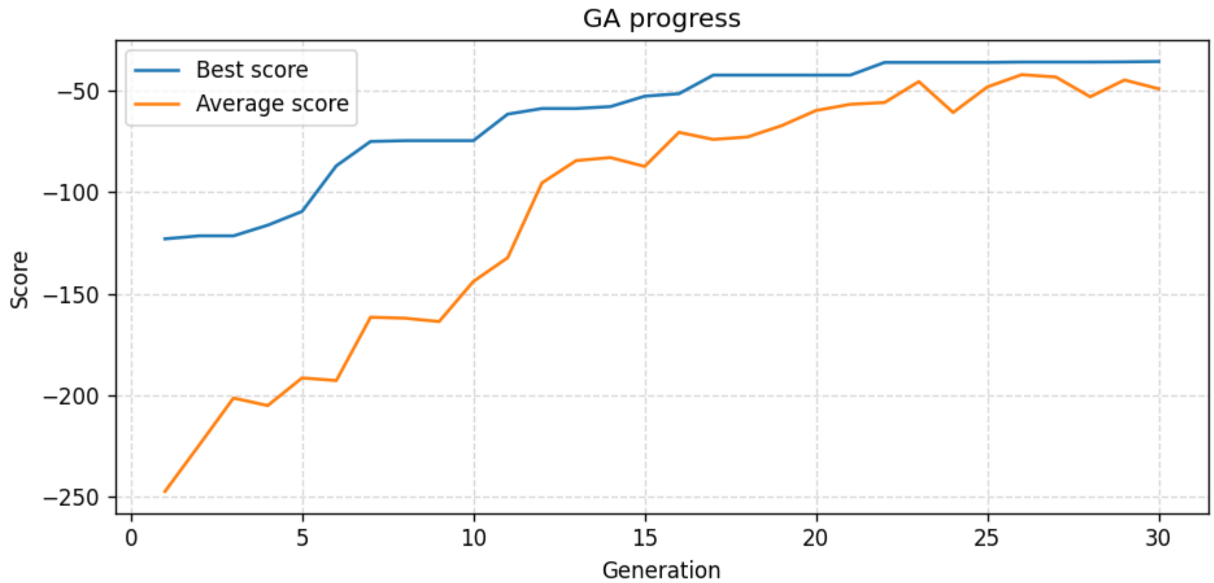


Figure 42: Graph displaying the GA progress with a -40 bin penalty fitness score.

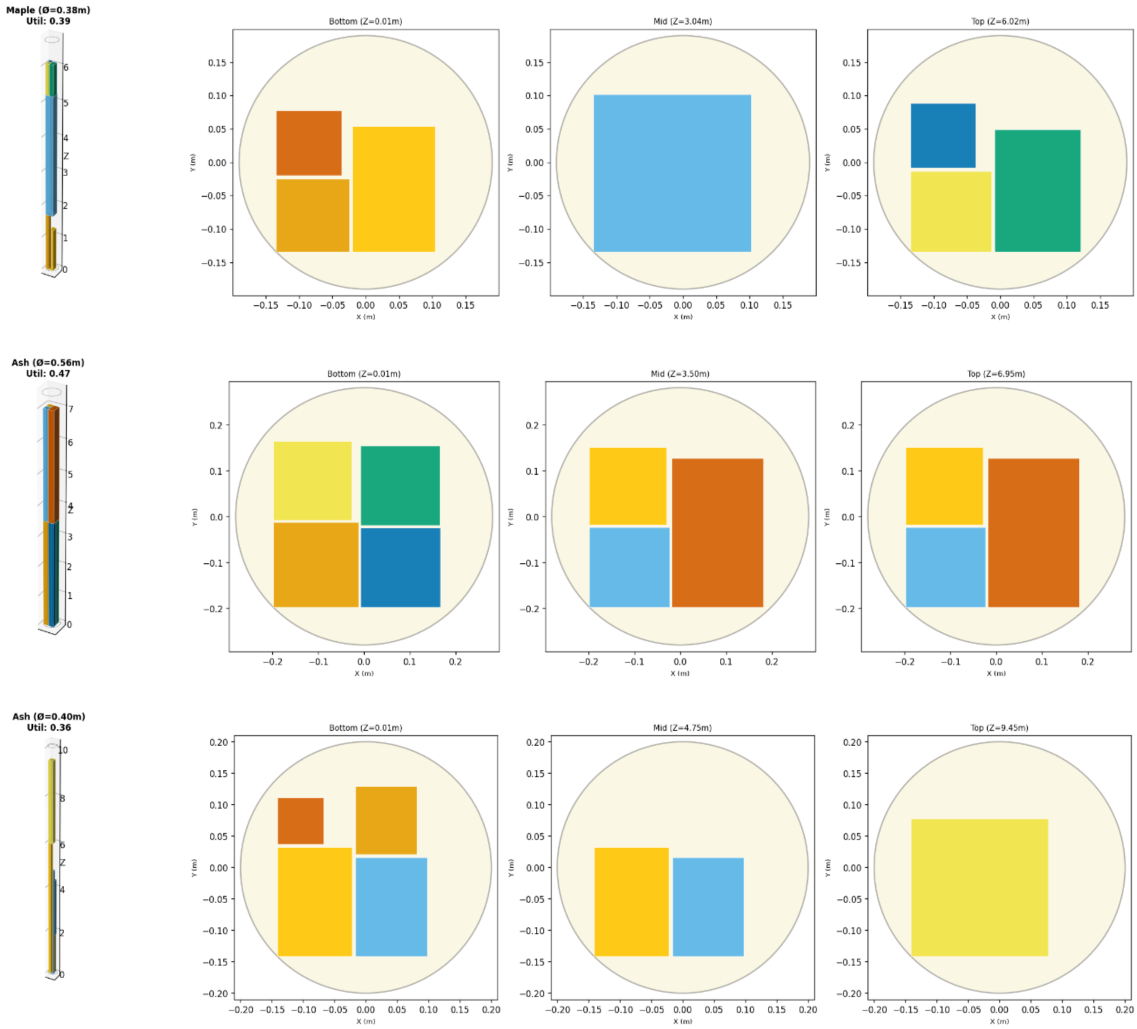


Figure 43: Three bins with cross-sections as a result of the -40 bin penalty.

5.3.7 Preserve tallest trees

As mentioned in 3.4.7, an alternative heuristic approach is tested in order to preserve the tallest trees. The heuristic is this scenario only accepts bins within a 20cm range of the initial element. In figure 44 the progress of the GA is plotted as a result of the height preservation-heuristic with a fitness-score bin penalty of 40 points. In table 15 the overall stock utilization of this solution is shown. In figure 45 (next page) four different bins are shown with their corresponding cross-sections. These 4 bins are chosen out of the 27 packed bins in order to best show the element stacking that occurs due to the heuristic logic change.

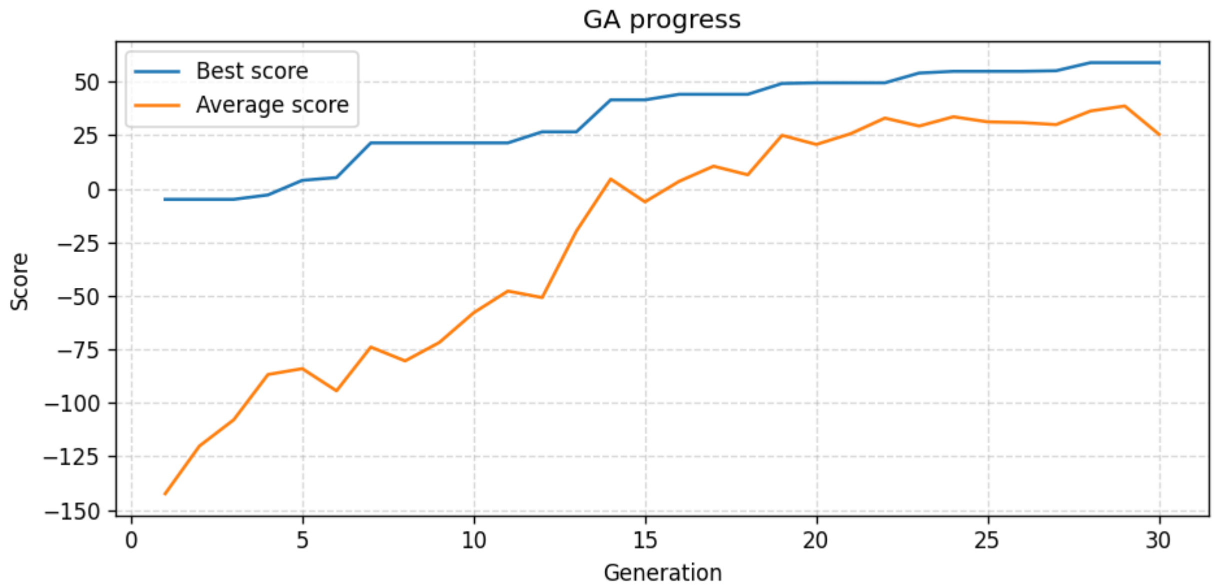


Figure 44: GA progress graph as a result of the height preservation-heuristic, with a fitness-score bin penalty of -40, with 50 population over 30 generations.

Results tallest tree preservation and -40 penalty	
Required logs	Total: 27
	9x Maple
	7x Robinia
	5x Ash
	4x Poplar
	2x Elm
Average utilisation	42.18%

Table 15: Utilization table of the stock usage with the solution of the height preservation heuristic, and a fitnees-score bin penalty of -40.

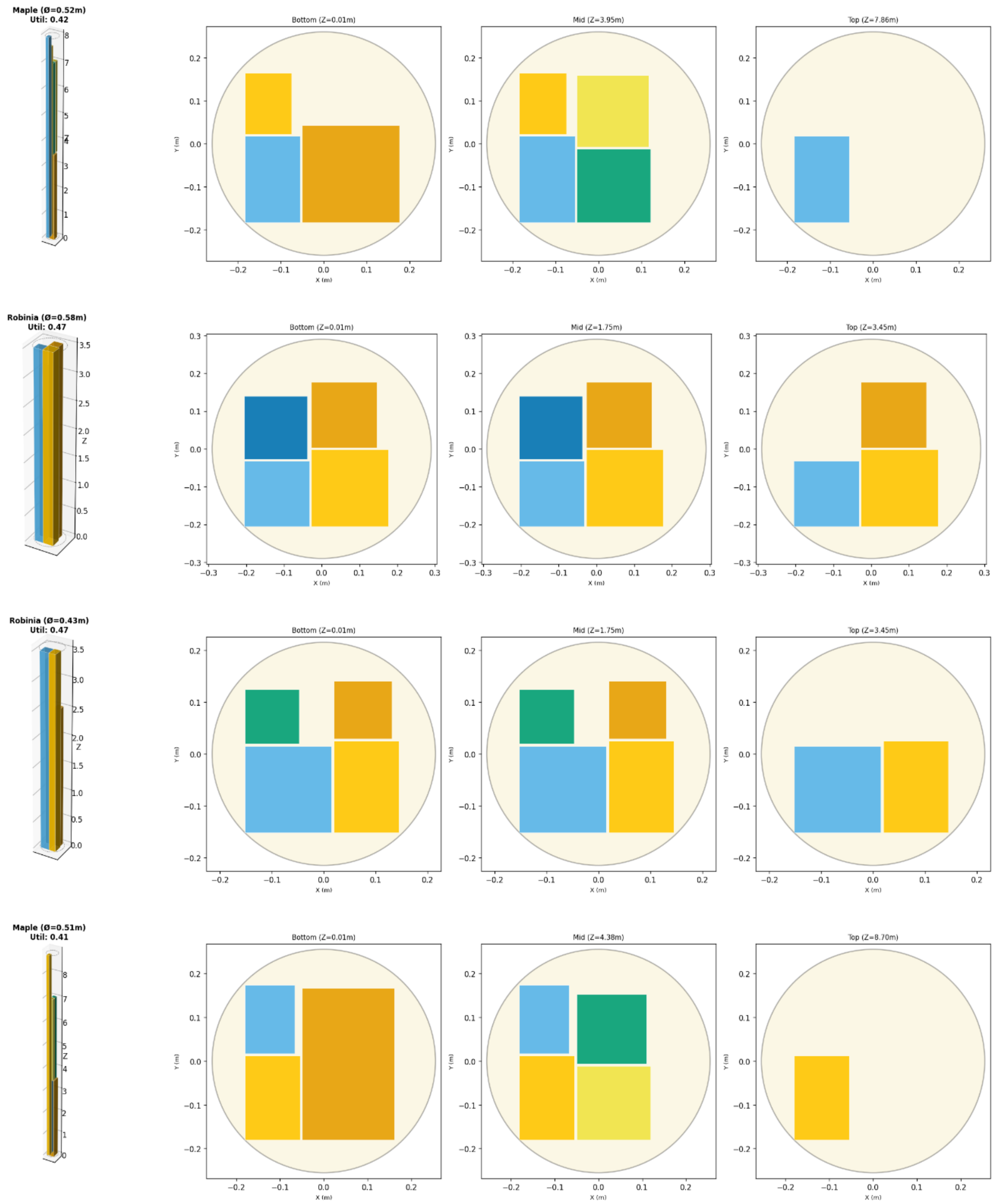


Figure 45: Four bins with cross-sections as a result of the height-preservation heuristic and a -40 bin penalty.

5.3.8 Alternative GA approach

5.3.8.1 Permutation based on a series generated by a formula of 4 variables

In chapter 3.3.2.4 alternative genetic algorithms were proposed. The first approach is based on a formula that generates a series of integers by which the indices of the permutation are moved around with a strength variable. The parameters that make up the variables in the formula, and the strength variable are the input variables for the genetic algorithm. The PyGAD library is used as the genetic algorithm for the results. In this section the results of this approach are presented. There is experimented with two different formula's. The first contains 4 variables for series generation and the second uses 8 variables. These formula's are tested for a population size of 50 and 500.

50 pop 30 gen (4 formula variables)

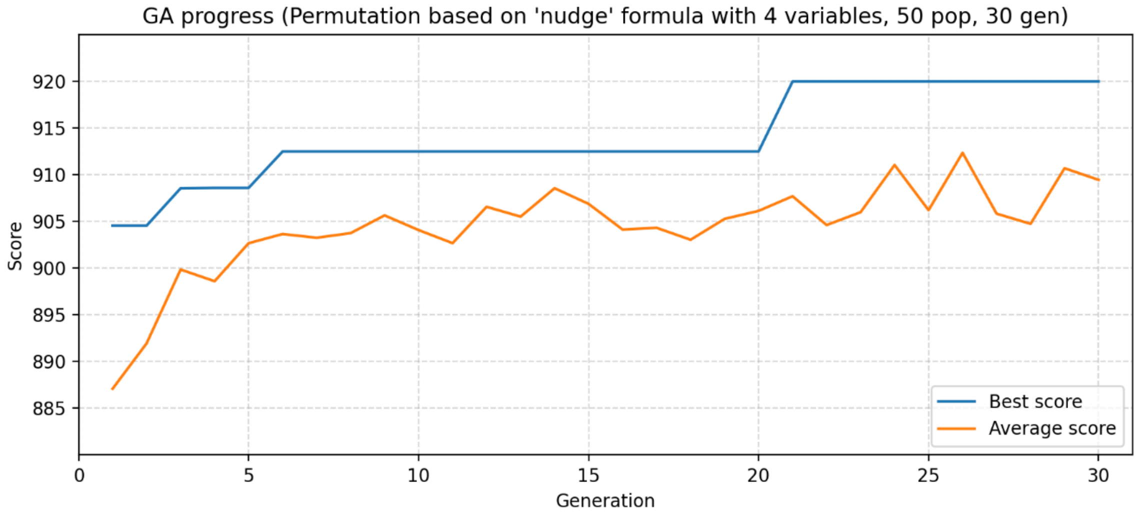


Figure 46: GA progress graph of the alternative formula-based approach, using variables to generate a series that nudges the permutation, 4 variables, 50 population size, 30 generations.

Results binpacking series generated 50 pop	
Required logs	Total: 39
	14x Robinia
	3x Oak
	3x Ash
	8x Maple
	1x Chestnut
	10x Poplar
Average utilisation	33.59%

Table 16: Stock utilization report of the alternative formula-based approach, using variables to generate a series that nudges the permutation, 4 variables 50 population size, 30 generations.

500 pop 30 gen (4 formula variables)

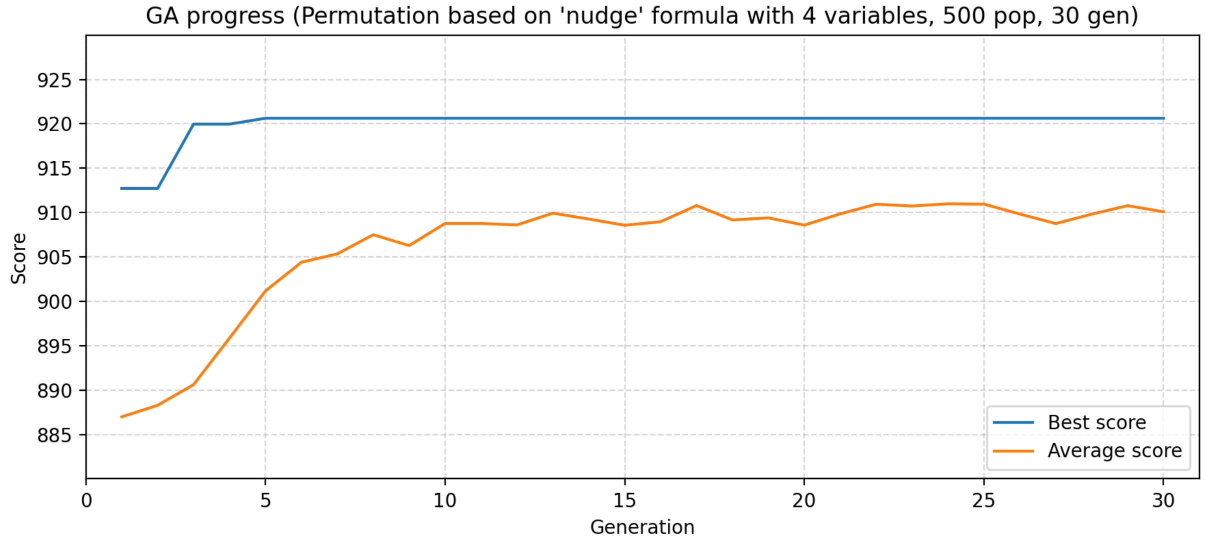


Figure 47: GA progress graph of the alternative formula-based approach, using variables to generate a series that nudges the permutation, 4 variables, 500 population size, 30 generations.

Results binpacking series generated 500 pop	
Required logs	Total: 39
	14x Robinia
	3x Oak
	3x Ash
	8x Maple
	1x Chestnut
	10x Poplar
Average utilisation	33.61%

Table 17: Stock utilization report of the alternative formula-based approach, using variables to generate a series that nudges the permutation, 4 variables 500 population size, 30 generations.

50 pop 30 gen (8 formula variables)

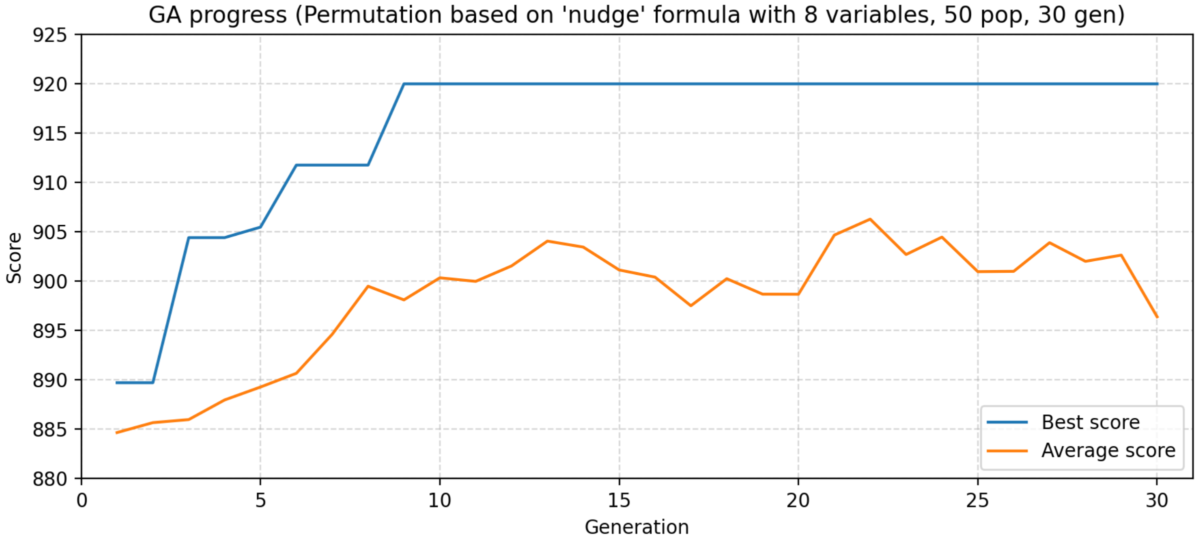


Figure 48: GA progress graph of the alternative formula-based approach, using variables to generate a series that nudges the permutation, 8 variables, 50 population size, 30 generations.

Results 8 variables series generated 50 pop	
Required logs	Total: 39
	14x Robinia
	3x Oak
	3x Ash
	8x Maple
	1x Chestnut
	10x Poplar
Average utilisation	33.59%

Table 18: Stock utilization report of the alternative formula-based approach, using variables to generate a series that nudges the permutation, 8 variables 50 population size, 30 generations

500 pop 30 gen (8 formula variables)

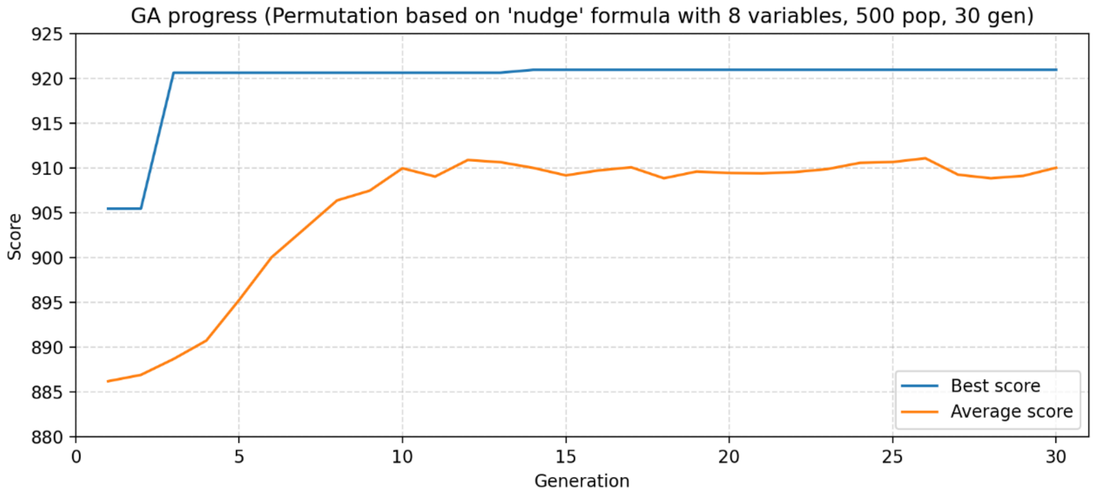


Figure 49: GA progress graph of the alternative formula-based approach, using variables to generate a series that nudges the permutation, 8 variables, 500 population size, 30 generations.

Results 8 variables series generated 500 pop	
Required logs	Total: 39
	14x Robinia
	3x Oak
	3x Ash
	8x Maple
	1x Chestnut
	10x Poplar
Average utilisation	33.61%

Table 19: Stock utilization report of the alternative formula-based approach, using variables to generate a series that nudges the permutation, 8 variables 500 population size, 30 generations

5.3.8.2 Permutation based on formula with property weights

In chapter 3.3.2.4 alternative genetic algorithms were proposed. The second approach is based on adding a weight-value to certain bin and element properties. These weights are variables that change the order by which the elements and bins are placed in the permutation. The weight-variables are the input parameters for the PyGAD genetic algrotihm. This property-weighted formula based approach is tested for a population size of 50 and 500.

Formula based approach with weighted properties, 50 pop 30 gen

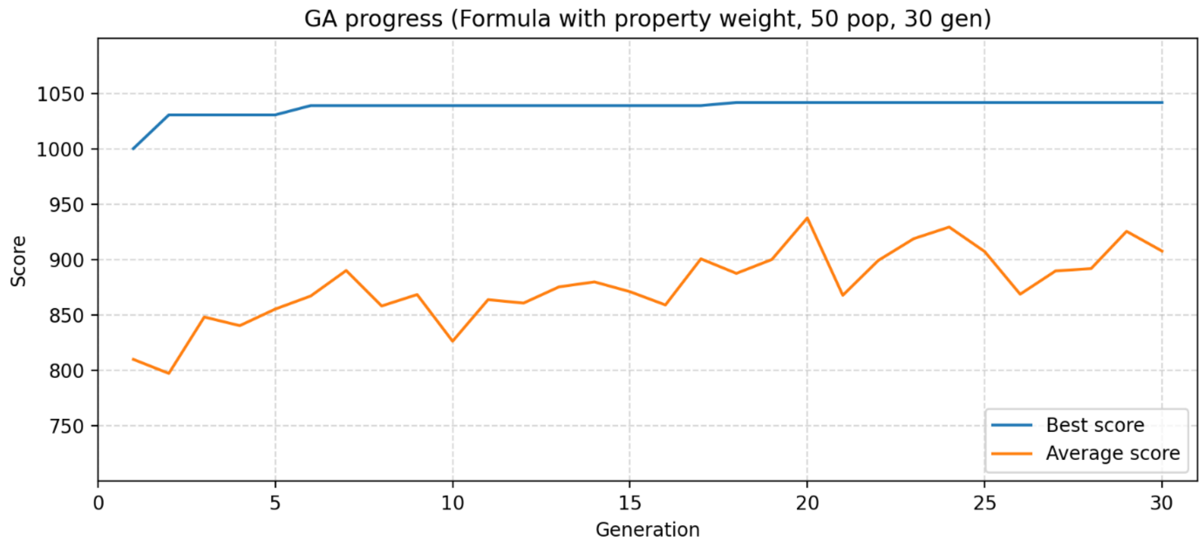


Figure 50: GA progress graph of the alternative formula-based approach, using property weights to generate new permutations, 50 population size, 30 generations.

Results packing weighted property fomula 50 pop	
Required logs	Total: 32
	9x Poplar
	9x Robinia
	8x Maple
	4x Ash
	1x Elm
	1x Oak
Average utilisation	40.61%

Table 20: Stock utilization report of the alternative formula-based approach, using property weights to generate new permutations, 50 population size, 30 generations.

Formula based approach with weighted properties, 500 pop 30 gen

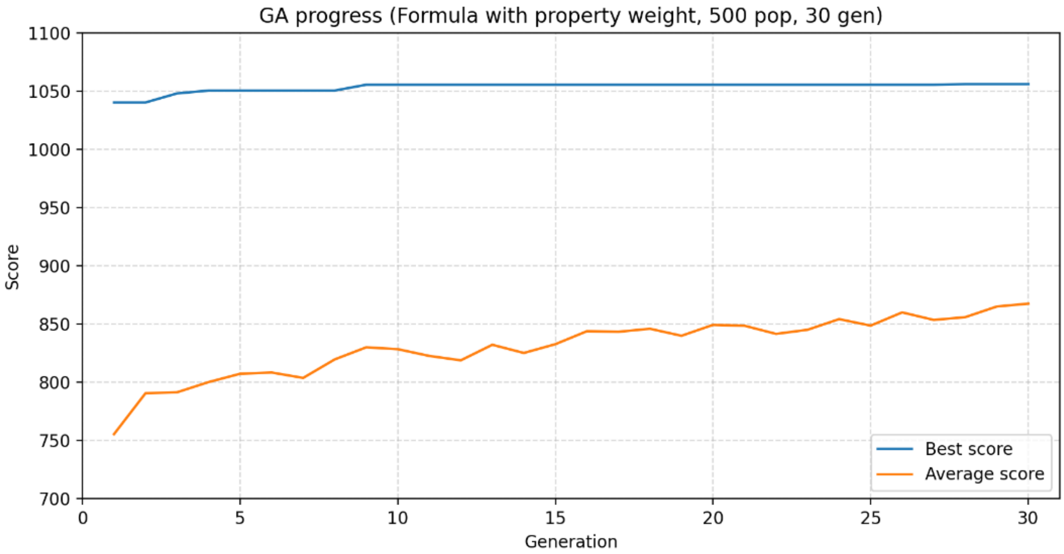


Figure 51: GA progress graph of the alternative formula-based approach, using property weights to generate new permutations, 500 population size, 30 generations.

Results packing weighted property fomula 500 pop	
Required logs	Total: 33
	10x Poplar
	9x Robinia
	7x Maple
	4x Ash
	1x Elm
	1x Oak
	1x Chestnut
Average utilisation	39.76%

Table 21: Stock utilization report of the alternative formula-based approach, using property weights to generate new permutations, 500 population size, 30 generations.


06
Discussion

Chapter 6: Discussion

In this chapter the results from the previous chapter will be discussed. The discussion will go over the findings in the order as they are presented in the results chapter, while also cross-referencing results to draw conclusions from them.

6.1 Element dimensioning

The findings of the element dimensioning as shown in C5.1 are discussed along with the results in the section itself, to justify the dimensioning choices that are used for the results of C5.2 onwards. Therefore the first results that will be discussed are from C5.2 – Cutting pattern generation.

6.2 Cutting pattern generation - Heuristic vs MetaHeuristic

C5.2 starts off with the results of the initial heuristic run based on ascending bin height and descending element order in figure 30. The average utilization of bins for these results lay around 30%, with the lowest being 25% and the best scoring solution having a 33% average utilization. These results are not near the goal of the industrial standard of 50% bin utilization as documented by Brandstetter et al. (2020).

Upon inspection it could be noticed how the solution for the 2x8x8x2 frame requires 41 trees, while the solution for the *Muiden* reference building requires only 39. This is interesting because the *Muiden* building requires more elements. This difference however can be explained by the fact that since the *Muiden* building has more elements it has also elements from more varying sizes. This means that the bins that will be opened are more diverse. This could leads the possibility of stacking smaller elements on top of each other in a taller bin. The *Muiden* reference building requires several 8 meter beams, this means that there are 8 meter tall bins opened. It could be possible that several 4 meter elements stack on each other within this 8 meter bin, leading to a lower required amount of trees. This is further explored in 5.3.6 and 5.3.7.

Another noticable phenomenon in these results is the primairy choice for Robinia and Poplar trees. This however can also be simply explained by the fact that a lot of the elements for these buildings are around 4 meters in height, and with the current stock of trees, a lot of the Robinia and Poplar trees are relatively close to 4 meters in height.

The GA graph of figure 31 shows that a score improvement of ~200 points is achieved in 30 generations of optimizing. The initial value of the first population is 770.9, where as the final value is 976.9. These 200 points are the product of the elimination of 5 bins together with 9% overall utilization gain. This shows that within a population size of 50 and 30 generations the use of a genetic optimizer has significant increasing results.

This is also confirmed in table 6 and figure 32, showing a significant increase in utilization and decrease is the amount of required trees. Overall the utilization now lies around 42%, when using genetic optimization. These numbers are much closer to the industry standard of 50%, and these solutions can be considered a viable and worthwhile in actual practice.

6.3 Parameter exploration

With the promising results of the overall approach of the heuristic and custom GA metaheuristic, it is interesting to apply the algorithm to different circumstances. In the parameter exploration minor changes to the overal framework or input are experimented with in order to get a grasp of the possibilities of the proposed framework.

6.3.1 Weighted timber species

The parameter exploration starts off with adding weight to every timber species in the fitness-score of the GA.

The results in table 8 show a strong distribution according to value preference. The three timber species that weight the most, Poplar, Maple and Chestnut together make up for 87% of the required logs. This shows that the GA can easily adept to the weighing of species and can create solutions based on the preference of it's user. The weighing of timber species can also prove useful for utilization the overall stock in a smart manner; by i.e. keeping high structurally timber away from the solutions and only utilize them when necessary.

It is interesting to notice that the average utilization is similar in comparison to the unweighted metaheuristic result. Apparently the stock allows for both Poplar as Robinia to be utilized similarly in the case of the *Woonhuis Muiden* structure reference. This can be explained by the height similarity mentioned earlier. Most of the Robinia and Poplar stock are within 4 meters height. Since Robinia is the lowest rated timber in this variant and Poplar the highest rated, it is logical that the solution would interchange most of the Robinia bins for Poplar, allowing it to score simple points.

6.3.2 Exploring a more extreme reference building

At the time of writing this thesis 605 trees are in the database that have been marked/requested for removal. Of these trees 270 are taken into consideration for the computational framework. The results as shown in table 9 suggest that with 167 trees a structure similar to the *FOR* could be made. This means that with roughly ~62% of the available trees a structure of this size could hypothetically be made according to the framework. By examining table 9 there is a clear preference for Maple and Ash trees. That is because in the current database used to represent the Rotterdam tree stock most of the Maple and Ash trees are around 6 to 8 meters in height. The primary grid size of the FOR building is also 6x6 meters, explaining why there is such a strong shift towards Maple and Ash.

By examing the cross-sections it is interesting to see that about half of the cutting patterns only consist of one element, as can be seen in figure 34. This can be explained by the fact that most the smaller elements could be more tightly packed into bins that were already opened, since the FOR building requires a lot of bin to be opened. This results in many bins that tightly fit just one element. This is not necessarily a bad result, but an interesting development within the larger element dataset of the FOR structure.

6.3.3 Inclusion of biological behavior

Table 10 shows two interesting changes in comparison to the results from the original framework. There is a drop in utilization, while remaining the same in required trees. Utilization drops from 34.46% in the original to 22.97% with biological restraints. This clear drop can be explained by the fact that a portion of the trees volume has been marked unusable. The volume however remains, meaning that even if the heartwood would be fully packed there would not be a utilization of 100%. Another reason the utilization is lower is because the current heuristic is not optimized for this approach. Including pith- and sapwood-restrictions would ideally require changes to the heuristic. A new approach to better suit the restrictions could be to make a heuristic that builds from the center outwards, primarily choosing a boxed-in pith as the start.

The similarity in required trees can be explained by the fact that the framework has chosen significantly more tall trees. This choice allowed stacking column on top of each other at the pith; as shown in figure 35. Since more tall trees have been utilized during this test run the numerical amount of trees remained the same even though the volume of trees would be in line with each other.

6.3.4 Inclusion of non-loadbearing elements

Figure 36 shows that with the inclusion of these smaller elements higher utilizations per tree could be achieved. This is in line with the findings of Klement et al. (2023), as mentioned before. That research saw conversion rates of 65% to 72% when processing 5x5cm blanks. The shades used for the model below are 3x10cm.

Even though these new non loadbearing elements allowed for a denser pack in some of the logs, the overal average utilisation is still below 40% (Table 11). This is because when all the non loadbearing elements were placed many bins were still required to account for all the loadbearing elements. This lead to many logs to be utilized <40%, as can be seen in figure 37. Since the majority of the logs did not include the shading elements, the average follows this the trend.

Eventhough it is interesting to see that the utilisation can improve by the addition of non-loadbearing elements, it is important to bear in mind that these shading elements in this example come from different tree types. This could be unfavorable for aesthetic and functional reasons.

6.3.5 Rounded load-bearing element dimensioning

The experiment with rounded elements show similar results, however with a 5% decrease in utilization. The approach with rounded elements and the current framework has a flaw; the kerf size is not flexible in size. At first glance this feels logical because it is a restriction and cuts should not be narrower than the thickness of the blade. In the current approach however the kerf restriction does not allow solutions where the reserved kerf space is bigger than the required thickness. Ideally this should be allowed, since the thickness is only a minimum requirement, not a maximum. For most of the solutions this is not a problem but for the rounded elements this means that 2x 5cm squares can not be fit above 1x 10cm square. Because the 10cm square would create a bounding box of 10cm + 1x kerf size, whereas the 2x5cm squares would require 2x5cm (10cm) + 2x kerf size. Meaning that a lot of configurations are deemed impossible, even though a simple increased kerf space would allow it. Ideally the heuristic would be improved upon to solve this problem.

6.3.6 Exploring different GA Hyperparameters & fitness-score

500 population size

In section 5.3.6 the genetic algorithm was ran with a populations size of 500 over 30 generations. The results of this experiment however are underwhelming. Table 13 shows the best solution, but there is improvement is negligible in comparison to the run with 50 population. There is only a 1% increase in utilization. This is interesting because the graph in figure 41 shows are similar trend to the GA progress graph of the run with 50 population. This seems to indicate that the genetic algorithm does most of its valuable exploring over multiple generations rather then over a big population. It could be the case that most of the GA progress is in the cross-over function, where high scoring parts of the permutation are copied in different solutions. More research into this behavior would be required draw a conclusion.

Different fitness score

This section also explores a different fitness score. The original score of:

$$\text{score} = (100 \times \text{util\_sum}) - (10 \times \text{bins\_used})$$

is changed to:

$$\text{score} = (100 \times \text{util\_sum}) - (40 \times \text{bins\_used})$$

This means that the penalty for opening a new bin is 4 times as much as before. For every bin that is opened and has a utilization lower than 40%, that bin scores negative points. This explains why the GA graph in figure 42 start at -250. If at the start i.e. point 30 bins are opened this means a point reduction of -1200. For the score to be at -250 that would mean the sum of utilizations ($\times$ 100) would need to be 950. Meaning that every bin needs to be around 31% utilized. This is a believable start, when compared to earlier results.

It is interesting to see that this change in fitness score leads to the significant drop of a 12 bins. From 34 trees with the original scoring to 22 trees with the current fitness score. This result can be favorable in same cases. The reason for this drop in required trees is the same as mentioned in paragraph 6.2; more tall trees are chosen, allowing for more stacking of elements. This is shown in figure 43. Figure 43 shows three bins where there are multiple layers of stacked elements inside a tall tree, where it would otherwise chose to spread those elements over smaller trees. The impact of the change in fitness-score is of such significance that it is interesting to experiment further on this. Which is why the preserve tallest tree heuristic would seem promising in combination with this fitness-score.

6.3.7 Preserve tallest trees

The results of the preservation heuristic change are in line with the expectations. The overall required amount of trees drop from 34 to 27. Although this is of less magnitude than the 22 trees solution, the heuristic allowed to remain within this 20 cm height difference restriction. This means that there were no bins opened on unnecessary height. This can be seen in figure 45, where all opened bins match in length with their biggest element. The fact that table 15 shows that the average utilization of bins is also higher, it points towards the idea that this heuristic approach, together with a bin penalty of 40, seems the best approach.

6.3.8 Alternative GA approach

Permutation based on nudging with a series from a formula of variables

As mentioned earlier, the genetic optimizer used in the framework is not benchmarked. In order to use the pre-existing Python library PyGAD, two different approaches have been experimented with. The variant where a formula is used to generate a series of integers does not seem to yield as much exploration as maybe expect. Overall the outcome of the top solution is not bad, but it does underperform in comparison to the cross-over based algortihm. Table 16 shows an average utilization of 33.59% with a required tree amount of 39. This is a 5% lower average utilization and an increase 5 bins, when compared to the original GA solution. Table 17 shows the same GA approach but with a population of 500 instead of 50. However the outcome of the best solution is near the same as with a population size of 50. In figure 47 the graph shows that the GA stagnates quickly and that nearly no improvement is made after the 5^{th} generation. This implies that with a population size of 500, the best solution when using this formula based approach is quickly found. This likely means one of two things; either this specific formula is not a good method of exploring the fitness landscape or the formula is badly implemented and requires different hyperparameters and perhaps another fitness-score.

These results are followed by a more diverse formula to generate the integer series. This time the formula knows 8 variables. Together with the strength parameter this means that the GA has 9 input-variables by which to optimize. Regardless of the doubling in variables, the best solution that results from this formula is the same as with 4 variables. The same can be said about the results with a population size of 500. This seems to indicate that the approach with nudging the permutation based on a generated series requires either a lot more tweaking or is not a good method to explore different permutations, using a genetic algorithm. Further experimention with this approach is required to draw conclusions. Due to a lack of time this thesis will not touch further on this implementation, and could see further research.

A limitation this parameterized series approach could expierence is that a certain specific series that performs well is not copied as a whole. It could overlook niche but high-quality permutations that lie in small, isolated regions of the overall permutation space.

Permutation based on formula with property weights

The results of the framework that uses weight variables for certain bin and element properties initially look more promising. Table 20 shows a result that is scores higher than the original GA approach, with the standerd heurstic and a bin penalty of -10. This result however does not improve with an increase of population from 50 to 500. The graph also shows to stagnate rather quickly. A possible explanation for the stagnation could be that the weighted propery approach does not allow for extreme swaps. Most of the bin properties are interlinked with eachother; a tall tree usually also means it is larger in width. This limits the exploration space of the genetic algorithm. If it is harder for the genetic algorithm to create permutations with extreme swaps it could miss out on synergies as discribed in figure 12 in chapter 3.3.2.3. This could mean that the algortihm is more reliant on random mutations and immigrants. Further research into this approach, with more tweaking of (hyper)parameters would give a better insight into the possibilities and limitations of this approach.

Looking at the results of this paragraph, it would suggest to be a good choice to also experiment with standardized **permutation** based genetic algorithms instead of a regular genetic algorithm library. For a specific permutation based GA, that is already benchmarked, there is the software of PermGA. This is a permutation based GA, that works outside the python enviroement as a third party software. There is not enough time in this master research to couple the proposed framework with this external software. Further research would be interesting.

07 Conclusion

Chapter 7: Conclusion & Recommendations

This thesis sets out to investigate how computational optimization can be applied to utilize a diverse roundwood timber stock in load-bearing structures. To this end, a framework was developed using element dimensioning paired with bin packing and tested with the case study of felled urban trees in Rotterdam. The goal is to link an architecture design to a set of trees that could yield the elements required for such design.

The process to link the design to the trees can be divided into two main steps. The first step is to go from design to element size, and the second step would be to allocated these elements within the wood dataset. To explore these steps, in the context of Rotterdam's urban trees, this thesis addresses three research questions:

1. What could the urban felled wood in Rotterdam supply and what are the structural qualities of these trees?
2. How to parameterize the dimensions of loadbearing elements in a structural design to suit the framework?
3. How to allocate the loadbearing elements into the available timber stock?

To answer the first question an interview was conducted with a party involved in the utilization of Rotterdam's urban trees and literature has been researched. Of the 605 trees that are marked to be taken down, 270 trees will be considered for loadbearing use. The species of these trees are: Robinia, Oak, Ash, Beech, Elm, Maple, Chestnut & Poplar; mentioned in order of material density. An estimation on their size could range from 3 to 12 meters in a trunk size with a diameter from 0.3 to 1 meter, based on the species.

In order to parameterize the dimensions of loadbearing elements the standard structural analyses formula's can be rewritten. Rewriting each formula to a variant where the element size is the product of mechanical values times a ratio factor x. By combining these formula's with a structural simulation tool, any design for a loadbearing structure could be converted to a dataset of timber elements with regard to each species.

A bin-packing heuristic, combined with a genetic algorithm for optimization can give a distribution of the element within a stock of trees. The heuristic on itself could perform with a utilization around 30 to 40 percent, where the optimizer could help cut down on the amount of trees, improve the utilization within a log or help with distributing along a certain specie preference.

With the answers to these research questions in mind the main research question be answered. The main research question is: ''How can felled urban trees be processed into tailor-made load-bearing architectural elements using computational optimization?''. It can be said that by reducing the urban stock to eight species, parameterizing load bearing formula's and combining them with structural simulation software and a bin-packing heuristic, the urban trees can become part of pipeline that would allow them to be processed into tailor-made loadbearing elements.

The case study of this thesis shows that Rotterdam's urban trees could certainly be used within architecture for load-bearing purposes. A computational framework could help prove the feasibility using city timber for a certain design. The pipeline as proposed in this paper could be used as an initial tool to a designer to utilize city timber. By giving the user the required element sizes for each specie and automatically relating them to the available stock, the designer could choose to use certain trees beforehand. Coupling design and material availability.

The literature as studied in chapter one has shown a multitude of different stock utilization approaches. However most of these papers either cover one-dimensional stock, like steel, or are very shape specific, like the tree fork truss by Zachary & Martin, 2016. This paper is the new to covering a 3D stock utilization of roundwood timber for rectangular load bearing elements. It combines this with the case-study of city timber, which is also a less documented supply of wood. This paper hopes to shed light on the under-utilization of these city trees and highlight the possibilities.

The tailor-made elements as proposed in this framework allow for the tiniest possible configuration for each timber element, reducing the required volume of wood. This tailor-made approach however does come with two mentionable down-sides. First of all it make modular building practices much harder. It is one of the primary goals in this research to stay as high up as possible in the wood cascade. The idea behind this is that the wooden elements could be reused in a later stage of it's lifecycle. Ideally being reused in an unmodified fashion. However the tailor-made elements occur in unconventional sizes. This makes it harder for these elements to be reused since they were dimensioned to suit one specific use case. Adding a standardization process to the current framework could make it more relevant to modular building practices.

Similar things can said in regard to engineered timber. Since this research prioritized the wood cascade the choice for sawn timber has been made over engineered timber. However, looking back at the results and the process in this research, the dimensioning required with sawn timber could be very limiting for volume utilization. With engineered timber the wood undergoes a more automated process and can lead to higher utilizations. Engineered timber could also allow for more options since beams could be made into large spans. For sawn timber the tree height is the limiting factor for how for a beam can span or how long a column could be.

For this research assumptions have been made to be able to explore a possible approach within a reasonable scope. For more realistic applications however multiple restrictions are to be taken into consideration that were left out in this research. A major constraint this paper avoided is the inhomogeneous quality of a log. By assuming that the felt tree would be a cylinder of homogeneous quality vital aspects of wood have been left out. The parameter exploration briefly touches on biological behavior but factor like knots and cracks have been left out. Including knots and cracks in the allocation of timber elements in a log more realistic results can be achieved. By utilizing software like the USMR split detection, a improved framework could arise in combination with the proposal from this thesis.

Another important aspect that has not been considered is the joining of timber elements. In the current framework the columns and beams are dimensioned to the point where they cross. Effectively making it so that every beam lays on half of the column. For timber structures joining techniques can play a very important role in distributing the forces properly. By leaving this out of consideration it could be that the framework generates dimensions that are too optimistic.

For further research this paper would have several suggestions. The first suggestion would be to include 3D scanning analyses of the trees into this framework. Having a 3D model for each tree could make the element allocation more realistic. Knowing the actual geometry of the tree would give insight into the possible dimensions it's elements could. The cylindrical approach of the current framework is optimistic and is not the best representation of Rotterdam's trees. Such a research could lead to developments within the municipality of Rotterdam, by stimulating them to catalogue their trees with simple 3D scans. Recent technologies make it easier to perform such scans and there is already literature available on the 3D analyses of trees. Having 3D information available could make the transition to use urban timber more easy.

For further research this paper would recommend taking engineered timber into consideration. Engineered timber would allow for more flexibility and uniformity in a design. Also, as mentioned before, engineered timber could lead to a higher volume conversion per tree. A study into hybrid engineered timber using the ten most common city trees could find combinations that would allow for a new pipeline to arise. Even without hybrid solutions, a research including single tree specie engineered timber could result in a framework that is more market ready, with higher conversion yields.

During this research there has not been enough time to extensively experiment with benchmarked genetic algorithms. The approach that is used in the framework of this thesis is custom for this application, and has not been tested on different circumstances. More experimentation with benchmarked existing genetic algorithms like PyGAD would seem interesting. As mentioned in 6.3.8, this thesis touches two approaches in combination with an existing library but more experimentation is required. Also the implementation of the third party permutation based GA '*PermGA'*, or similar software could be interesting for further research.

That last suggestion for further development would be to like more in depth at the biological behavior of wood. In the parameter exploration is some experimentation with sapwood and the pith, however the restrictions were applied to the existing heuristic. A more in depth study of bin-packing with regard to warping and wood properties could lead vast improvements in the heuristic. Such a research would require strong computational knowledge but would fill a notable gap in the current literature.

In summary, this thesis has shown that a framework which utilizes urban timber for tailor-made load-bearing elements can be a valuable tool. It enables the use of material that is currently largely neglected. By explicitly linking an architectural design to trees that are scheduled for removal, the framework makes it easier for architects to incorporate such trees into their projects. In doing so, it establishes a pipeline where none currently exists and provides a roadmap for integrating city wood into structural design.

08 Reflection

Chapter 8: Reflection

Graduation process

-How is your graduation topic positioned in the studio?

My graduation topic touches structural design and computational design, within the scope of architecture. My topic can be subdivided into two parts. The first part relies on load bearing calculations and structural simulations of buildings. This requires a skill set that is taught in the first year of the Building Technology master program. The secondary part includes programming with software like Grasshopper and Python. Both are integrated subjects within the masters program. By combining computational design with structural design a hybrid workflow is researched in my graduation topic.

-How did the research approach work out (and why or why not)? And did it lead to the results you aimed for? (SWOT of the method)

For this thesis I used a design driven research approach. Before reaching a method by which consistent and scientific results could be gotten different strategies and methods were experimented with beforehand. This experimenting took more time then expected and was not properly documented meaning it could not be well presented in the findings. The final methodology however does yield the results I aimed for and looks promising as a proof of concept.

-If applicable: what is the relationship between the methodical line of approach of the graduation studio (related research program of the department) and your chosen method?

My findings can be presented in the common methodical line of approach where it consists of a literature research leading to a methodology by which the results will be won and discussed upon. This is a very traditional line of approach for the BT graduation.

-How are research and design related?

The research conducted in the earlier phase of the master thesis have directly influenced the final design of this thesis. By first defining a clear knowledge gap in the current research state a well defined and restricted research approach could be formulated. The research has pointed out that for stock constrained design there is no literature regarding a 3D framework where wood is utilized in a more standardized beam-column loadbearing structure. This framework is tested on Rotterdam's city wood as a case-study. Utilizing City wood is a topic this is getting more coverage in literature in the past few years but it is not as well documented as other under-utilized materials like i.e. the reuse of demolition waste.

-Did you encounter moral/ethical issues or dilemmas during the process? How did you deal with these?

During my research I have not encountered any moral/ethical issues in the process.

Societal impact

-To what extent are the results applicable in practice?

A primary goal of my research was to try to reach a high applicability to practice. However for the sake of research and to manage it within the time scope of a master thesis limitations have been set. The case-study of Rotterdam's city timber is a real scenario with real data being used in the experiments. However since not all information is available about the trees in regards to size and shape some estimation and simplifications have been made. This means that the experiments are more an estimation and speculation of the possibilities rather then directly applicable factual results. It does however show proof of concept and allows for further development to reach a more realistic state. Other limitations within the computational framework like 'guillotine cutting' are meant to represent real life practices like the bandsaw cutting that is most commonly used in sawmills. This restriction is meant to make the results of this thesis more suitable for the current state of the market rather than a niche advanced approach that can only be used in specialized situations.

-To what extent has the projected innovation been achieved?

The goal of this research was to explore the possibilities of a framework that automates cutting patterns to utilize city timber for loadbearing structures. This goal is reached in this thesis by the fact that a framework is proposed that theoretically reaches a timber yield that would seem viable in real life situations. The framework however does require further development and experimentation for it to be marketable. Meaning the framework is still in a early stage of innovation.

-Does the project contribute to sustainable development?

Currently city wood is a underutilized material. The built environment accounts for up to 40% of the current energy demand. Timber structures can have a Global Warming Potential ten times lower then steel structures, when taking the embodied carbon into account. By constructing a framework where the city timber could be used for timber structures both the utilization of city wood and the construction of sustainable building could be boosted. Meaning this framework could contribute better material usage and increased sustainable building practices.

-What is the socio-cultural and ethical impact?

The proposed framework can provide handles for a discussion in regard to city trees. By showing that there could be a pipeline for city trees into timber construction it can spark the discussion to increase utilization and change the way we handle the felling of city trees right now.

-What is the relation between the project and the wider social context?

There is a national housing problem and a global climate crisis. This master thesis researches a framework that touches on both subjects. This thesis give a suggestion for a more sustainable building approach, allowing for the continuation of housing construction while still aiming to meet the Paris climate agreement.

-How does the project affects architecture / the built environment?

This project affects the built environment by proposing a new approach to the way we use city trees. Ideally the framework as researched in this thesis creates a pipeline where there is an automated, scalable city wood supply that would promote and facilitate timber construction.

/References

. Arup. (2019). *Rethinking timber buildings: Seven Perspectives on the use of timber in building design and construction* [PDF]. https://www.arup.com/globalassets/downloads/insights/rethinking-timber-buildings.pdf

. Bekin, M. (2019, September 27). *Sapwood vs Heartwood*. Ecochoice. https://ecochoice.co.uk/sapwood-vs-heartwood/

. BRANDSTETTER, M., ISPAS, M., & CAMPEAN, M. (2020). CONVERSION EFFICIENCY OF FIR SAWLOGS INTO LUMBER [Journal-article]. *PRO LIGNO*, 68–74. https://www.proligno.ro/en/articles/2020/4/BRANDSTETTER_Final.pdf

. Brütting, J., De Wolf, C., & Fivet, C. (2019). The reuse of load-bearing components. *IOP Conference Series Earth and Environmental Science*, *225*, 012025. https://doi.org/10.1088/1755-1315/225/1/012025

. Brütting, J., Senatore, G., & Fivet, C. (2018). Optimization Formulations for the Design of Low Embodied Energy Structures Made from Reused Elements. In *Lecture notes in computer science* (pp. 139–163). https://doi.org/10.1007/978-3-319-91635-4_8

. Brütting, J., Ohlbrock, P. O., Hofer, J., & D'Acunto, P. (2021). Stock-constrained truss design exploration through combinatorial equilibrium modeling. *International Journal of Space Structures*, *36*(4), 253–269. https://doi.org/10.1177/09560599211064100

. Brütting, J., Senatore, G., Schevenels, M., & Fivet, C. (2020). Optimum design of frame structures from a stock of reclaimed elements. *Frontiers in Built Environment*, *6*. https://doi.org/10.3389/fbuil.2020.00057

. Brütting, J., Senatore, G., & Fivet, C. (2021). Design and fabrication of a reusable kit of parts for diverse structures. *Automation in Construction*, *125*, 103614. https://doi.org/10.1016/j.autcon.2021.103614

. Bukauskas, A., Mayencourt, P., Shepherd, P., Sharma, B., Mueller, C., Walker, P., & Bregulla, J. (2019). Whole timber construction: A state of the art review. *Construction and Building Materials*, *213*, 748–769. https://doi.org/10.1016/j.conbuildmat.2019.03.043

. Bukauskas, A., Shepherd, P., Walker, P., Sharma, B., & Bregula, J. (2017). Form-Fitting Strategies for Diversity-Tolerant design. *Interfaces: Architecture, Engineering, Science, Annual Meeting of the International Association of Shell & Spatial Structures (IASS), Hamburg, 25-27 September 2017*, 1. https://purehost.bath.ac.uk/ws/files/157995605/hamburg_preprint5.pdf

. Clark, A., Taras, M. A., Schroeder, J. G., & USDA. (1974). *PREDICTED GREEN LUMBER AND RESIDUE YIELDS FROM THE MERCHANTABLE STEM OF YELLOVV-POPLAR*. https://www.srs.fs.usda.gov/pubs/rp/rp_se119.pdf

. Designing Buildings. (2023, April 5). *Charring rate*. https://www.designingbuildings.co.uk/wiki/Charring_rate

. European Committee for Standardization. (2002). *EN 1991-1-1:2002 Eurocode 1: Actions on structures—Part 1-1: General actions—Densities, self-weight, imposed loads for buildings*. CEN.

. European Committee for Standardization. (2003). *EN 1991-1-3:2003 Eurocode 1: Actions on structures—Part 1-3: General actions—Snow loads*. CEN.

. European Committee for Standardization. (2005). *EN 1991-1-4:2005 Eurocode 1: Actions on structures—Part 1-4: General actions—Wind actions*. CEN.

. Hematabadi, H., Madhoushi, M., Khazaeian, A., & Ebrahimi, G. (2021). Structural performance of hybrid Poplar-Beech cross-laminated-timber (CLT). *Journal of Building Engineering*, *44*, 102959. https://doi.org/10.1016/j.jobe.2021.102959

. Hemmati, M., Messadi, T., Gu, H., Seddelmeyer, J., & Hemmati, M. (2024). Comparison of Embodied Carbon Footprint of a Mass Timber Building Structure with a Steel Equivalent. *Buildings*, *14*(5), 1276. https://doi.org/10.3390/buildings14051276

. Hinostroza, I., Pradenas, L., & Parada, V. (2013). Board cutting from logs: Optimal and heuristic approaches for the problem of packing rectangles in a circle. *International Journal of Production Economics*, *145*(2), 541–546. https://doi.org/10.1016/j.ijpe.2013.04.047

. *Houtinfo - houtsoorten*. (2019). Houtinfo.nl. https://houtinfo.nl

. Küpfer, C., Bertola, N., Brütting, J., & Fivet, C. (2021). Decision framework to balance environmental, technical, logistical, and economic criteria when designing structures with reused components. *Frontiers in Sustainability*, *2*. https://doi.org/10.3389/frsus.2021.689877

. Llana, D. F., González-Alegre, V., Portela, M., & Íñiguez-González, G. (2022). Cross Laminated Timber (CLT) manufactured with European oak recovered from demolition: Structural properties and non-destructive evaluation. *Construction and Building Materials*, *339*, 127635. https://doi.org/10.1016/j.conbuildmat.2022.127635

. Negeo, T. S., Rawat, Y. S., & Nebiyu, M. (2024). The effects of sawing methods on the lumber recovery rate and lumber grading of Eucalyptus globulus at the small-scale sawmill enterprise, Addis Ababa, Ethiopia. *Journal of the Indian Academy of Wood Science*, *21*(2), 345–362. https://doi.org/10.1007/s13196-024-00353-2

. Ramage, M. H., Burridge, H., Busse-Wicher, M., Fereday, G., Reynolds, T., Shah, D. U., Wu, G., Yu, L., Fleming, P., Densley-Tingley, D., Allwood, J., Dupree, P., Linden, P., & Scherman, O. (2016). The wood from the trees: The use of timber in construction. *Renewable and Sustainable Energy Reviews*, *68*, 333–359. https://doi.org/10.1016/j.rser.2016.09.107

. *Tabel Loofhout*. (n.d.). Houtinfobois. https://www.houtinfobois.be/wp-content/uploads/2015/08/Tabel-loofhout.pdf

. Von Buelow, Peter & Oliyan Torghabehi, Omid & Mankouche, Steven & Vliet, Kasey. (2018). Combining parametric form generation and design exploration to produce a wooden reticulated shell using natural tree crotches.

. Van Marcke, A., Laghi, V., & Carstensen, J. V. (2024). Automated planar truss design with reclaimed partially disassembled steel truss components. *Journal of Building Engineering*, *84*, 108458. https://doi.org/10.1016/j.jobe.2024.108458

. Warmuth, J., D'Acunto, P., & Fivet, C. (2024). Shaping Structures from Reclaimed Elements A Computational Framework for Stock-Constrained Design of Static Equilibrium. In *Scalable Disruptors* (pp. 42–56). https://doi.org/10.1007/978-3-031-68275-9_4

.	Warmuth, J., Brütting, J., & Fivet, C. (2021). Computational tool for stock-constrained design of structures. *Infoscience (Ecole Polytechnique Fédérale De Lausanne)*, 1–9. https://infoscience.epfl.ch/handle/20.500.14299/180799

.	Zachary, N. M., & Martin, N. S. (2016). *Advances in Architectural Geometry 2015 - Tree Fork Truss: geometric Strategies for Exploiting Inherent Material Form*. https://doi.org/10.3218/3778-4_11

.Tree specie images

.Robinia
https://meye.dk/wp-content/uploads/2020/08/meye_robinia-pseudoacacia_S8559.png

.Oak
https://www.pngall.com/wp-content/uploads/5/Green-Oak-Tree-PNG-Picture.png

.Ash
https://png.pngtree.com/png-clipart/20230524/ourmid/pngtree-shady-ash-trees-side-view-a-png-image_7108084.png

.Beech
https://static.vecteezy.com/system/resources/thumbnails/045/385/022/small_2x/beech-tree-for-architecture-visualization-png.png

.Elm
https://yimages360.yellowimages.com/products/mdi/031/M0MwEMTyBV/1x.jpg

.Maple
https://e7.pngegg.com/pngimages/876/970/png-clipart-european-horse-chestnut-tree-branch-sweet-chestnut-maple-tree-maple-leaf-thumbnail.png

.Chestnut
https://static.vecteezy.com/system/resources/previews/010/832/715/non_2x/3-season-set-of-chestnut-tree-transparent-background-free-png.png

.Plane
https://t3.ftcdn.net/jpg/05/78/78/60/360_F_578786066_uGhsfVAHO11W0NcyymjS7BiB3JBcquDq.jpg

.Willow
https://i.pinimg.com/736x/68/16/c7/6816c73088fdbb59dde53a2c684c962e.jpg

.Poplar
https://t3.ftcdn.net/jpg/06/14/91/62/360_F_614916259_OI3t3vbOVoWgNEN0rwZT14GdtfEIJMQs.jpg

.ShortURLs

***1** https://shorturl.at/8ul72 = https://www.arcgis.com/apps/mapviewer/index.html?webmap=986eda8d3c714e6cacbc6070be1bff1f

***2** https://shorturl.at/VSoB4 = https://www.royaloakstair.ca/blog/stairs-rails/what-is-the-difference-between-quarter-sawn-vs-plain-sawn-vs-rift-sawn-wood/

***3** https://shorturl.at/RRrBh = https://sahateollisuuskirja.fi/wp-content/uploads/esaha_log_patterns.jpg

***4** https://shorturl.at/QLrK9 = https://www.usnr.com/assets/images/SplitDetection-HDR.jpg

***5** https://shorturl.at/bQZn2 = https://static.dezeen.com/uploads/2022/10/floating-office-rotterdam-powerhouse-company_dezeen_2364_col_23-1704x1136.jpg

.Chapter banner images

Chapter 1: Introduction =
https://unsplash.com/photos/brown-tree-bark-in-closeup-photography-mI-QcAP95Ok, retrieved January 2026

Chapter 2: Background =
Image generated by ChatGPT, based on the prompt: ‘’Make an Oak Tree Bark texture similar to this one: https://unsplash.com/photos/brown-tree-bark-in-closeup-photography-mI-QcAP95Ok’’

OpenAI. (2026). *ChatGPT* (GPT-5.2) [AI image generation]. https://chat.openai.com/

Chapter 3: Methodology =
https://www.freepik.com/free-photo/wooden-background_21071219.htm, retrieved January 2026

Chapter 4: Case Study =
Photo made by author

Chapter 5: Results =
Photo made by author

Chapter 6: Discussion=
https://www.freepik.com/free-photo/closeup-bark-texture_3077314.htm, retrieved January 2026

Chapter 7:Conclusion & Recommendations =
Photo made by author

Chapter 8: Reflection =
https://www.freepik.com/free-photo/top-view-tree-bark_11768197.htm, retrieved January 2026

/Appendix

Formulaic conversation for loadbearing calculations

Buckling

For the buckling calculations all column elements are regarded as pin ended struts meaning L_e=L. The standard Euler Buckling formula for perfect axial loading is:

$$Pcrit \ = \ \frac{(\pi^2 EI)}{(Le)^2}$$

Since loads are rarely perfectly axial a factor of 5 is added to the P*crit*.

$$Pcrit \ = \ \frac{(\pi^2 EI)}{5(Le)^2}$$

This can be rewritten as:

$$I = \frac{Pcrit \ * \ 5 * Le^2}{\pi^2 E}$$

The formula for Moment of Inertia (I) for rectangular shapes is:

$$Ix = \frac{bh^3}{12}$$

$$Iy = \frac{hb^3}{12}$$

In order to allow for different ratio's in rectangular profiles b becomes a parameter of h, where x is the ratio of width to height:

$$b = \frac{h}{x} \qquad h = bx$$

Since h will be a factor of b it can be said that for every rectangle h≥b. This means I_y will be the smaller value for the moment of inertia and thereby governing.

$$Iy = \frac{bx * b^3}{12}$$

$$b = \sqrt[4]{\frac{12Iy}{x}}$$

Compression strength

The compression in the columns must not exceed the characteristic compressive strength.

$$\sigma_c = \frac{F}{A} \leq f_{c,0,d}$$

Since the loads and the characteristic strength of each timber type are given the formula can be rewritten as:

$$\frac{F}{f_{c,0,d}} \leq A$$

The area (A) that comes from this formula is divided into five possible configurations similar to as in buckling. Parameter x is the ratio in which h scales to b.

$$h = xb$$

$$A = hb$$

$$A = xb^2$$

$$b = \sqrt[2]{\frac{A}{x}}$$

Bending Moment

σ_m is material the property per tree species, see chapter 4. M is the moment for each element, as calculated by the Karamba software.

$$\sigma_m = \frac{M}{W}$$

$$W = \frac{M}{\sigma_m}$$

Required section modulus *W*.

$$W = \frac{bh^2}{6}$$

Again ratio parameter x is added to this formula to allow for simple different dimensions.

$$b = \frac{h}{x}$$

$$W = \frac{h^3}{6x}$$

$$h = \sqrt[3]{6Wx}$$

Longitudinal shear equation

$$\tau = \frac{S\,A_c\,\bar{y}}{I\,b}$$

The formula for Moment of Inertia (I) for rectangular shapes is:

$$I = \frac{bh^3}{12}$$

For a rectangular profile the values for A_c and $\bar{y}$ can be assumed as:

$$A_c = b\left(\frac{h}{2} - y\right)$$

$$\bar{y} = \frac{1}{2}\left(\frac{h}{2} - y\right)$$

At the neutral axis the shear force is the highest meaning τ_{max} is at y = 0. Substituting A_c and $\bar{y}$ at level y = 0 together with I gives:

$$\tau_{\max} = \frac{S\,b\left(\frac{h}{2}\right)\frac{1}{2}\left(\frac{h}{2}\right)}{\frac{bh^3}{12}\,b}$$

$$\tau_{\max} = \frac{3S}{2hb}$$

The value for $\tau_{\max}$ should not exceed the characteristic shear strength of each timber type. Meaning $f_{v,k} \geq \tau_{\max}$. The shear force *S* is the result of the loads as calculated by Karamba. This is allows for the equation to be:

$$hb \geq \frac{3S}{2f_{v,k}}$$

Adding in ratio x as a substitute to *h* makes:

$$h = bx$$

$$b \geq \sqrt[2]{\frac{3S}{2f_{v,k}x}}$$

Pseudocode algorithm 1, 2 & 3

ALGORITHM 1: GreedyPack (open-bins first, then open a new bin)

```
INPUT:
  - order: permutation of element-group IDs
  - bin_order: permutation of bin IDs
  - G: element dimension groups with species variants
  - C: bins (height, radius, species)
OUTPUT:
  - Packing result R (placements + metrics)
opened ← ∅
placements ← ∅
FOR each element id g in order DO
    e ← InstantiateElementFromGroup(G[g])          // choose/enable its allowed variants
    placed ← FALSE
    // Two-stage candidate search: prefer already-open bins, else open a new one
    open_candidates ← {b ∈ opened | Compatible(b,e)}          //where b is a single bin
    new_candidates  ← {b ∈ bins \ opened | Compatible(b,e)}
    FOR candidate_list in [open_candidates, new_candidates] DO
        FOR each bin b in candidate_list DO
            IF TryPlaceElement*alg2(b, e, cfg) THEN
                placed ← TRUE
                opened ← opened ∪ {b}
                placements ← placements ∪ {Placement(b,e)}
                BREAK
            END IF
        END FOR
        IF placed THEN BREAK
    END FOR
END FOR
packed_total ← CountPlaced(placements)
util_sum ← Σ Utilization(b) over bins with ≥1 placement
bins_used ← CountBinsWithPlacements(placements)
return R = (placements, packed_total, util_sum, bins_used)
```

ALGORITHM 2: TryPlaceElement(bin b, element e, cfg)

```
INPUT:
  - b: one bin with partitioned free-space representation
  - e: element with feasible variants (L,W,H) depending on species and rules
  - cfg: configuration settings, like i.e. kerf size and partition priority
OUTPUT:
  - TRUE if e placed in b, else FALSE
V ← FeasibleVariants(e, species = b.species)
IF V = ∅ THEN return FALSE
Sort(V by footprint area ascending)                 // try smaller footprints first
P ← OrderedPartitions(cfg.partition_order)          // e.g., edges-first or center-first
best ← NONE
FOR each partition p in P DO
    MergeVerticallyAlignedFreeBoxes(p.free_boxes)
    FOR each variant v in V DO
        IF v.H > b.height THEN CONTINUE
        FOR each orientation (L,W) in {(v.L,v.W),(v.W,v.L)} DO
            (Lk,Wk) ← AddKerfAndClearance(L,W,cfg)
            FOR each free box FB in p.free_boxes DO
                IF Fits(FB, Lk, Wk, v.H) THEN
                    score ← PlacementPreferenceScore(FB, cfg)
                    IF best = NONE OR score < best.score THEN
                        best ← (p, FB, v, Lk, Wk, score)
                    END IF
                END IF
            END FOR
        END FOR
    END FOR
END FOR
IF best = NONE THEN return FALSE
CommitPlacement(b, e, best)                         // store geometry & update max height
Remove(best.FB from best.p.free_boxes)
new_boxes ← GuillotineSplit(best.FB, best.Lk, best.Wk, best.v.H)
Add(new_boxes to best.p.free_boxes)
return TRUE
```

ALGORITHM 3: GeneticAlgorithm(G, C, cfg, seed)

```
ENCODING:  Individual ind = (order, bin_order)  // both are permutations
FITNESS:  score =  100·util_sum − 30·bins_used
// Generation 0: initialize population (random permutations) + optional heuristic seed
population ← Ø
IF cfg.use_seed THEN
    Add(population, seed)
END IF
WHILE |population| < cfg.pop_size DO
    order_rand ← RandomPermutation(1..|G|)
    bin_order_rand ← RandomPermutation(1..|C|)
    Add(population, (order_rand, bin_order_rand))
END WHILE

WHILE |population| < cfg.pop_size DO
    order_rand ← RandomPermutation(1..|G|)
    bin_order_rand ← RandomPermutation(1..|C|)
    Add(population, (order_rand, bin_order_rand))
END WHILE

best_global ← NONE
FOR gen = 1..cfg.generations DO
    scored ← empty list
    FOR each individual ind in population DO
        R ← GreedyPack(ind.order, ind.bin_order, G, C, cfg)
        s ← Fitness(R)
        scored ← scored append (s, ind)
        IF best_global is NONE OR s > best_global.score THEN
            best_global ← (s, individual)
        END IF
    END FOR

    Sort(scored by score descending)
    new_pop ←  TakeElite(scored, cfg.elite_count)

    WHILE |new_pop| < cfg.pop_size DO
        p1 ← TournamentSelect(scored, cfg.tournament_size)
        p2 ← TournamentSelect(scored, cfg.tournament_size)

        child.order ← OrderCrossoverOX(p1.order, p2.order)
        child.bin_order ← OrderCrossoverOX(p1.bin_order, p2.bin_order)

        MutateSwap(child.order, cfg.mutation_rate)
        MutateInversion(child.order, cfg.mutation_rate)
        MutateSwap(child.bin_order, cfg.mutation_rate)
        MutateInversion(child.bin_order, cfg.mutation_rate)

        new_pop ← new_pop append child
    END WHILE
    population ← new_pop
END FOR
return best_individual
```

Tree Index – Rotterdam Wood

Table 11: the amount & dimensions for each tree log used in the urban wood database.

ID	Species	Diameter	Length
0	Robinia	0.59	6.63
1	Robinia	0.45	4.42
2	Robinia	0.7	3
3	Robinia	0.56	7.91
4	Robinia	0.7	2.65
5	Robinia	0.59	3.84
6	Robinia	0.5	6.81
7	Robinia	0.45	4.67
8	Robinia	0.52	3.35
9	Robinia	0.3	2.07
10	Robinia	0.54	6.59
11	Robinia	0.68	2.17
12	Robinia	0.56	2.05
13	Robinia	0.47	5.06
14	Robinia	0.6	4.29
15	Robinia	0.57	3.68
16	Robinia	0.5	6.22
17	Robinia	0.52	3.41
18	Robinia	0.59	6.39
19	Robinia	0.43	2.53
20	Robinia	0.41	2.5
21	Robinia	0.5	4.03
22	Robinia	0.67	3.98
23	Robinia	0.6	6.39
24	Robinia	0.54	7.38
25	Robinia	0.51	6.88
26	Robinia	0.61	5.92
27	Robinia	0.52	7.56
28	Robinia	0.6	2.6
29	Robinia	0.61	6.15
30	Robinia	0.52	7.55
31	Robinia	0.57	7.88
32	Robinia	0.51	2.61
33	Robinia	0.69	2.23
34	Robinia	0.54	4.05
35	Robinia	0.6	7.1
36	Robinia	0.62	6.12
37	Robinia	0.49	6.46
38	Robinia	0.5	6.11
39	Robinia	0.43	2.56
40	Robinia	0.71	2.24
41	Robinia	0.45	3.63
42	Robinia	0.42	3.58
43	Robinia	0.64	5

44	Robinia	0.57	7.77
45	Robinia	0.63	5.34
46	Robinia	0.38	2.11
47	Robinia	0.46	4.93
48	Robinia	0.54	4.05
49	Robinia	0.65	4.91
50	Robinia	0.61	5.85
51	Robinia	0.48	5.72
52	Robinia	0.74	2.44
53	Robinia	0.54	7.37
54	Robinia	0.48	5.57
55	Robinia	0.45	4.59
56	Oak	0.7	8.96
57	Oak	0.51	7.46
58	Oak	0.86	6.48
59	Oak	0.66	9.84
60	Oak	0.86	6.25
61	Oak	0.71	7.06
62	Oak	0.59	9.09
63	Oak	0.51	7.63
64	Oak	0.61	6.72
65	Ash	0.65	10.32
66	Ash	0.48	8.74
67	Ash	0.78	7.71
68	Ash	0.61	11.24
69	Ash	0.78	7.47
70	Ash	0.65	8.32
71	Ash	0.54	10.45
72	Ash	0.48	8.92
73	Ash	0.57	7.97
74	Ash	0.3	7.05
75	Ash	0.58	10.29
76	Ash	0.75	7.12
77	Ash	0.61	7.03
78	Ash	0.5	9.19
79	Ash	0.66	8.64
80	Ash	0.63	8.2
81	Ash	0.54	10.02
82	Ash	0.56	8.01
83	Ash	0.64	10.14
84	Ash	0.46	7.38
85	Ash	0.43	7.36
86	Ash	0.54	8.46
87	Ash	0.75	8.42
88	Ash	0.65	10.14
89	Ash	0.59	10.85
90	Ash	0.55	10.49
91	Ash	0.68	9.81

92	Ash	0.57	10.99
93	Ash	0.66	7.43
94	Ash	0.67	9.97
95	Ash	0.57	10.98
96	Ash	0.62	11.21
97	Ash	0.55	7.44
98	Ash	0.76	7.17
99	Ash	0.59	8.47
100	Ash	0.66	10.65
101	Ash	0.69	9.95
102	Ash	0.52	10.2
103	Ash	0.54	9.95
104	Ash	0.46	7.4
105	Ash	0.79	7.17
106	Ash	0.48	8.17
107	Ash	0.45	8.13
108	Ash	0.71	9.15
109	Ash	0.63	11.13
110	Ash	0.7	9.39
111	Ash	0.39	7.08
112	Ash	0.49	9.1
113	Ash	0.58	8.47
114	Ash	0.72	9.09
115	Ash	0.67	9.76
116	Ash	0.51	9.67
117	Ash	0.83	7.32
118	Ash	0.59	10.85
119	Ash	0.52	9.56
120	Ash	0.47	8.86
121	Ash	0.39	7.32
122	Ash	0.66	8.87
123	Ash	0.39	7.38
124	Ash	0.45	8.27
125	Ash	0.58	9.81
126	Ash	0.62	10.76
127	Ash	0.6	8.49
128	Ash	0.64	9.29
129	Ash	0.79	7.37
130	Ash	0.57	10.38
131	Ash	0.56	10.99
132	Ash	0.58	11.17
133	Beech	0.59	8.57
134	Elm	0.39	10.18
135	Elm	0.25	7.91
136	Elm	0.5	6.43
137	Elm	0.36	11.51
138	Elm	0.5	6.08
139	Elm	0.39	7.3

140	Elm	0.3	10.37
141	Elm	0.25	8.16
142	Elm	0.32	6.79
143	Elm	0.1	5.47
144	Elm	0.34	10.15
145	Elm	0.48	5.58
146	Elm	0.36	5.45
147	Elm	0.27	8.56
148	Elm	0.4	7.77
149	Elm	0.37	7.13
150	Elm	0.3	9.76
151	Maple	0.53	9.05
152	Maple	0.42	7.7
153	Maple	0.62	6.81
154	Maple	0.51	9.84
155	Maple	0.62	6.6
156	Maple	0.53	7.33
157	Maple	0.46	9.17
158	Maple	0.42	7.85
159	Maple	0.48	7.03
160	Maple	0.3	6.24
161	Maple	0.49	9.03
162	Maple	0.6	6.31
163	Maple	0.51	6.23
164	Maple	0.43	8.09
165	Maple	0.54	7.61
166	Maple	0.52	7.23
167	Maple	0.46	8.8
168	Maple	0.48	7.07
169	Maple	0.53	8.91
170	Maple	0.4	6.53
171	Maple	0.39	6.51
172	Maple	0.46	7.45
173	Maple	0.6	7.42
174	Maple	0.54	8.91
175	Maple	0.49	9.52
176	Maple	0.47	9.21
177	Maple	0.55	8.62
178	Maple	0.48	9.63
179	Maple	0.54	6.57
180	Maple	0.55	8.76
181	Maple	0.48	9.62
182	Maple	0.51	9.83
183	Maple	0.46	6.58
184	Maple	0.61	6.34
185	Maple	0.49	7.46
186	Maple	0.54	9.34
187	Maple	0.56	8.74

188	Maple	0.45	8.95
189	Maple	0.46	8.74
190	Maple	0.4	6.54
191	Maple	0.63	6.35
192	Maple	0.42	7.2
193	Maple	0.4	7.17
194	Maple	0.58	8.05
195	Maple	0.52	9.76
196	Maple	0.56	8.26
197	Maple	0.36	6.27
198	Maple	0.42	8.01
199	Maple	0.49	7.47
200	Maple	0.58	8
201	Maple	0.55	8.58
202	Maple	0.44	8.49
203	Maple	0.65	6.47
204	Maple	0.49	9.51
205	Maple	0.45	8.4
206	Maple	0.42	7.8
207	Maple	0.36	6.47
208	Maple	0.54	7.81
209	Maple	0.36	6.53
210	Maple	0.4	7.3
211	Maple	0.48	8.62
212	Maple	0.51	9.44
213	Maple	0.5	7.48
214	Maple	0.53	8.17
215	Maple	0.63	6.52
216	Maple	0.48	9.11
217	Maple	0.48	9.63
218	Maple	0.49	9.78
219	Maple	0.58	6.83
220	Maple	0.54	8.17
221	Chestnut	0.53	7.99
222	Chestnut	0.42	6.38
223	Chestnut	0.62	5.33
224	Chestnut	0.51	8.93
225	Poplar	0.59	8.17
226	Poplar	0.45	5.23
227	Poplar	0.7	3.33
228	Poplar	0.56	9.88
229	Poplar	0.7	2.87
230	Poplar	0.59	4.45
231	Poplar	0.5	8.42
232	Poplar	0.45	5.56
233	Poplar	0.52	3.8
234	Poplar	0.3	2.09
235	Poplar	0.54	8.12

236	Poplar	0.68	2.23
237	Poplar	0.56	2.06
238	Poplar	0.47	6.08
239	Poplar	0.6	5.06
240	Poplar	0.57	4.24
241	Poplar	0.5	7.62
242	Poplar	0.52	3.88
243	Poplar	0.59	7.85
244	Poplar	0.43	2.71
245	Poplar	0.41	2.66
246	Poplar	0.5	4.71
247	Poplar	0.67	4.64
248	Poplar	0.6	7.85
249	Poplar	0.54	9.17
250	Poplar	0.51	8.5
251	Poplar	0.61	7.23
252	Poplar	0.52	9.41
253	Poplar	0.6	2.79
254	Poplar	0.61	7.53
255	Poplar	0.52	9.4
256	Poplar	0.57	9.84
257	Poplar	0.51	2.81
258	Poplar	0.69	2.31
259	Poplar	0.54	4.73
260	Poplar	0.6	8.8
261	Poplar	0.62	7.5
262	Poplar	0.49	7.95
263	Poplar	0.5	7.49
264	Poplar	0.43	2.74
265	Poplar	0.71	2.32
266	Poplar	0.45	4.17
267	Poplar	0.42	4.11
268	Poplar	0.64	6
269	Poplar	0.57	9.69

Average dimensions for timber elements:

4x4		
	Average Width x Height [cm]	Average Width x Height [cm]
Wood type	1:1 Column	1:3 Beam
Robinia	17.7x17.7	11.6x10.0
Oak	18.4x18.4	12.3x10.3
Ash	18.4x18.4	12.0x10.2
Beech	18.2x18.2	11.9x10.1
Elm	18.8x18.8	12.6x10.4
Maple	18.4x18.4	11.9x10.1
Chestnut	19.3x19.3	13.3x10.6
Poplar	19.3x19.3	13.6x10.7

Average dimensions for timber elements in 4x4 structure.

8x8		
	Average Width x Height [cm]	Average Width x Height [cm]
Wood type	1:1 Column	1:3 Beam
Robinia	19.5x19.5	15.0x11.2
Oak	20.2x20.2	16.1x11.5
Ash	20.2x20.2	15.6x11.4
Beech	20.0x20.0	15.4x11.3
Elm	20.7x20.7	16.5x11.7
Maple	20.2x20.2	15.4x11.3
Chestnut	21.3x21.3	17.4x12.0
Poplar	21.3x21.3	17.8x12.1

Average dimensions for timber elements in 8x8 structure.

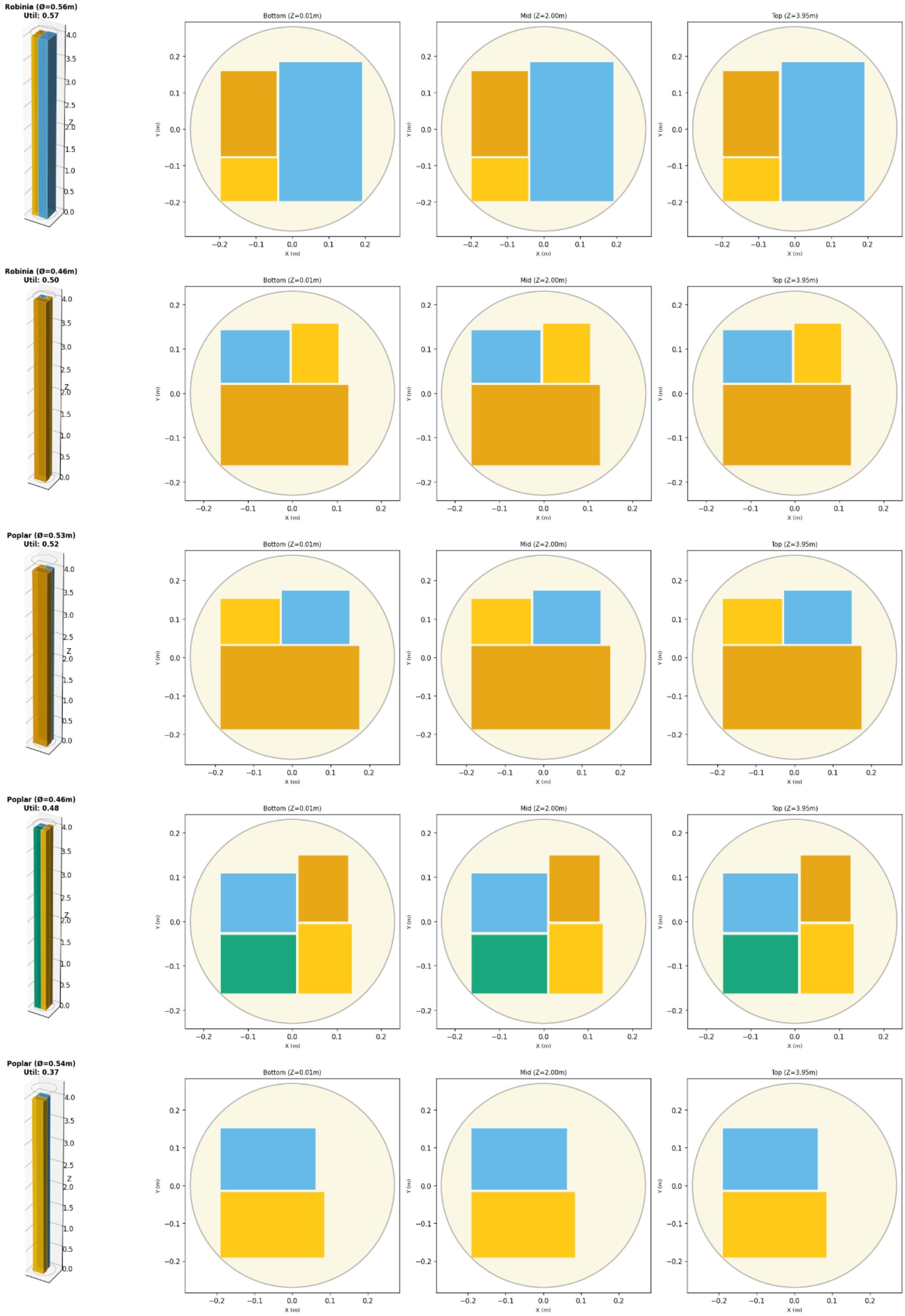
8x8x2		
	Average Width x Height [cm]	Average Width x Height [cm]
Wood type	1:1 Column	1:3 Beam
Robinia	20.0x20.0	14.2x10.9
Oak	20.8x20.8	15.1x11.2
Ash	20.8x20.8	14.6x11.1
Beech	20.5x20.5	14.6x11.0
Elm	21.2x21.2	15.5x11.3
Maple	20.8x20.8	14.6x11.0
Chestnut	21.9x21.9	16.4x11.6
Poplar	21.9x21.9	16.7x11.7

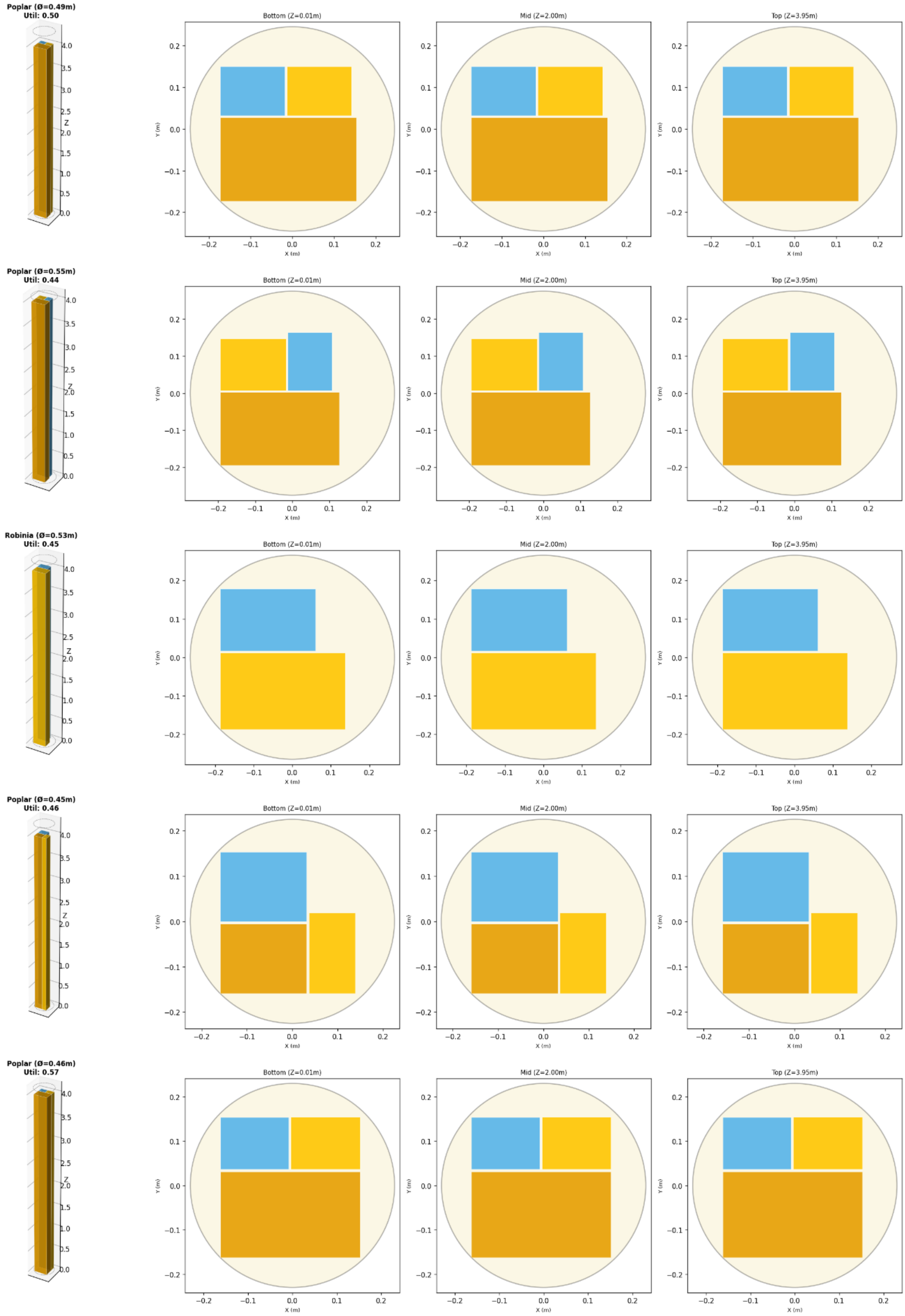
Average dimensions for timber elements in 8x8x2 structure.

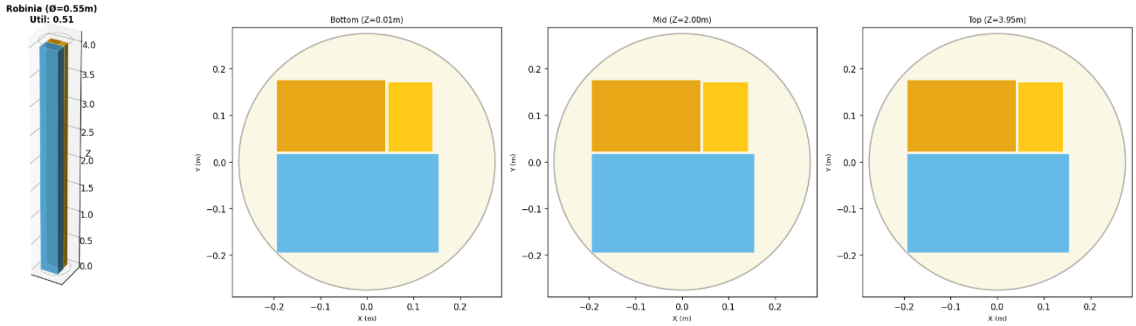
2x8x8x2		
	Average Width x Height [cm]	Average Width x Height [cm]
Wood type	1:1 Column	1:3 Beam
Robinia	20.0x20.0	14.2x10.9
Oak	20.8x20.8	15.1x11.2
Ash	20.8x20.8	14.7x11.0
Beech	20.6x20.6	14.6x11.0
Elm	21.3x21.3	15.5x11.3
Maple	20.8x20.8	14.6x11.0
Chestnut	22.0x22.0	16.4x11.6
Poplar	22.0x22.0	16.8x11.7

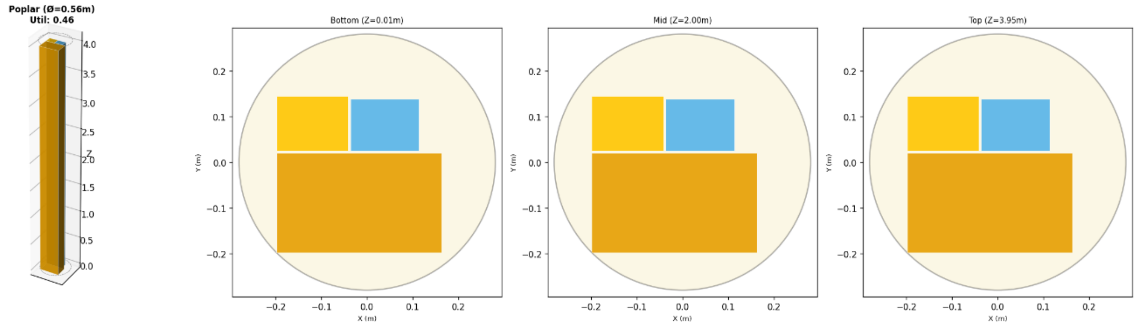
Average dimensions for timber elements in 2x8x8x2 structure.

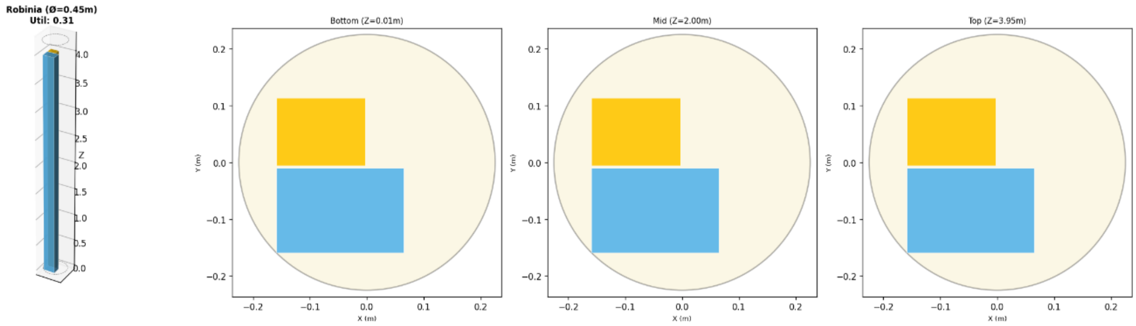
Cross-sections Greedy Heuristic: Best-Fit 8x8x2

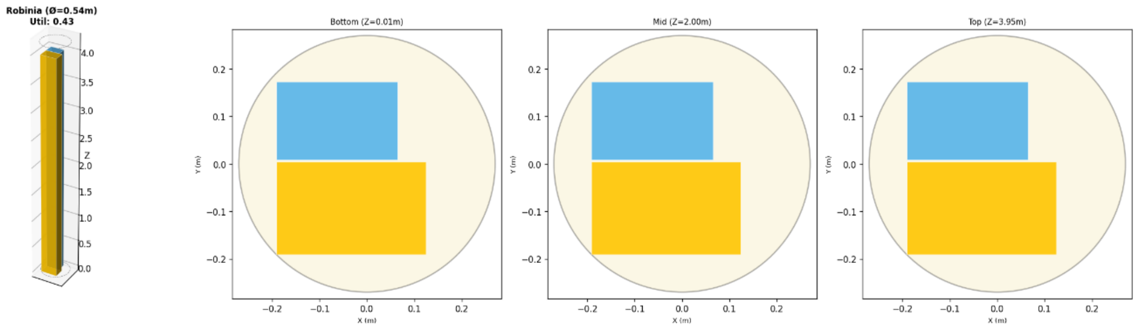


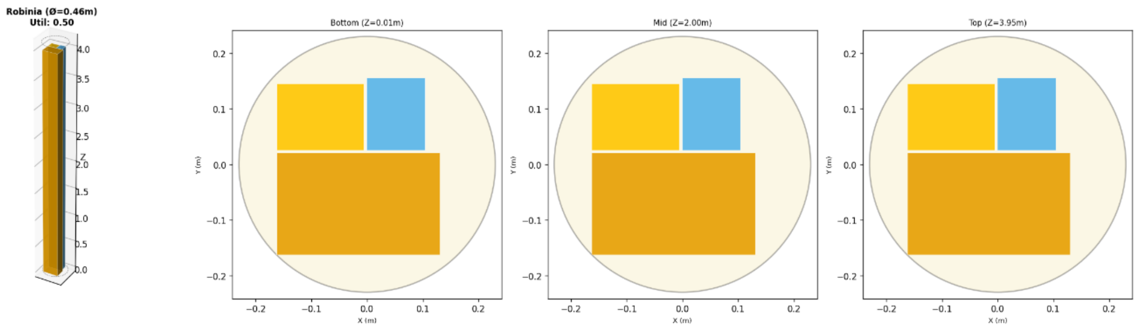

Poplar (Ø=0.49m)
Util: 0.50
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.55m)
Util: 0.44
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Robinia (Ø=0.53m)
Util: 0.45
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.45m)
Util: 0.46
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.46m)
Util: 0.57
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
X (m)
Y (m)
Z


Robinia (Ø=0.55m)
Util: 0.51
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)

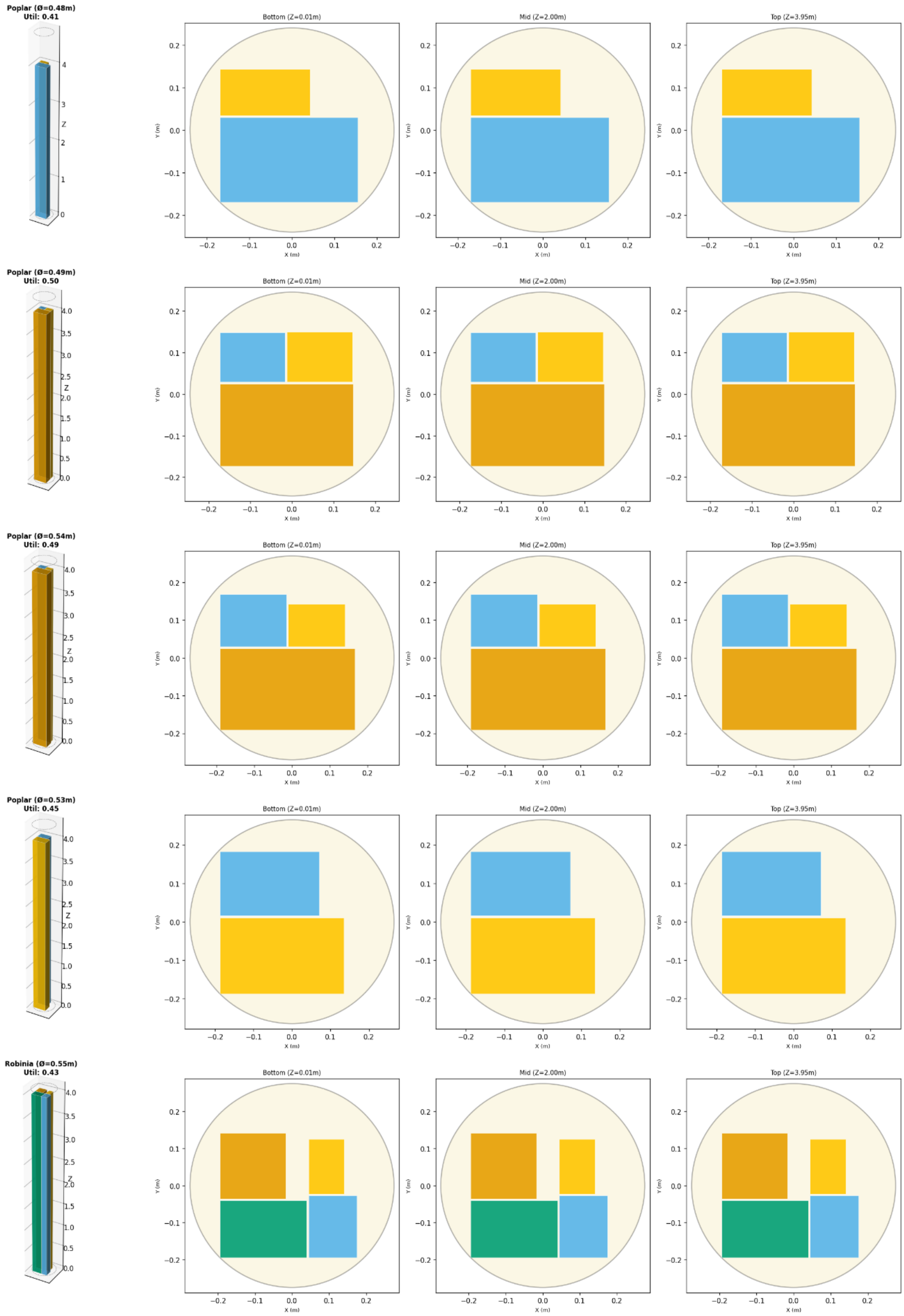

Poplar (Ø=0.56m)
Util: 0.46
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)

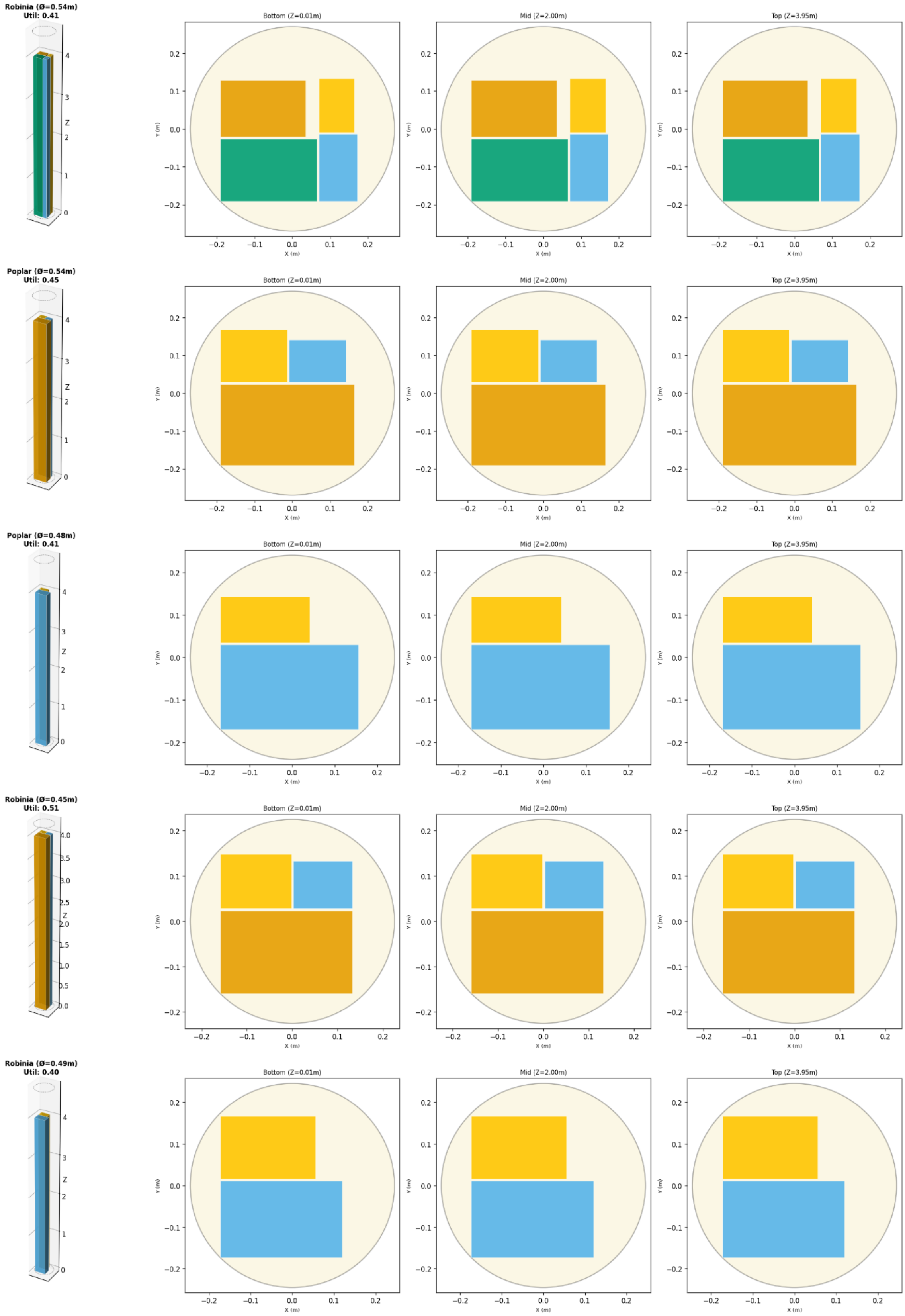

Robinia (Ø=0.45m)
Util: 0.31
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)

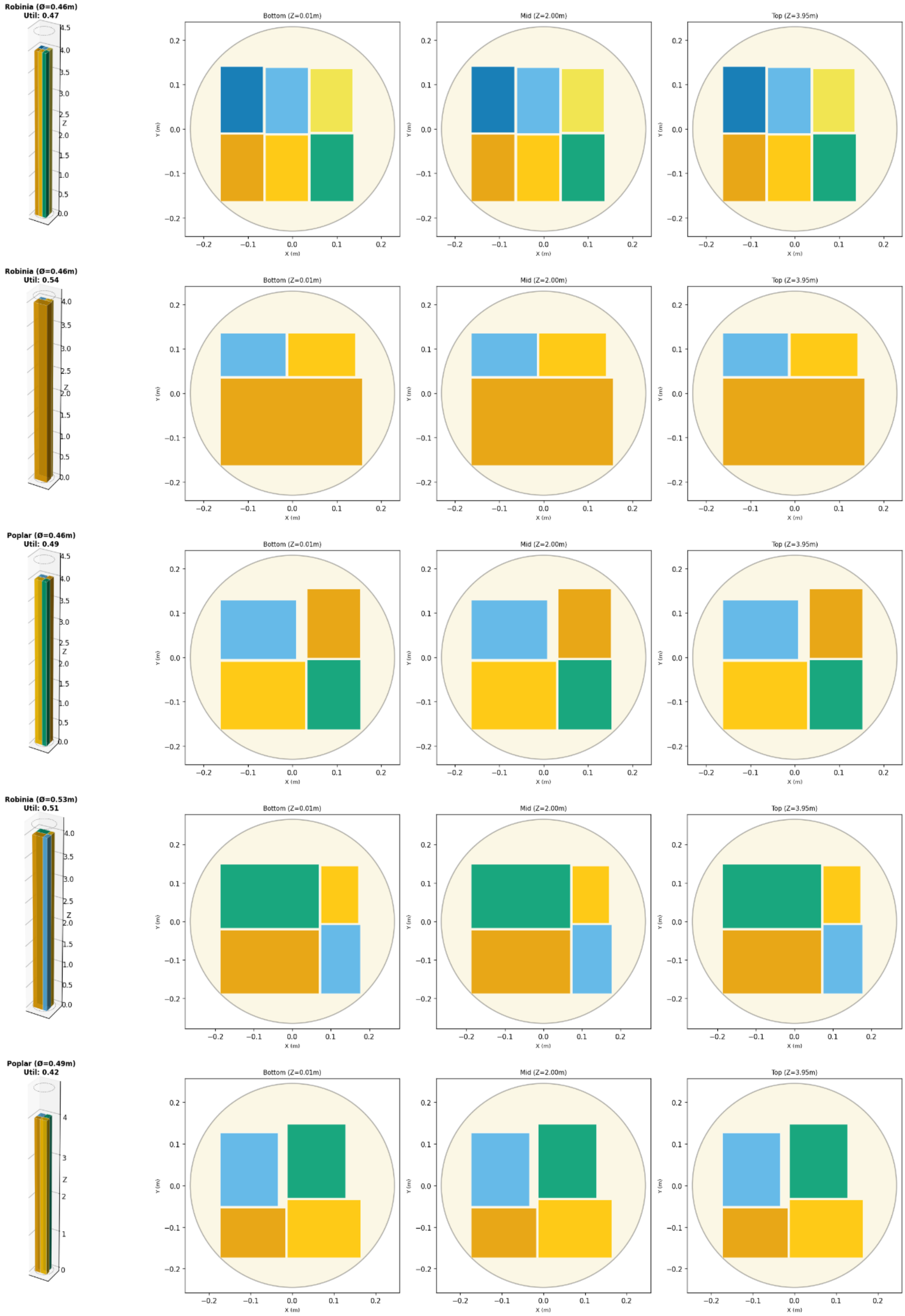

Robinia (Ø=0.54m)
Util: 0.43
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)

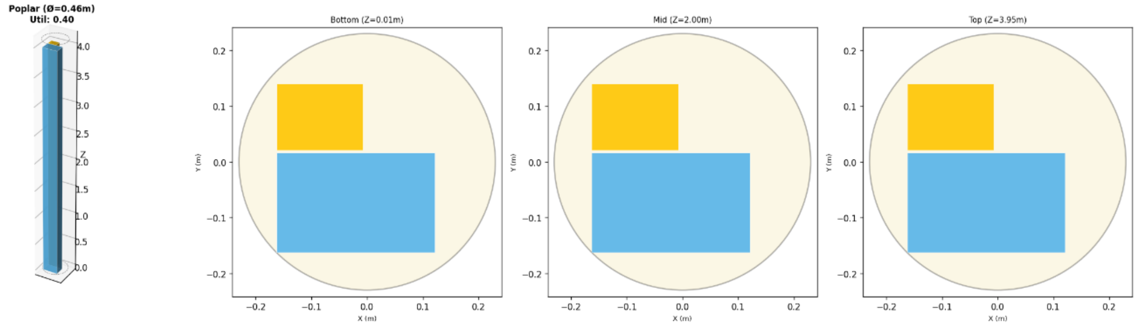

Robinia (Ø=0.46m)
Util: 0.50
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)

Cross-sections Greedy Heuristic: Best-Fit 2x8x8x2

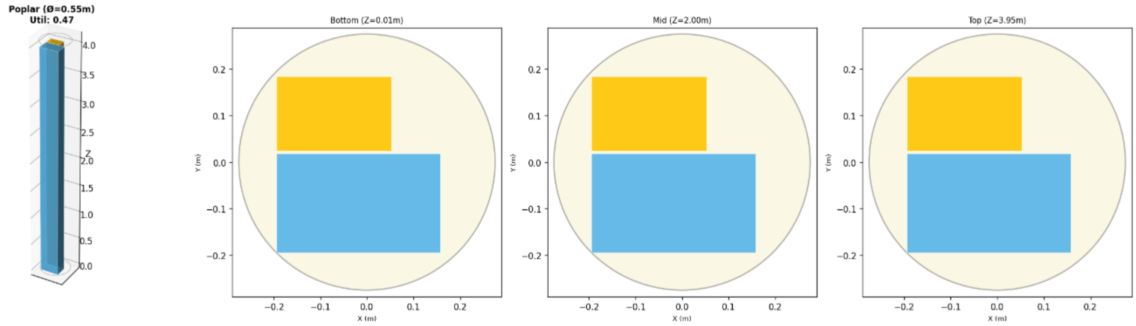



Robinia (Ø=0.54m)
Util: 0.41
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.54m)
Util: 0.45
Poplar (Ø=0.48m)
Util: 0.41
Robinia (Ø=0.45m)
Util: 0.51
Robinia (Ø=0.49m)
Util: 0.40
Z
X (m)
Y (m)

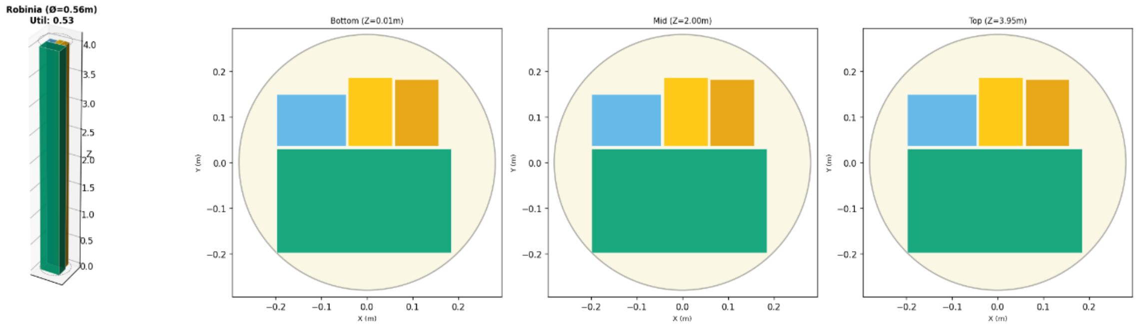

Robinia (Ø=0.46m)
Util: 0.47
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Robinia (Ø=0.46m)
Util: 0.54
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.46m)
Util: 0.49
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Robinia (Ø=0.53m)
Util: 0.51
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.49m)
Util: 0.42
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
X (m)
Y (m)
Z



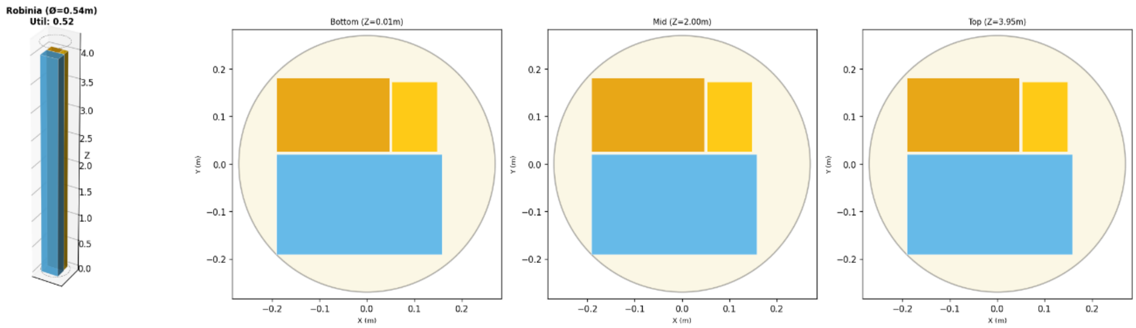
Poplar (Ø=0.46m)
Util: 0.40
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)



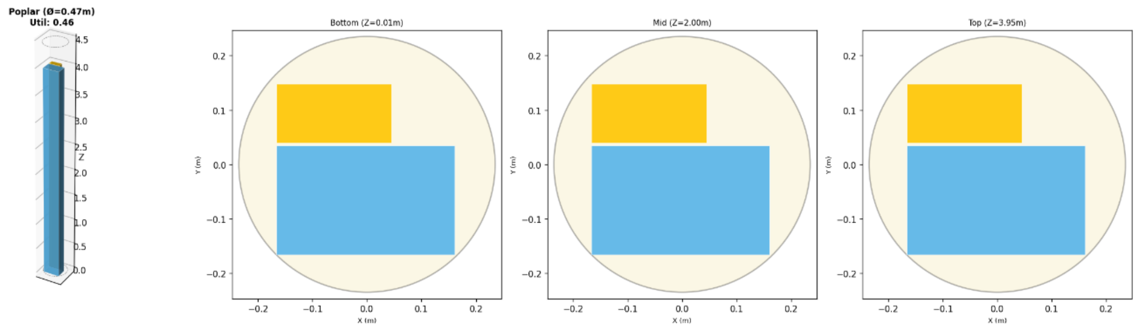
Poplar (Ø=0.55m)
Util: 0.47
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)



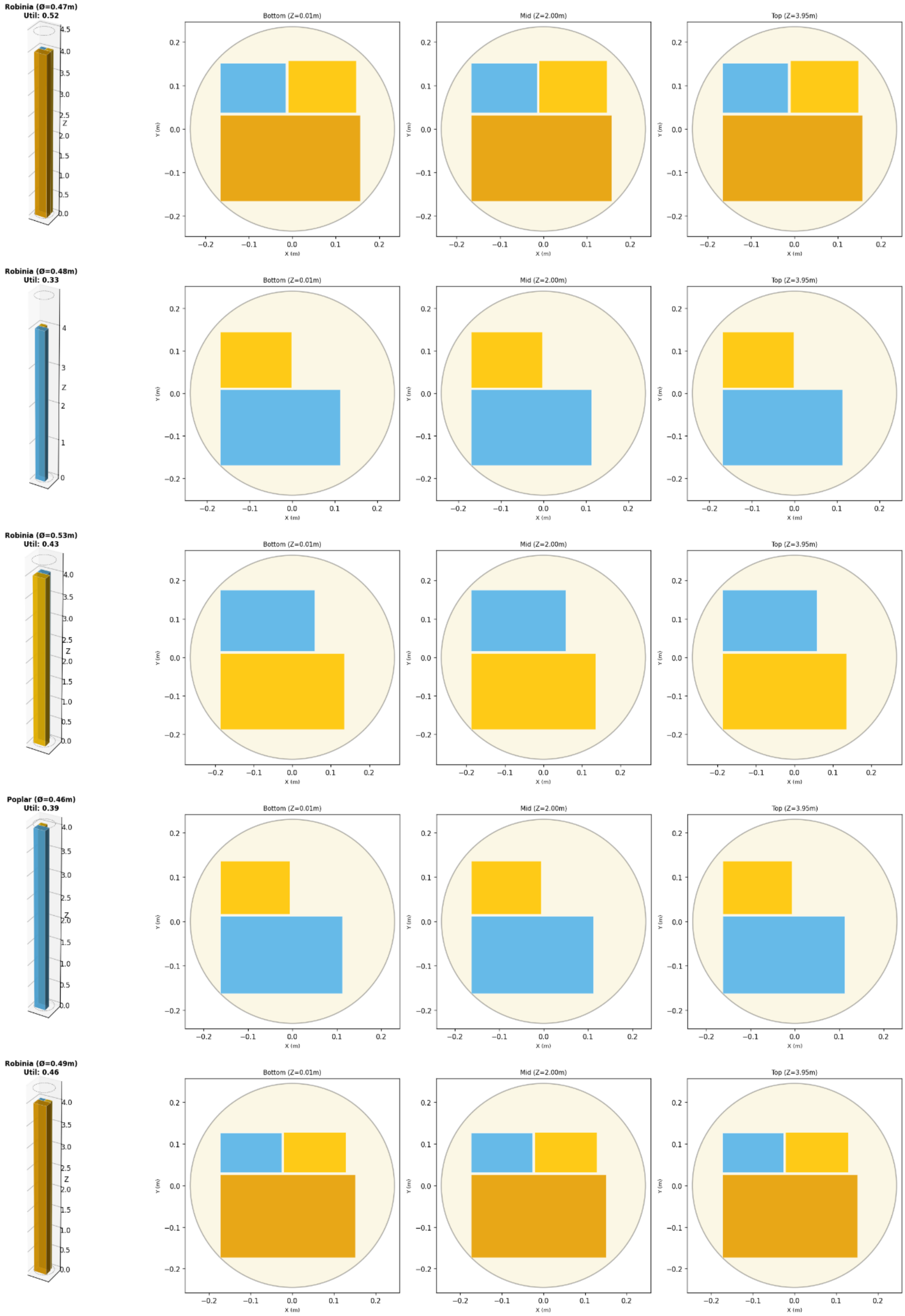
Robinia (Ø=0.56m)
Util: 0.53
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)

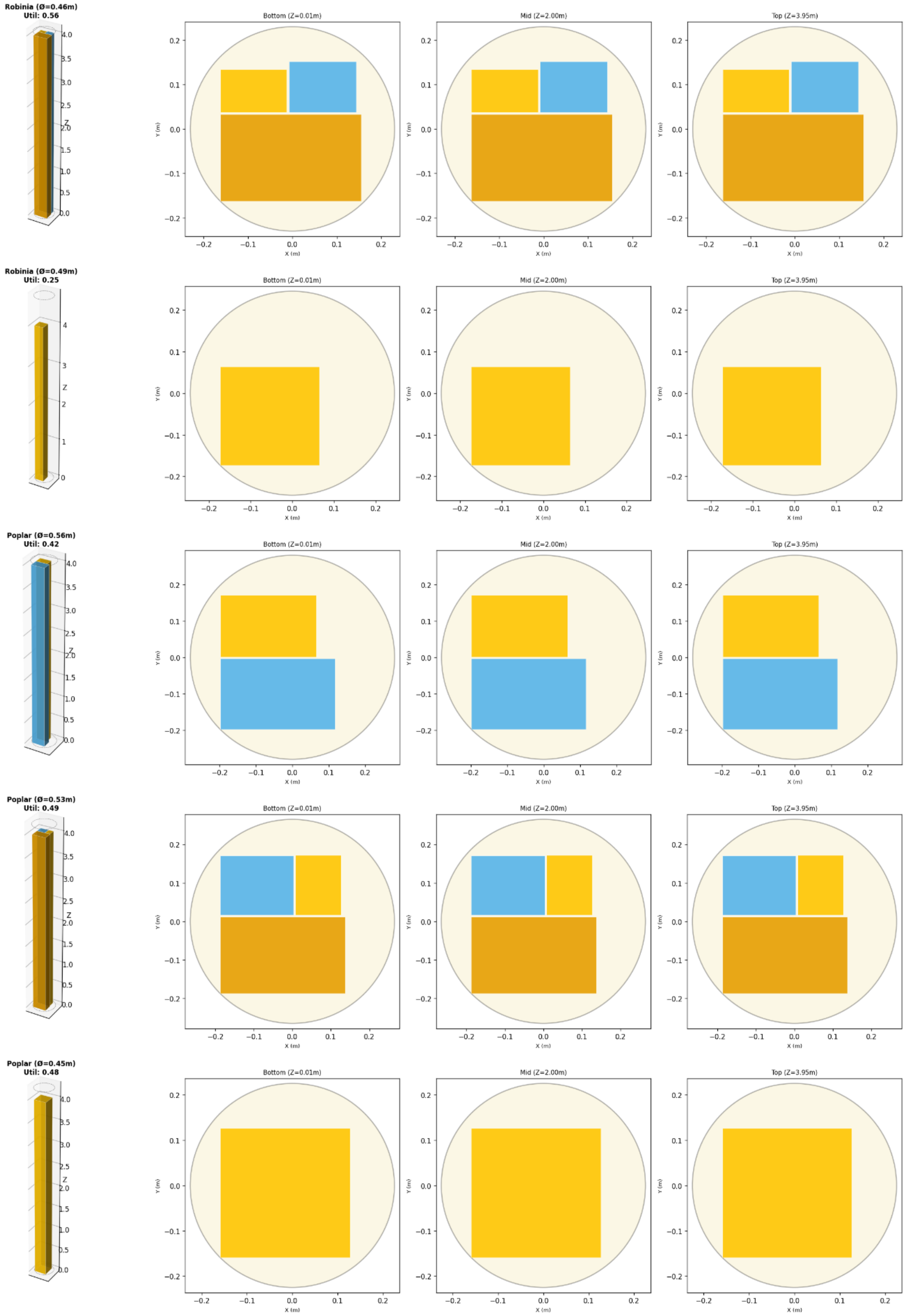


Robinia (Ø=0.54m)
Util: 0.52
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)



Poplar (Ø=0.47m)
Util: 0.46
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)


Robinia (Ø=0.47m)
Util: 0.52
Robinia (Ø=0.48m)
Util: 0.33
Robinia (Ø=0.53m)
Util: 0.43
Poplar (Ø=0.46m)
Util: 0.39
Robinia (Ø=0.49m)
Util: 0.46
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
X (m)
Y (m)
Z

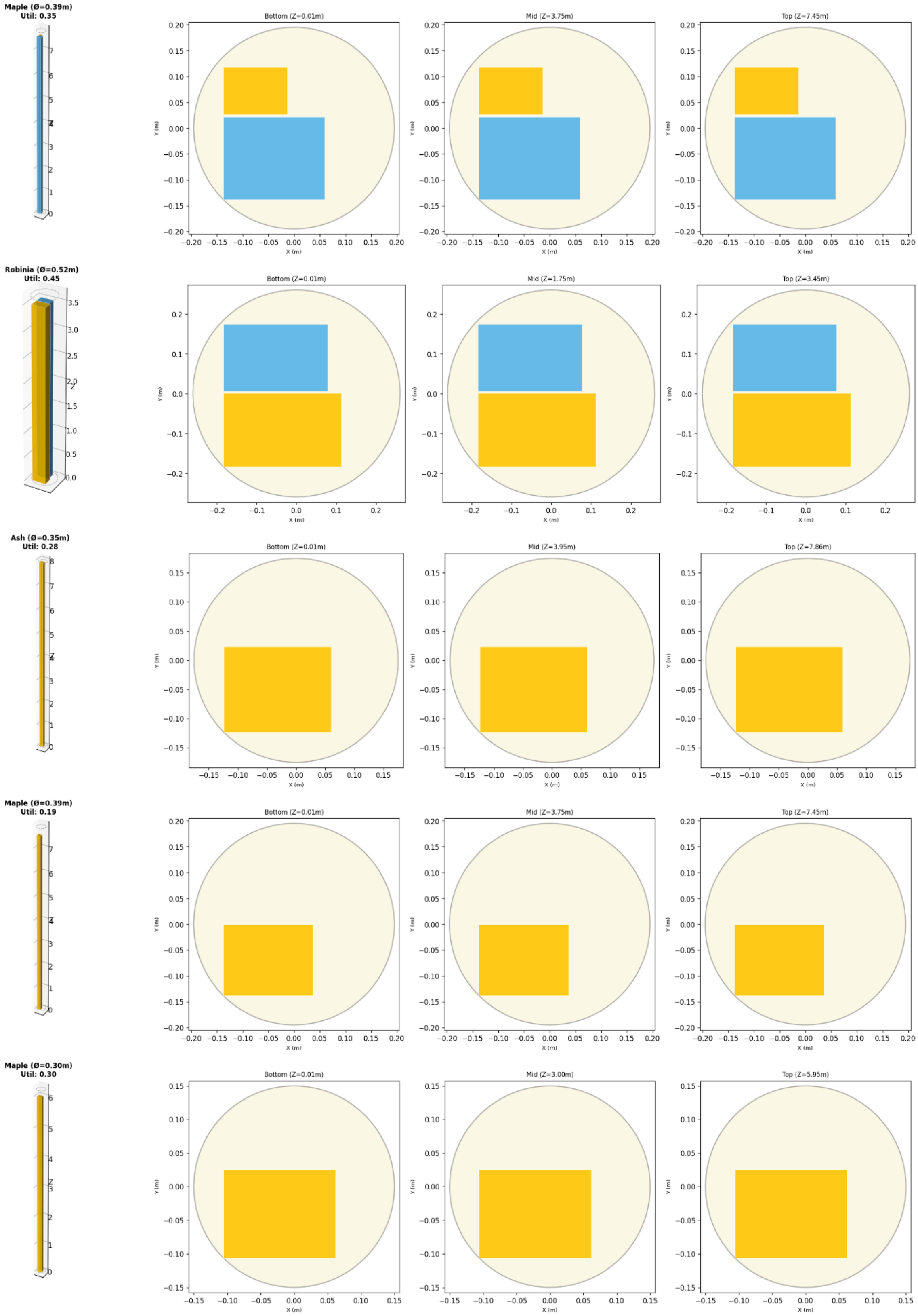


Robinia (Ø=0.46m)
Util: 0.56
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Robinia (Ø=0.49m)
Util: 0.25
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.56m)
Util: 0.42
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.53m)
Util: 0.49
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.45m)
Util: 0.48
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
X (m)
Y (m)

Cross-sections Greedy Heuristic: Best-Fit Muiden structure

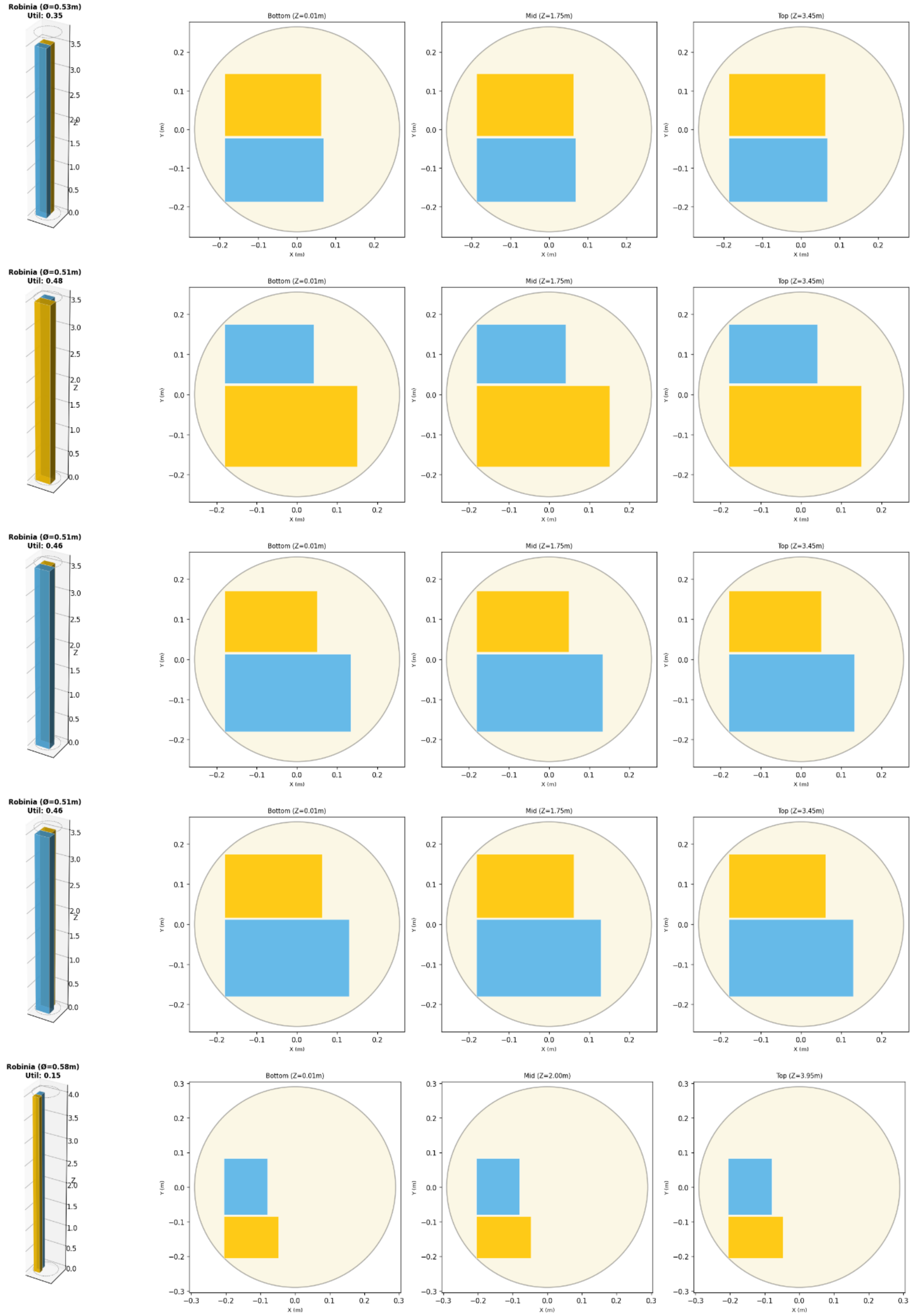

Poplar (Ø=0.51m)
Util: 0.57
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Robinia (Ø=0.40m)
Util: 0.27
Bottom (Z=0.01m)
Mid (Z=1.10m)
Top (Z=2.15m)
Poplar (Ø=0.51m)
Util: 0.54
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Poplar (Ø=0.40m)
Util: 0.23
Bottom (Z=0.01m)
Mid (Z=1.06m)
Top (Z=2.08m)
Poplar (Ø=0.51m)
Util: 0.50
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
X (m)
Y (m)
Z

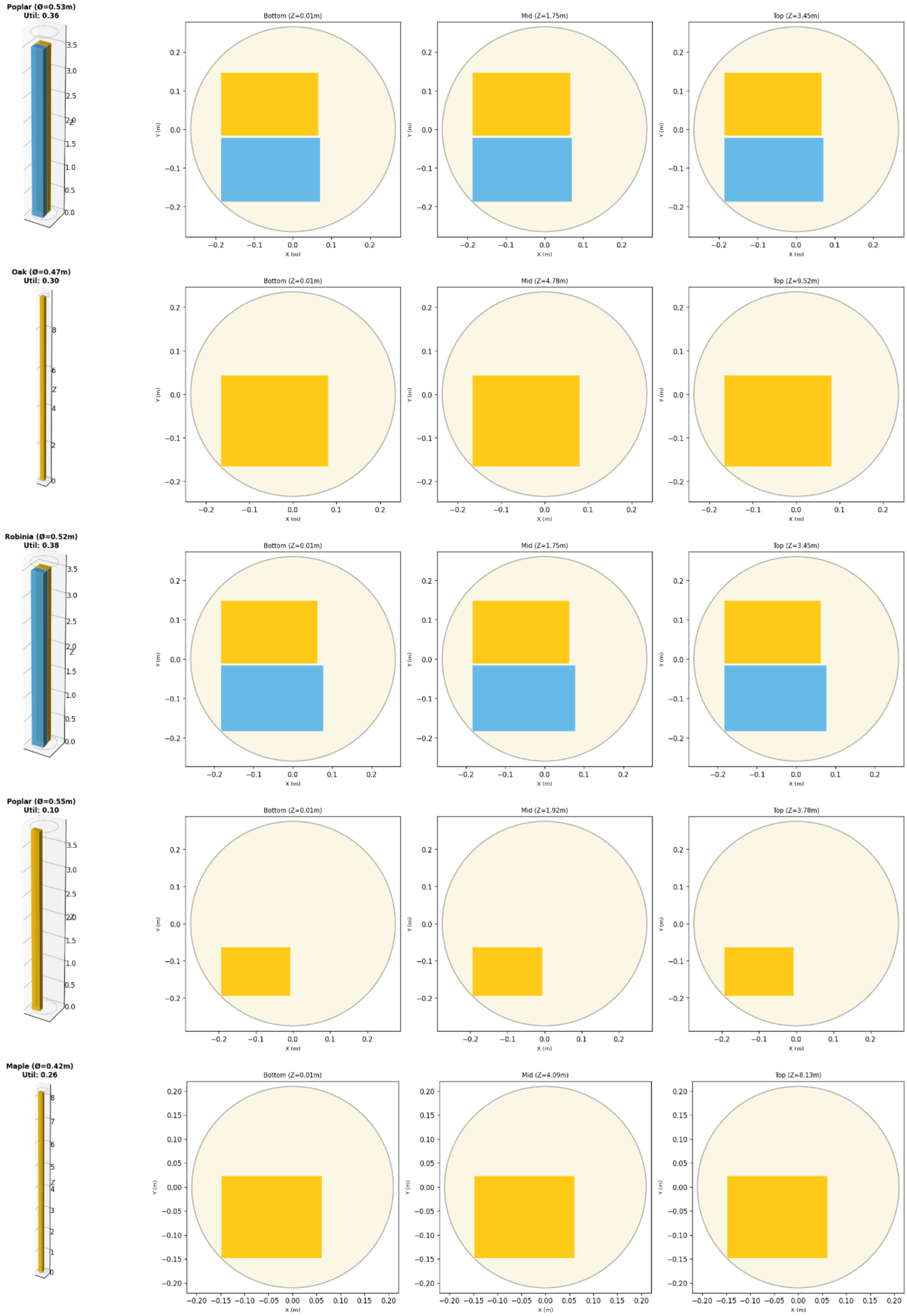

Poplar (Ø=0.53m)
Util: 0.46
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Maple (Ø=0.38m)
Util: 0.32
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
Robinia (Ø=0.53m)
Util: 0.29
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Robinia (Ø=0.30m)
Util: 0.39
Bottom (Z=0.01m)
Mid (Z=0.98m)
Top (Z=1.92m)
Oak (Ø=0.39m)
Util: 0.28
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
X (m)
Y (m)



Maple (Ø=0.39m)
Util: 0.35
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
Robinia (Ø=0.52m)
Util: 0.45
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Ash (Ø=0.35m)
Util: 0.28
Bottom (Z=0.01m)
Mid (Z=3.95m)
Top (Z=7.86m)
Maple (Ø=0.39m)
Util: 0.19
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
Maple (Ø=0.30m)
Util: 0.30
Bottom (Z=0.01m)
Mid (Z=3.00m)
Top (Z=5.95m)
X (m)
Y (m)

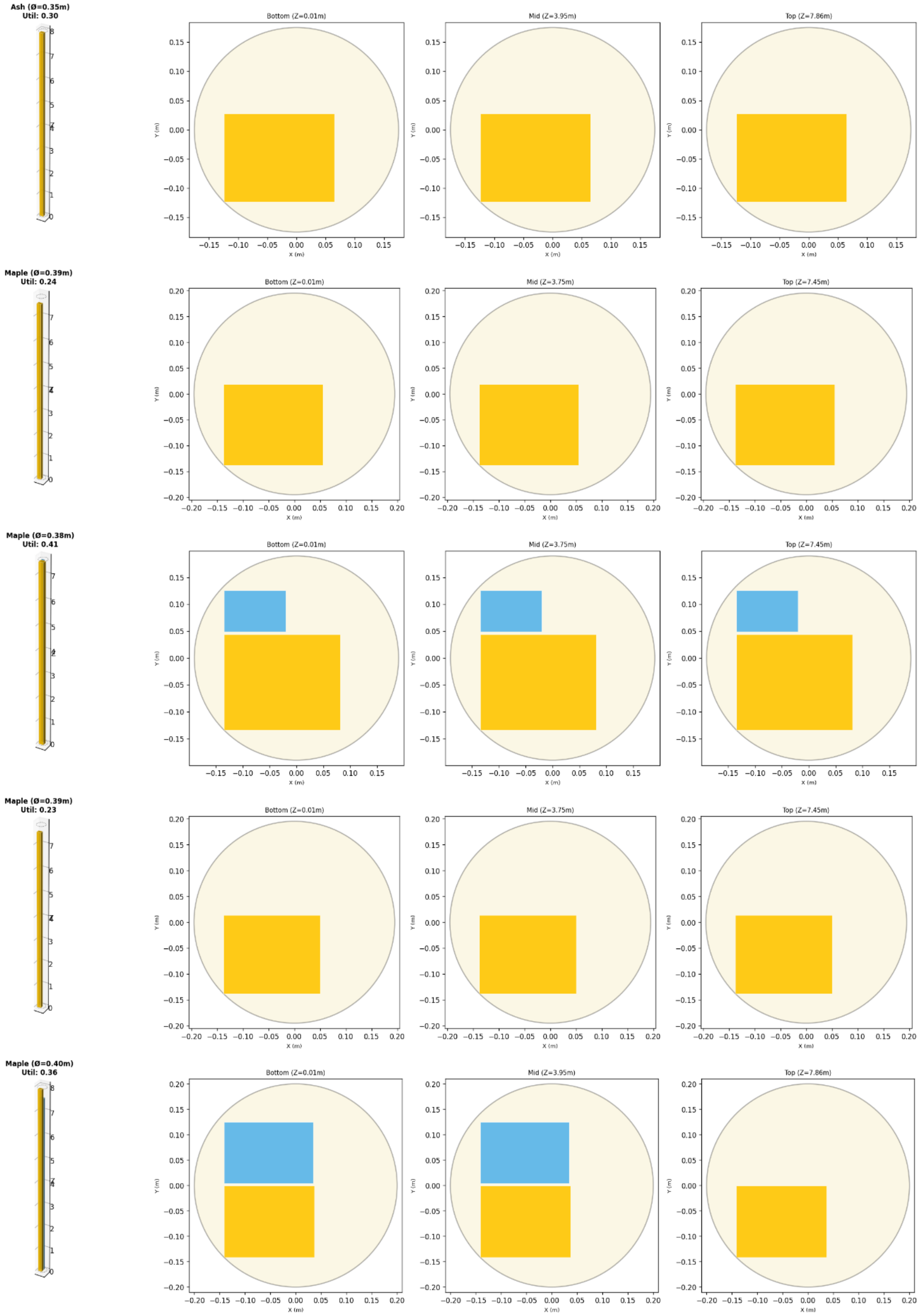

Chestnut (Ø=0.42m)
Util: 0.13
Bottom (Z=0.01m)
Mid (Z=2.39m)
Top (Z=4.73m)
Chestnut (Ø=0.51m)
Util: 0.38
Bottom (Z=0.01m)
Mid (Z=3.27m)
Top (Z=6.50m)
Poplar (Ø=0.53m)
Util: 0.39
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Poplar (Ø=0.30m)
Util: 0.34
Bottom (Z=0.01m)
Mid (Z=0.90m)
Top (Z=1.74m)
Poplar (Ø=0.52m)
Util: 0.46
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
X (m)
Y (m)
Z


Robinia (Ø=0.53m)
Util: 0.35
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Robinia (Ø=0.51m)
Util: 0.48
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Robinia (Ø=0.51m)
Util: 0.46
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Robinia (Ø=0.51m)
Util: 0.46
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Robinia (Ø=0.58m)
Util: 0.15
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
X (m)
Y (m)
Z


Poplar (Ø=0.53m)
Util: 0.36
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Oak (Ø=0.47m)
Util: 0.30
Bottom (Z=0.01m)
Mid (Z=4.78m)
Top (Z=9.52m)
Robinia (Ø=0.52m)
Util: 0.38
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Poplar (Ø=0.55m)
Util: 0.10
Bottom (Z=0.01m)
Mid (Z=1.92m)
Top (Z=3.78m)
Maple (Ø=0.42m)
Util: 0.26
Bottom (Z=0.01m)
Mid (Z=4.09m)
Top (Z=8.13m)
X (m)
Y (m)

Elm (Ø=0.23m)
Util: 0.42
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
Ash (Ø=0.39m)
Util: 0.23
Bottom (Z=0.01m)
Mid (Z=4.38m)
Top (Z=8.70m)
Robinia (Ø=0.54m)
Util: 0.39
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Robinia (Ø=0.39m)
Util: 0.40
Bottom (Z=0.01m)
Mid (Z=1.25m)
Top (Z=2.45m)
Robinia (Ø=0.53m)
Util: 0.43
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
X (m)
Y (m)
Z

Ash (Ø=0.39m)
Util: 0.22
Bottom (Z=0.01m)
Mid (Z=4.38m)
Top (Z=8.70m)
Ash (Ø=0.44m)
Util: 0.33
Bottom (Z=0.01m)
Mid (Z=4.88m)
Top (Z=9.71m)
Robinia (Ø=0.36m)
Util: 0.37
Bottom (Z=0.01m)
Mid (Z=1.10m)
Top (Z=2.15m)
Maple (Ø=0.39m)
Util: 0.25
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
Oak (Ø=0.39m)
Util: 0.38
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
X (m)
Y (m)
Z



Ash (Ø=0.35m)
Util: 0.30
Bottom (Z=0.01m)
Mid (Z=3.95m)
Top (Z=7.86m)
Maple (Ø=0.39m)
Util: 0.24
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
Maple (Ø=0.38m)
Util: 0.41
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
Maple (Ø=0.39m)
Util: 0.23
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
Maple (Ø=0.40m)
Util: 0.36
Bottom (Z=0.01m)
Mid (Z=3.95m)
Top (Z=7.86m)
X (m)
Y (m)

Graphs of GA progress for default parameters

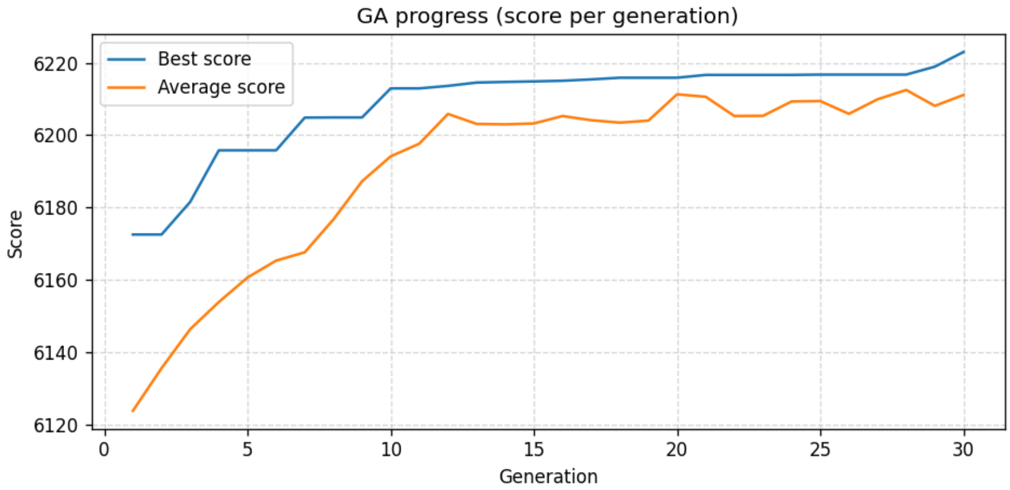


Figure 1: Genetic Algorithm progression graph of the bin-packing optimization for the *2x2* structure.

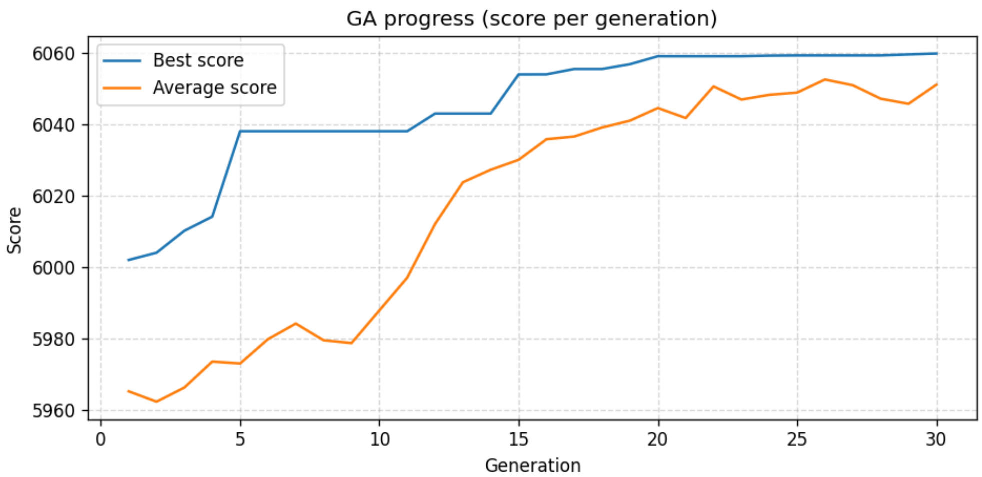


Figure 2: Genetic Algorithm progression graph of the bin-packing optimization for the 2x8x8x2 structure.

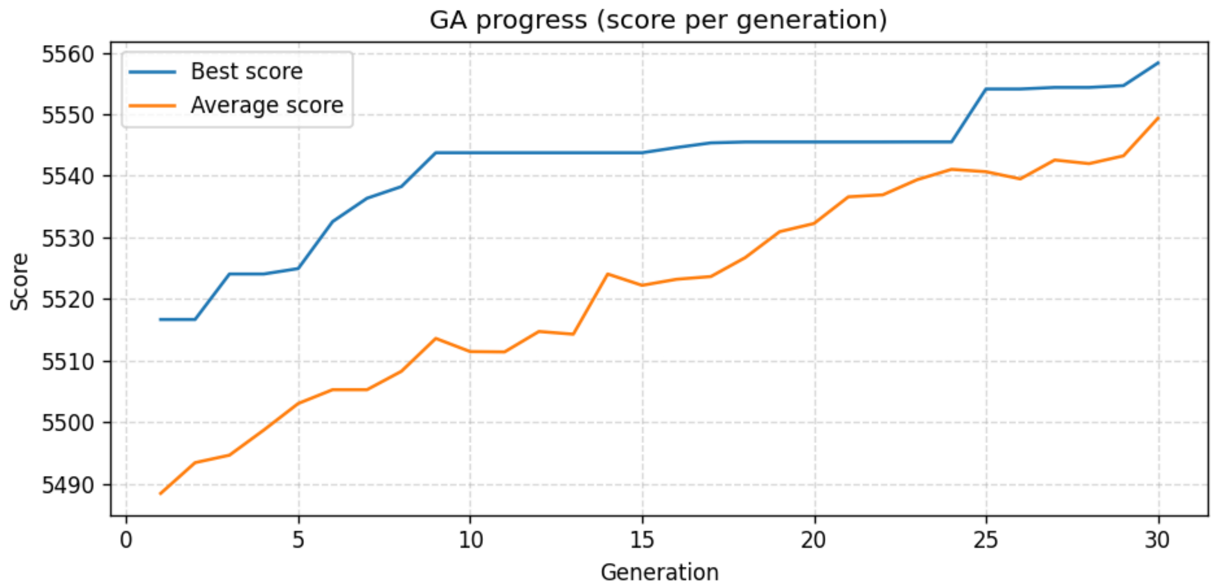


Figure 3: Genetic Algorithm progression graph of the bin-packing optimization for the 8x8x2 structure.

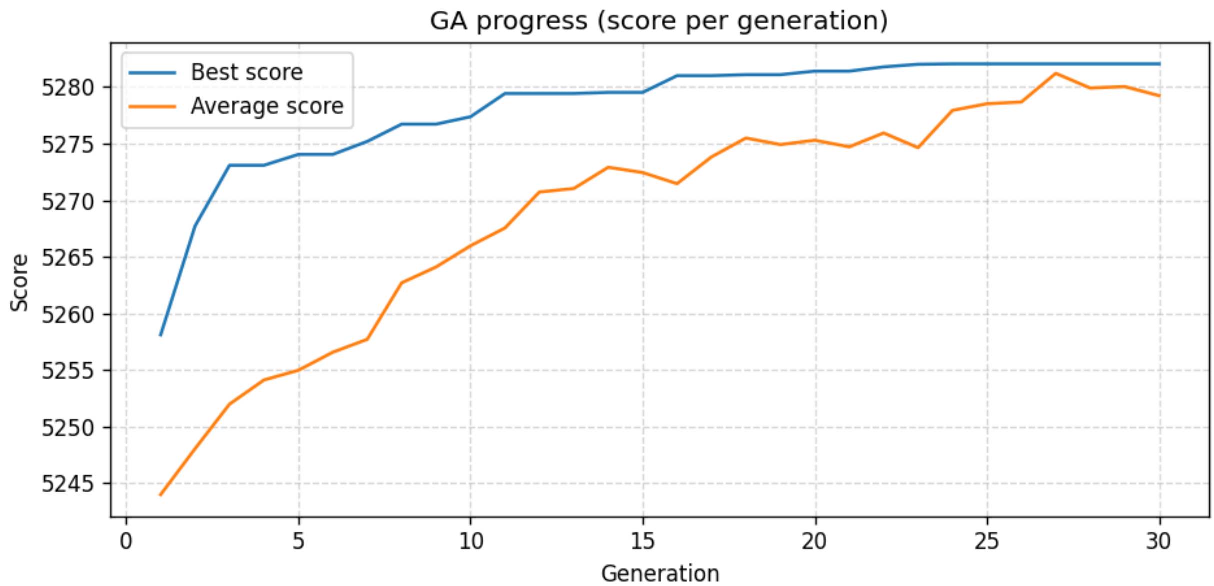


Figure 4: Genetic Algorithm progression graph of the bin-packing optimization for the 8x8 structure.

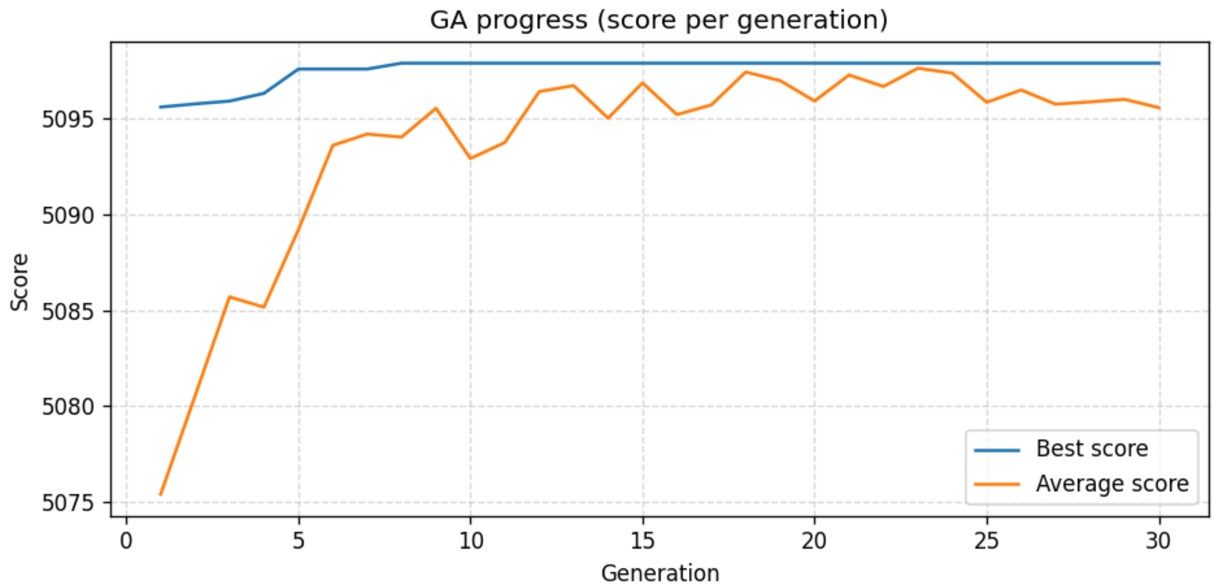
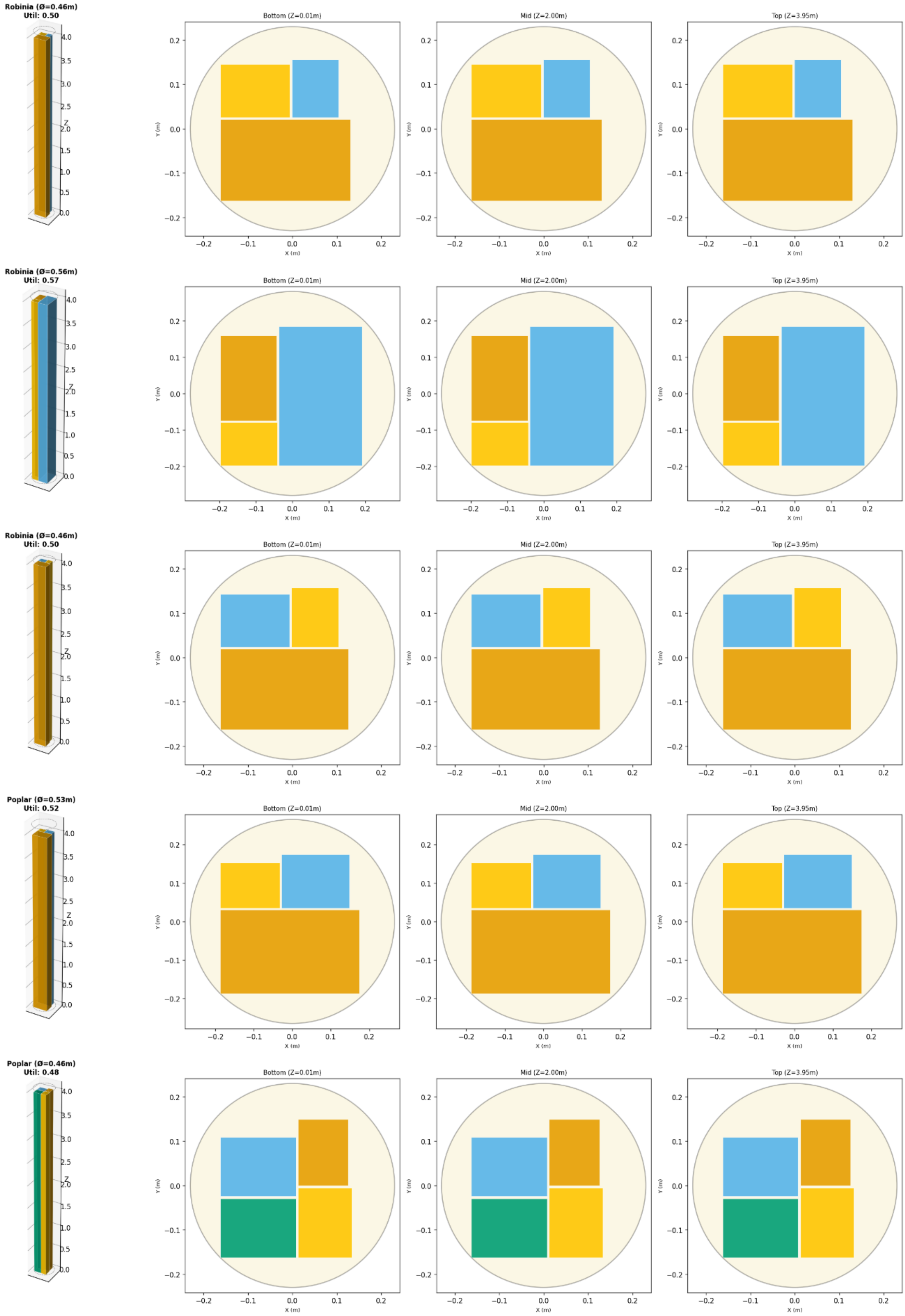
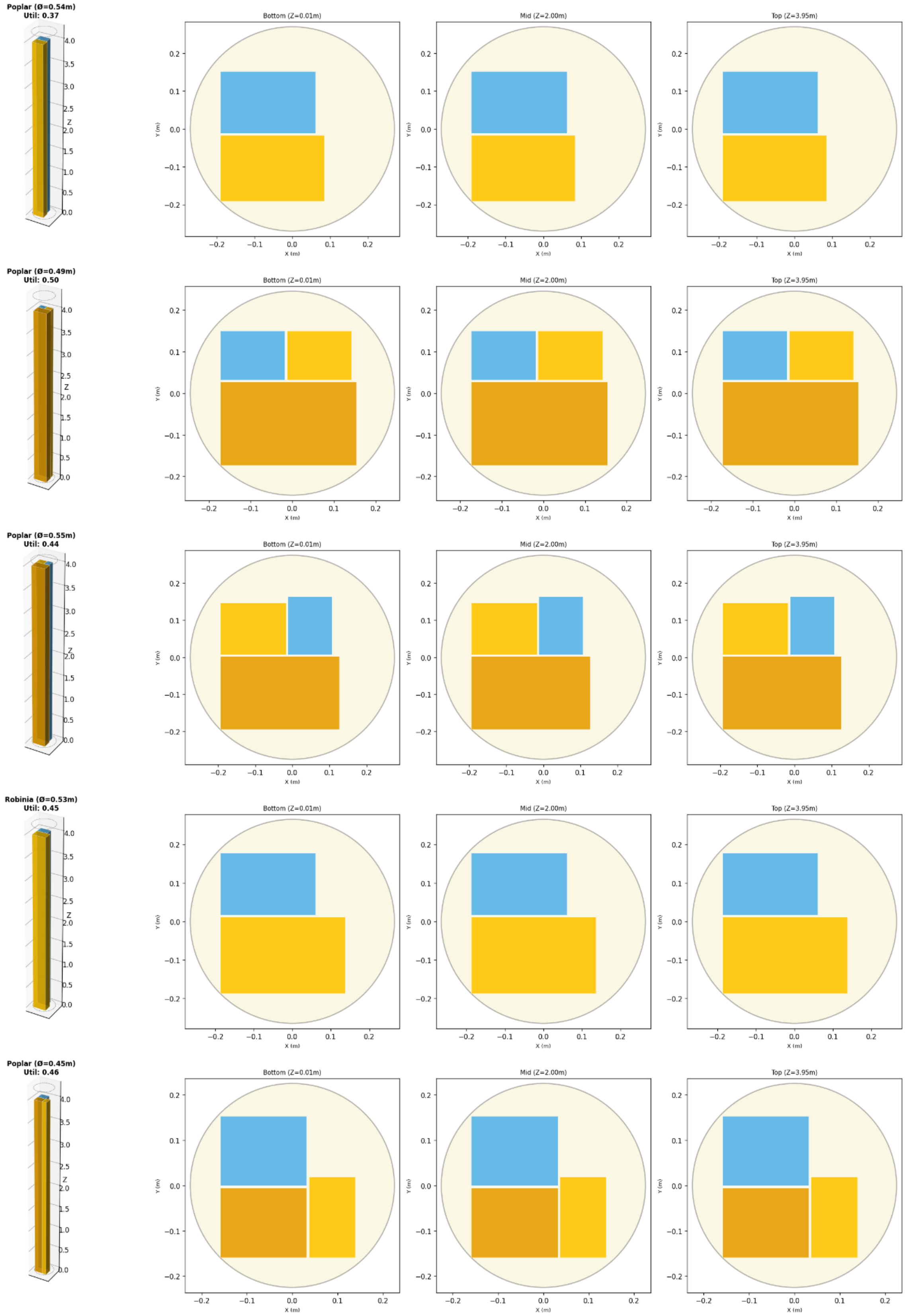
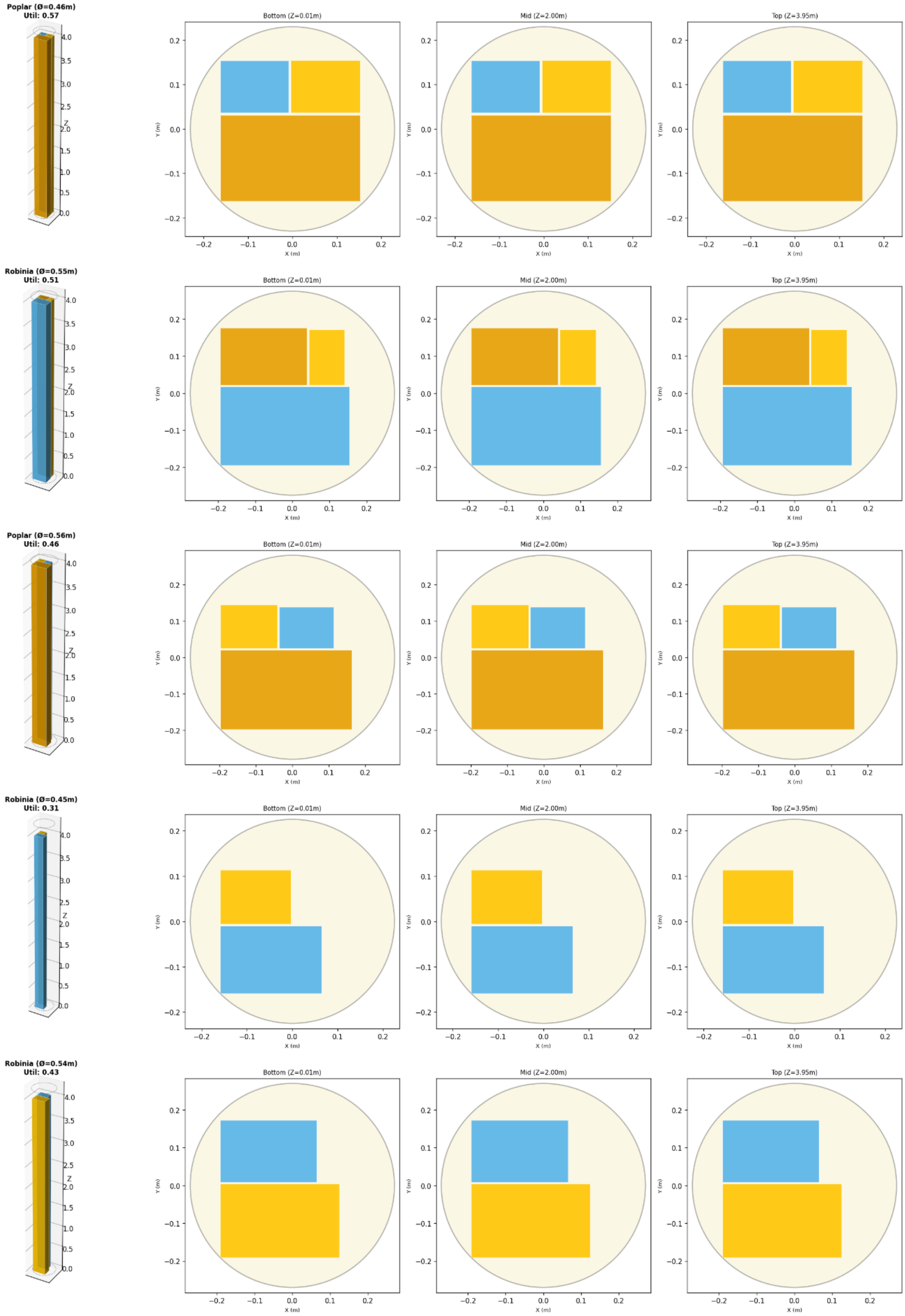


Figure 5: Genetic Algorithm progression graph of the bin-packing optimization for the 4x4 structure.

Cross-sections Metaheuristic 8x8x2

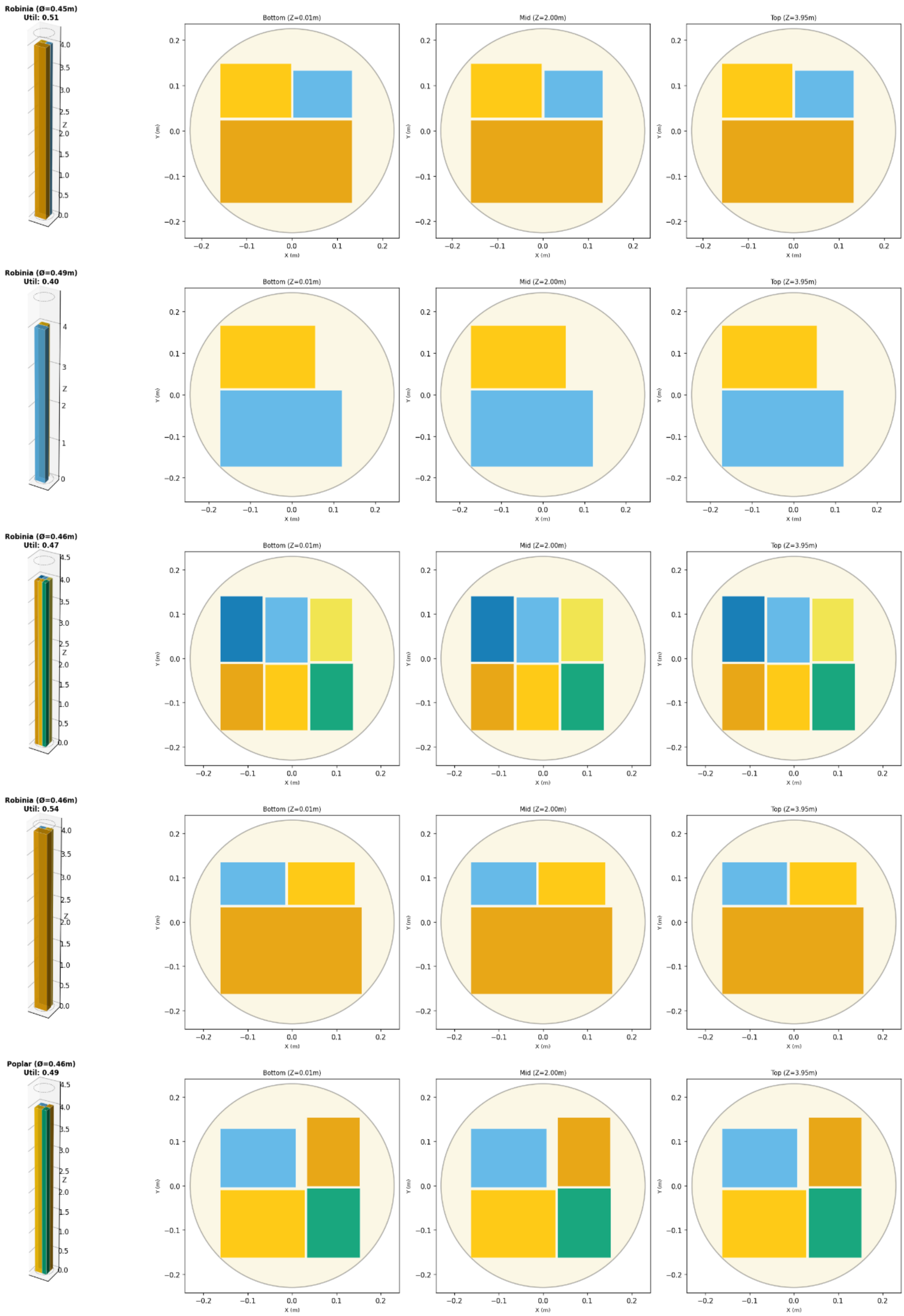


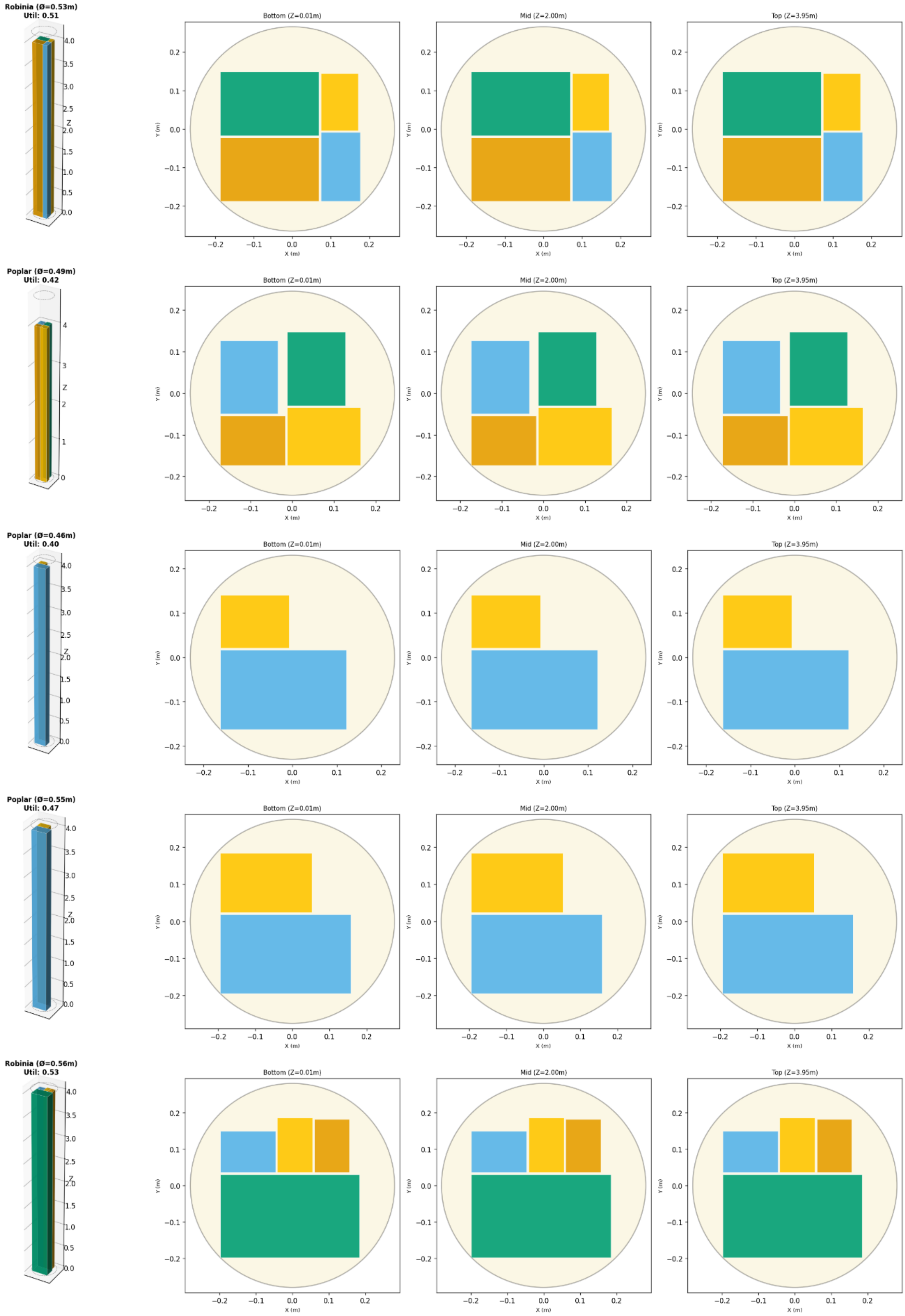

Poplar (Ø=0.54m)
Util: 0.37
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.49m)
Util: 0.50
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.55m)
Util: 0.44
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Robinia (Ø=0.53m)
Util: 0.45
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.45m)
Util: 0.46
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
X (m)
Y (m)
Z

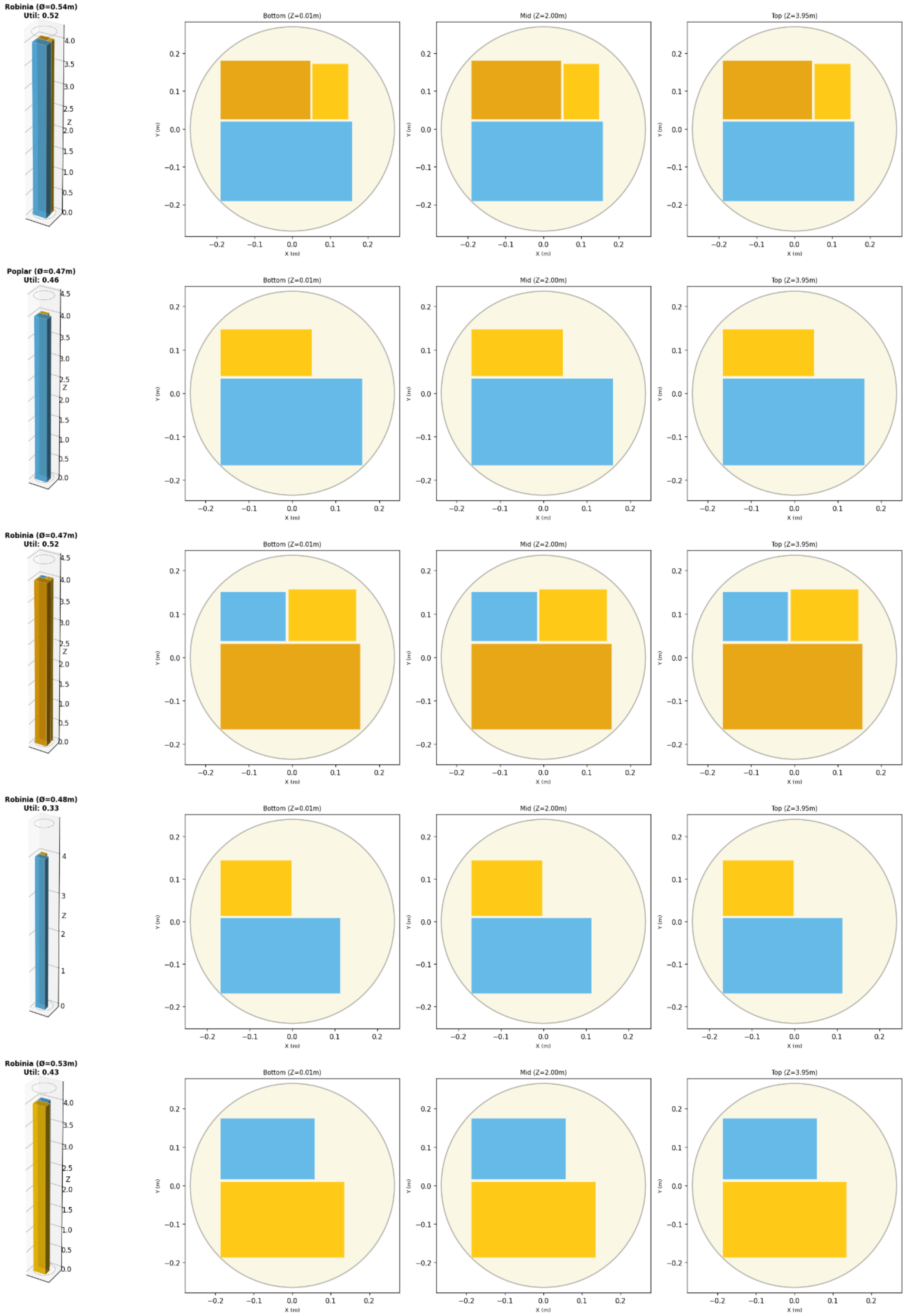


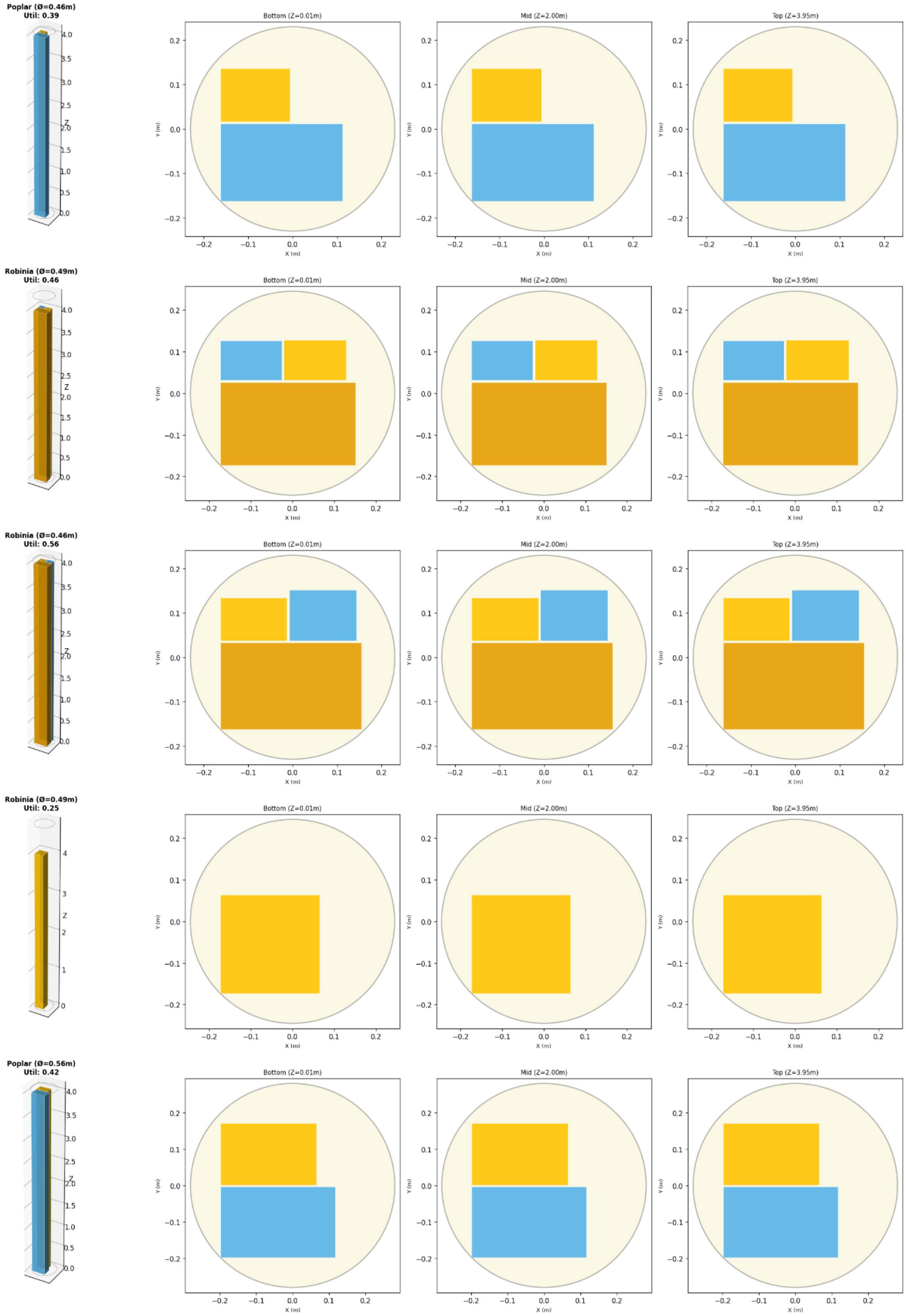
Poplar (Ø=0.46m)
Util: 0.57
Robinia (Ø=0.55m)
Util: 0.51
Poplar (Ø=0.56m)
Util: 0.46
Robinia (Ø=0.45m)
Util: 0.31
Robinia (Ø=0.54m)
Util: 0.43
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
X (m)
Y (m)
Z

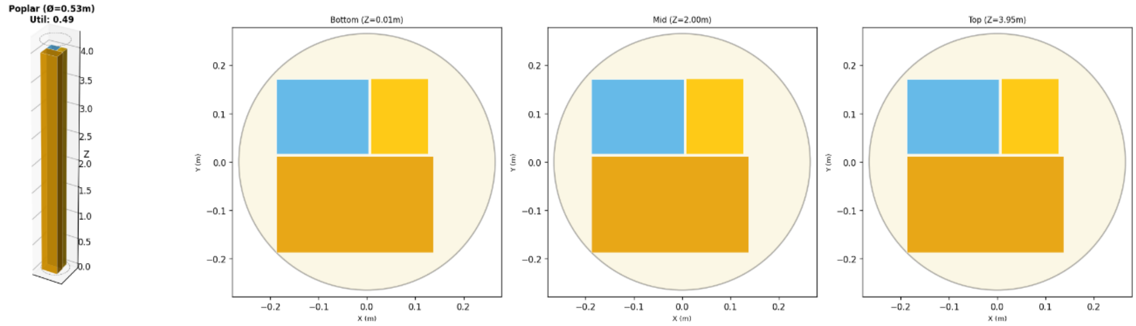
Cross-sections MetaHeuristic 2x8x8x2

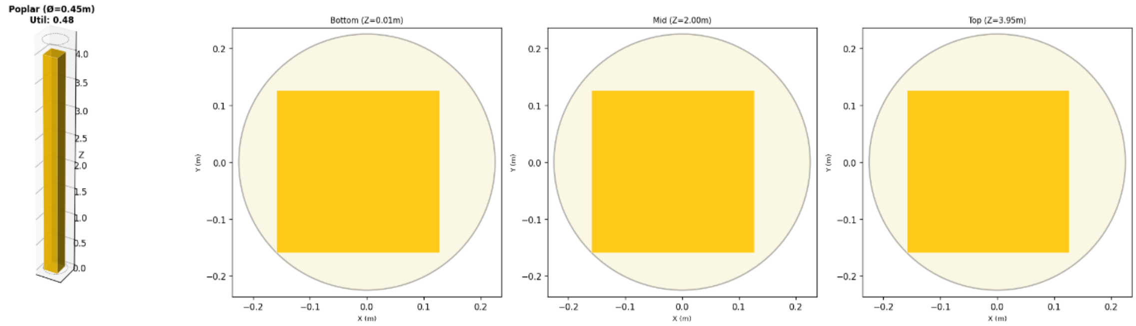


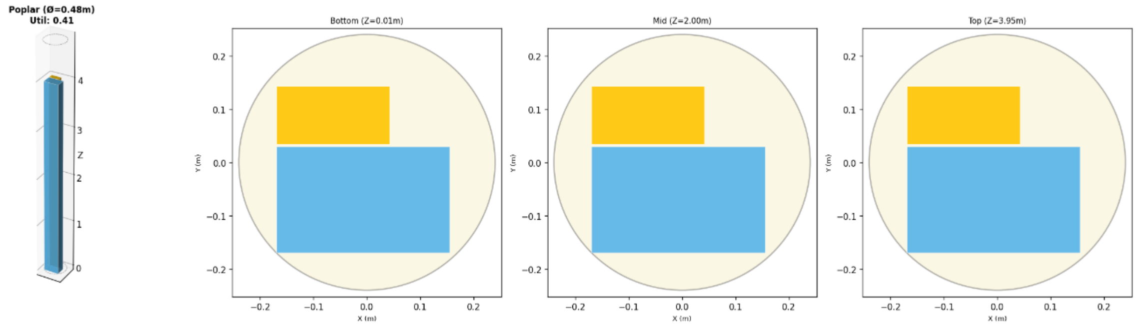

Robinia (Ø=0.53m)
Util: 0.51
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.49m)
Util: 0.42
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.46m)
Util: 0.40
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.55m)
Util: 0.47
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Robinia (Ø=0.56m)
Util: 0.53
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)

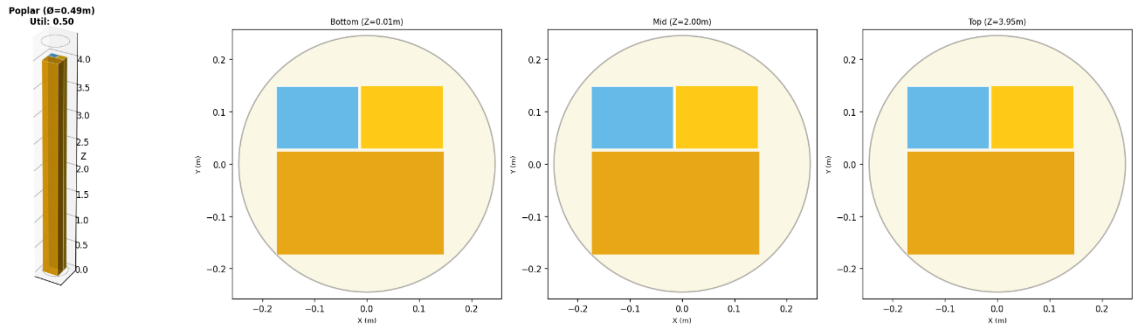

Robinia (Ø=0.54m)
Util: 0.52
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.47m)
Util: 0.46
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Robinia (Ø=0.47m)
Util: 0.52
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Robinia (Ø=0.48m)
Util: 0.33
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Robinia (Ø=0.53m)
Util: 0.43
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
X (m)
Y (m)
Z

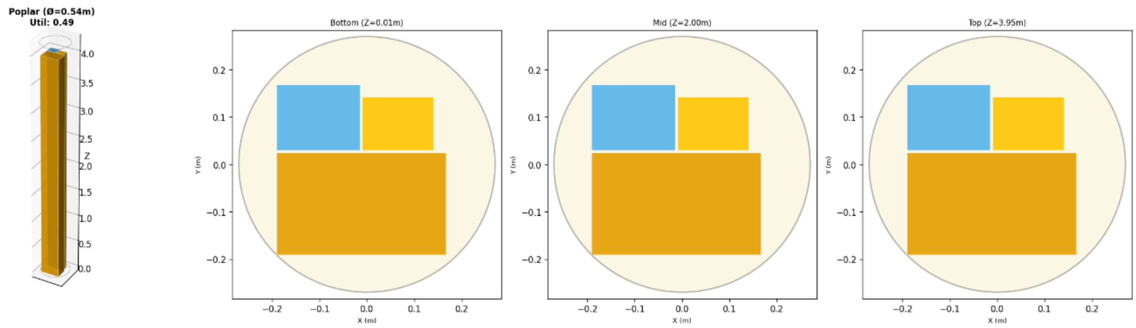

Poplar (Ø=0.46m)
Util: 0.39
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Robinia (Ø=0.49m)
Util: 0.46
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Robinia (Ø=0.46m)
Util: 0.56
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Robinia (Ø=0.49m)
Util: 0.25
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.56m)
Util: 0.42
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
X (m)
Y (m)
Z

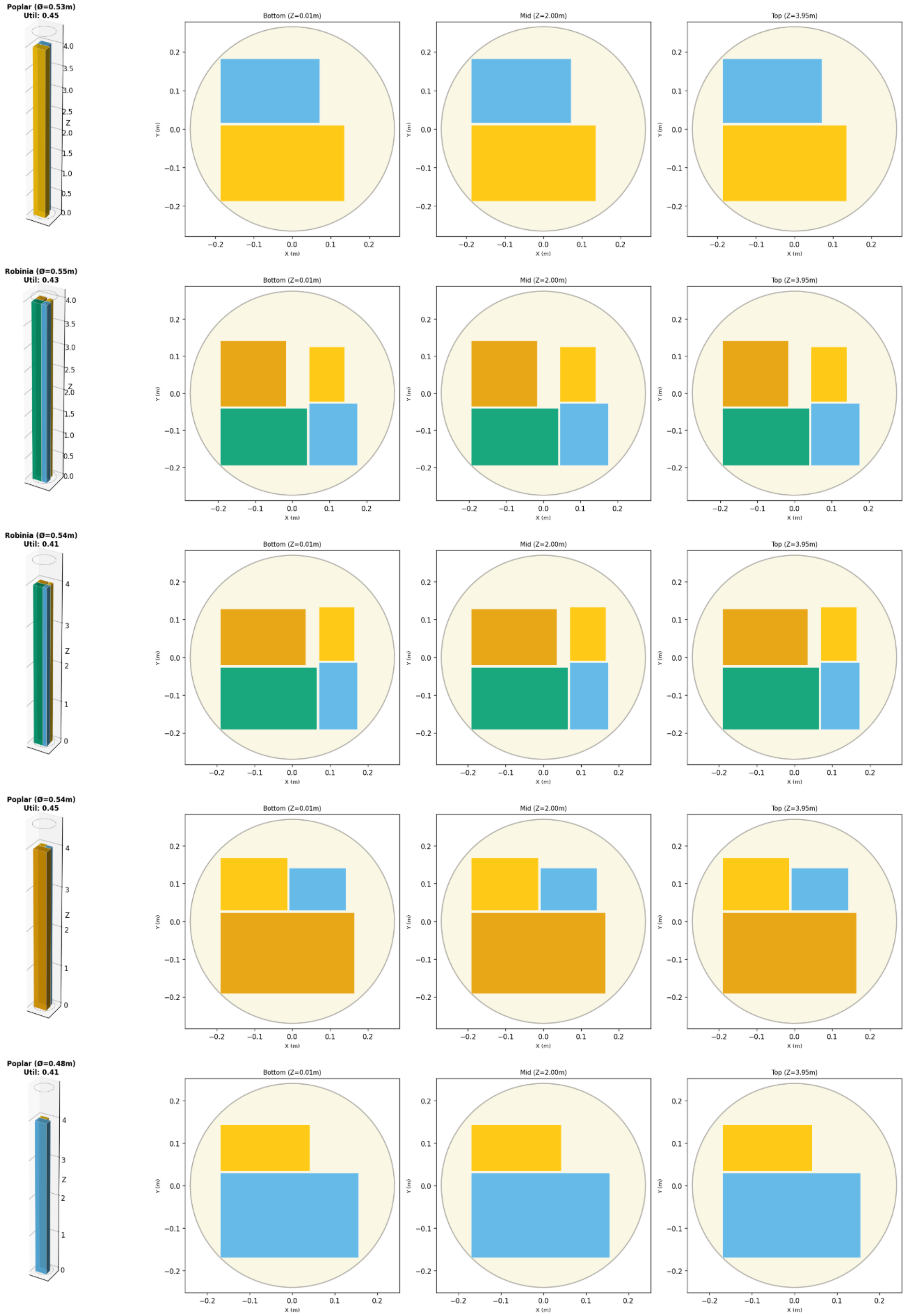

Poplar (Ø=0.53m)
Util: 0.49
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
X (m)
Y (m)


Poplar (Ø=0.45m)
Util: 0.48
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
X (m)
Y (m)

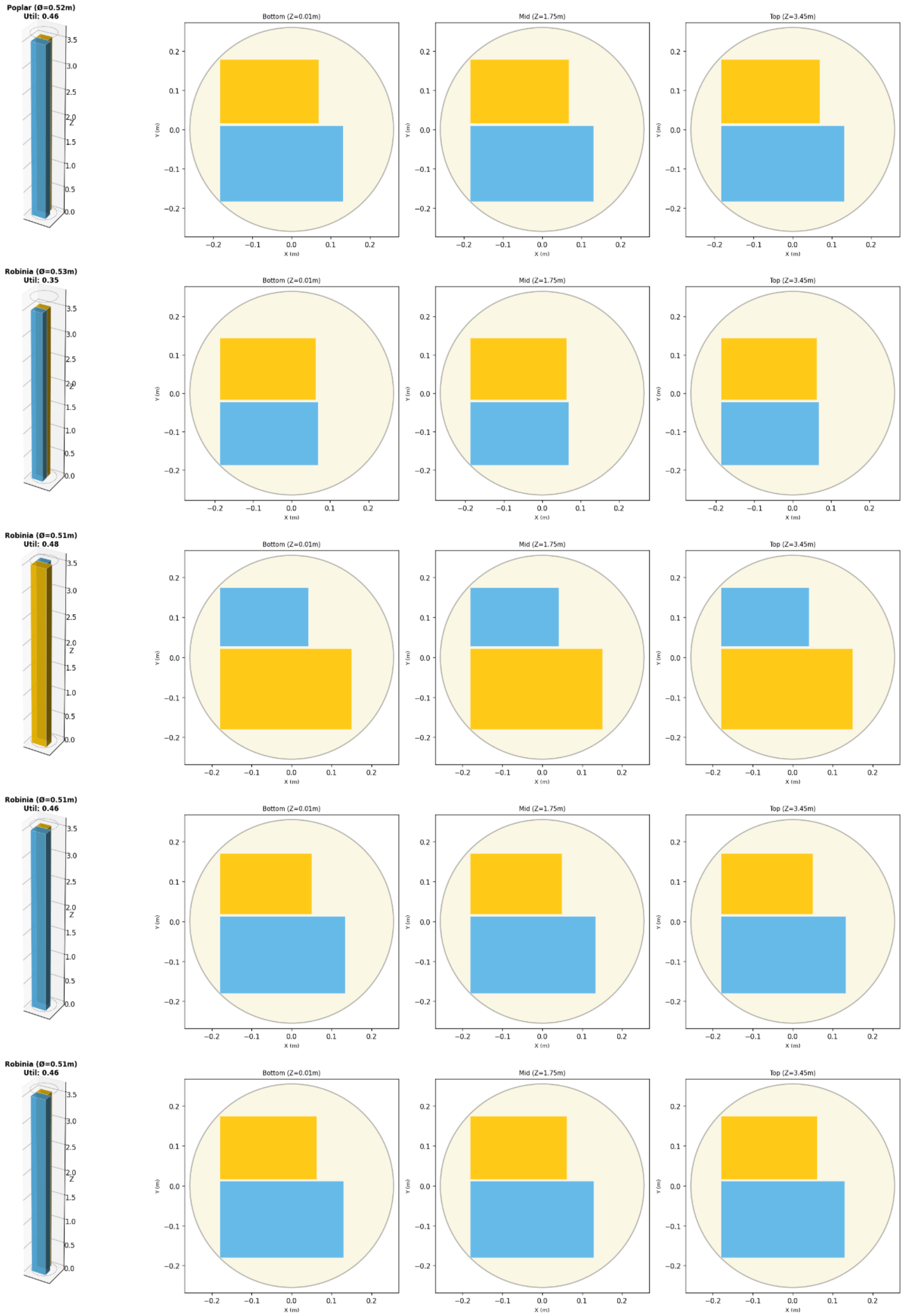

Poplar (Ø=0.48m)
Util: 0.41
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
X (m)
Y (m)

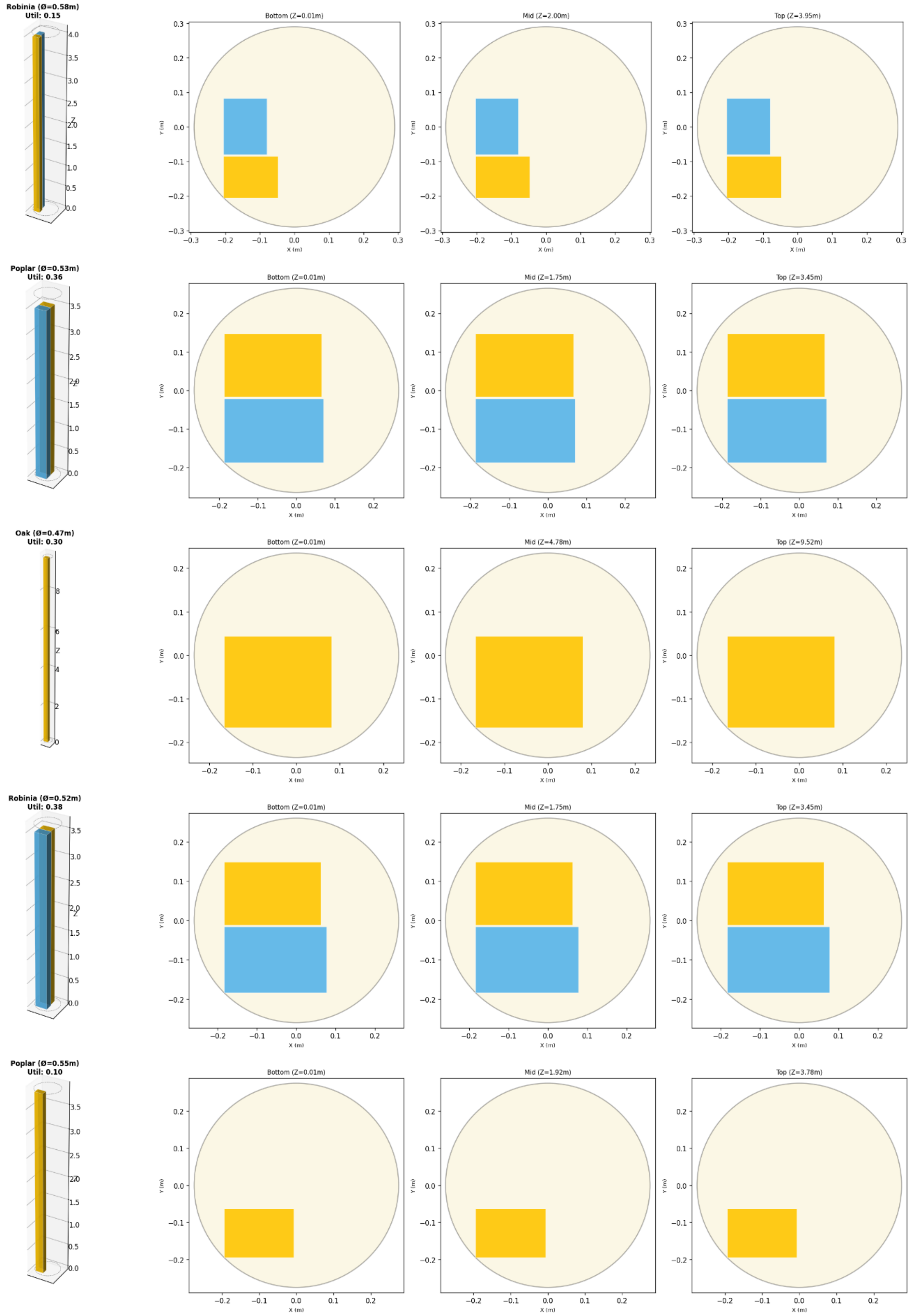

Poplar (Ø=0.49m)
Util: 0.50
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
X (m)
Y (m)


Poplar (Ø=0.54m)
Util: 0.49
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
X (m)
Y (m)

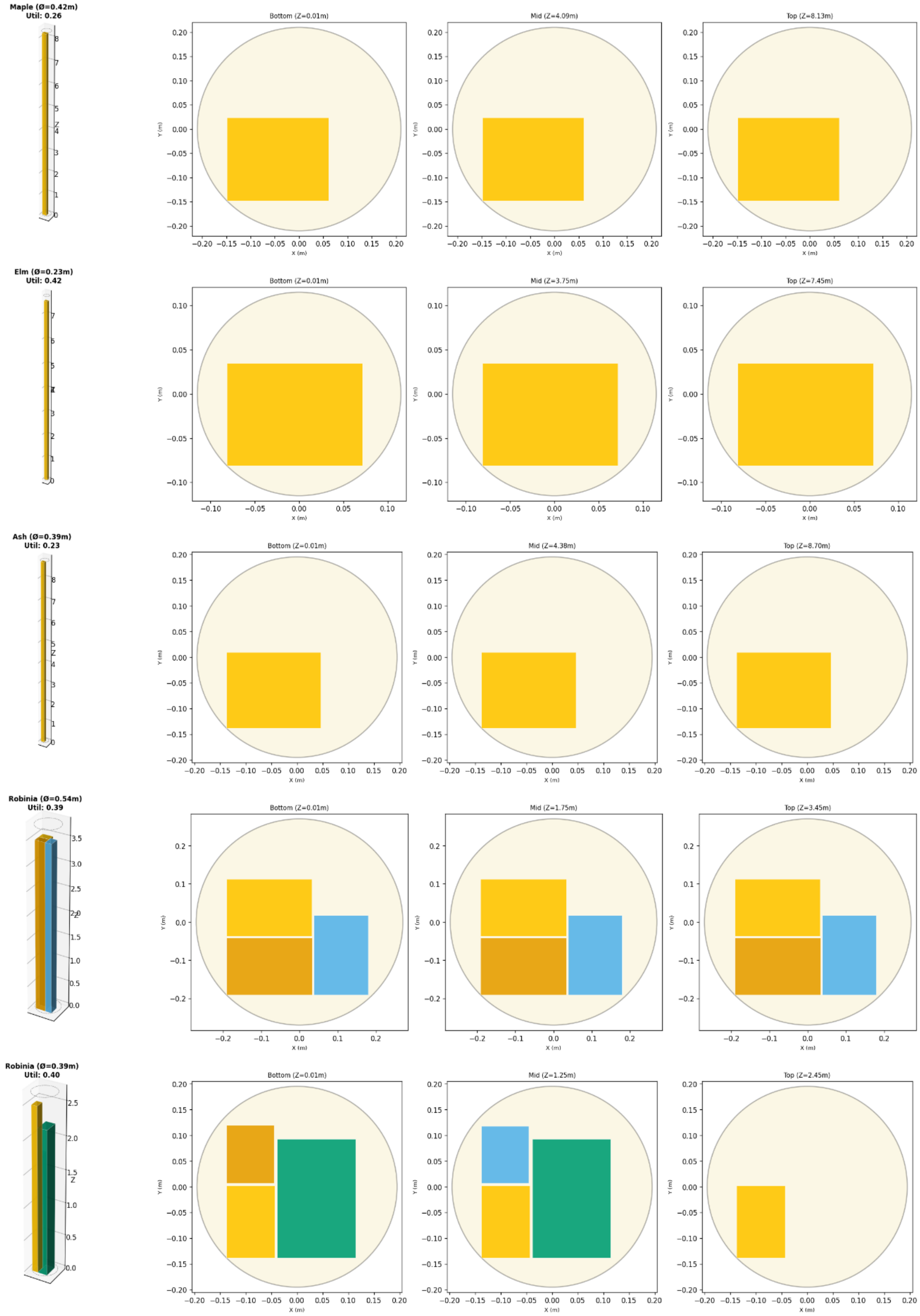

Poplar (Ø=0.53m)
Util: 0.45
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Robinia (Ø=0.55m)
Util: 0.43
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Robinia (Ø=0.54m)
Util: 0.41
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.54m)
Util: 0.45
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.48m)
Util: 0.41
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
X (m)
Y (m)
Z

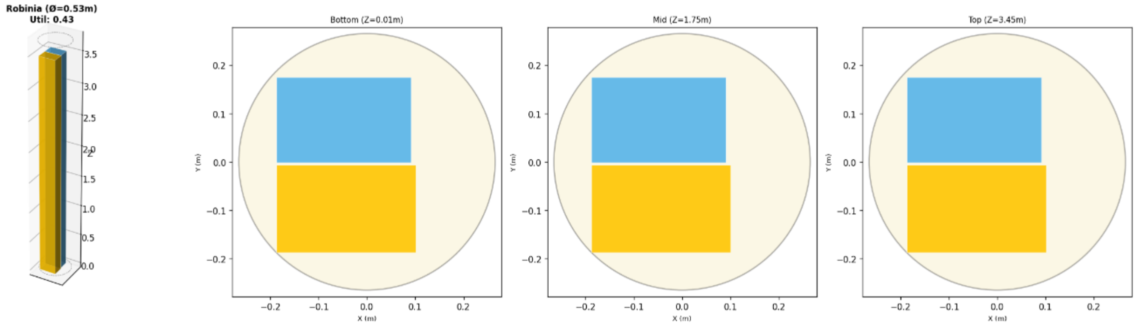
Cross-sections MetaHeuristic Muiden structure

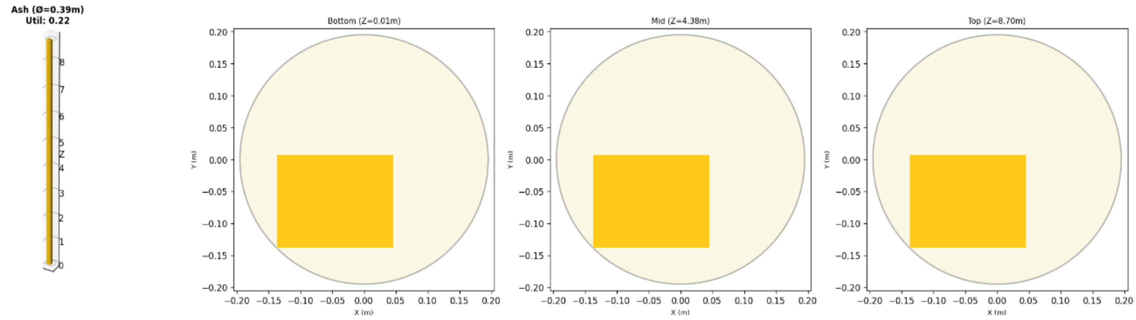


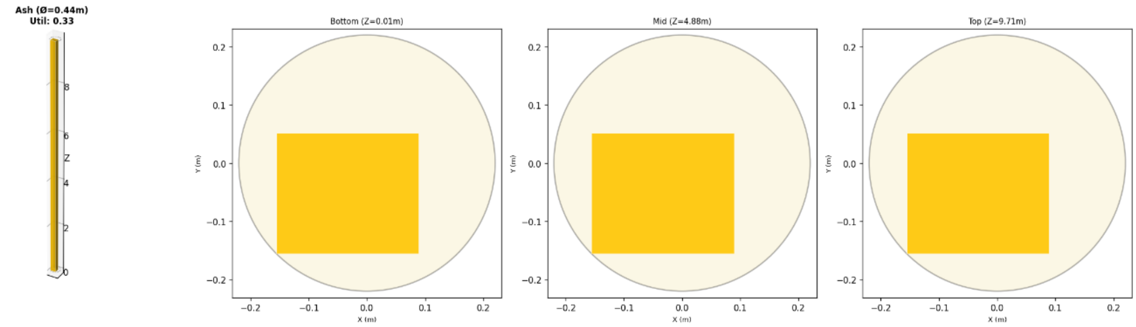


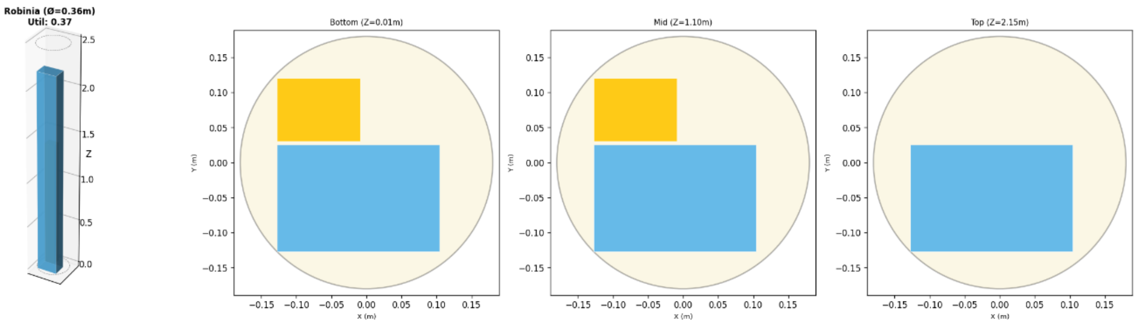
Robinia (Ø=0.58m)
Util: 0.15
Bottom (Z=0.01m)
Mid (Z=2.00m)
Top (Z=3.95m)
Poplar (Ø=0.53m)
Util: 0.36
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Oak (Ø=0.47m)
Util: 0.30
Bottom (Z=0.01m)
Mid (Z=4.78m)
Top (Z=9.52m)
Robinia (Ø=0.52m)
Util: 0.38
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Poplar (Ø=0.55m)
Util: 0.10
Bottom (Z=0.01m)
Mid (Z=1.92m)
Top (Z=3.78m)
X (m)
Y (m)
Z

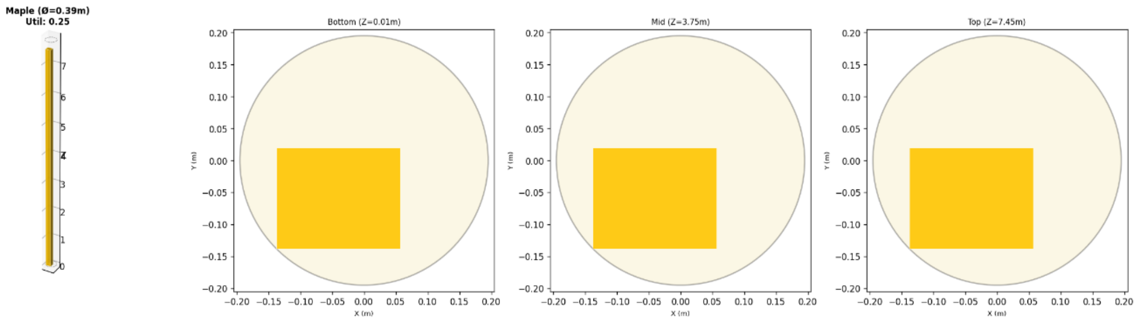

Maple (Ø=0.42m)
Util: 0.26
Bottom (Z=0.01m)
Mid (Z=4.09m)
Top (Z=8.13m)
Elm (Ø=0.23m)
Util: 0.42
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
Ash (Ø=0.39m)
Util: 0.23
Bottom (Z=0.01m)
Mid (Z=4.38m)
Top (Z=8.70m)
Robinia (Ø=0.54m)
Util: 0.39
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Robinia (Ø=0.39m)
Util: 0.40
Bottom (Z=0.01m)
Mid (Z=1.25m)
Top (Z=2.45m)
X (m)
Y (m)
Z

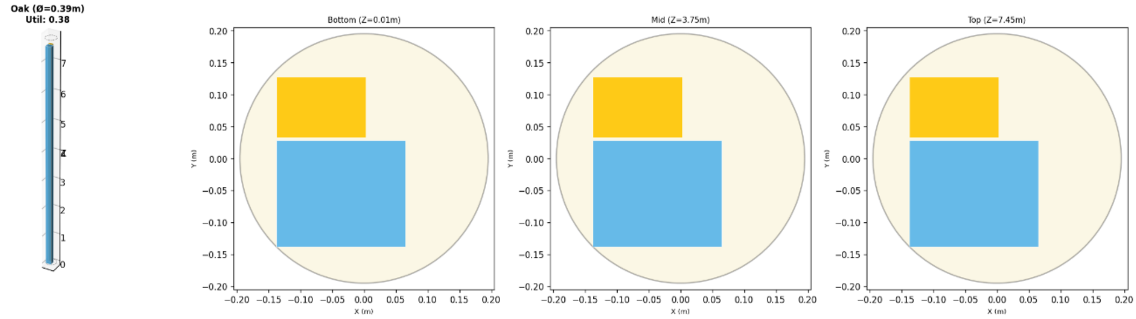

Robinia (Ø=0.53m)
Util: 0.43
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
X (m)
Y (m)

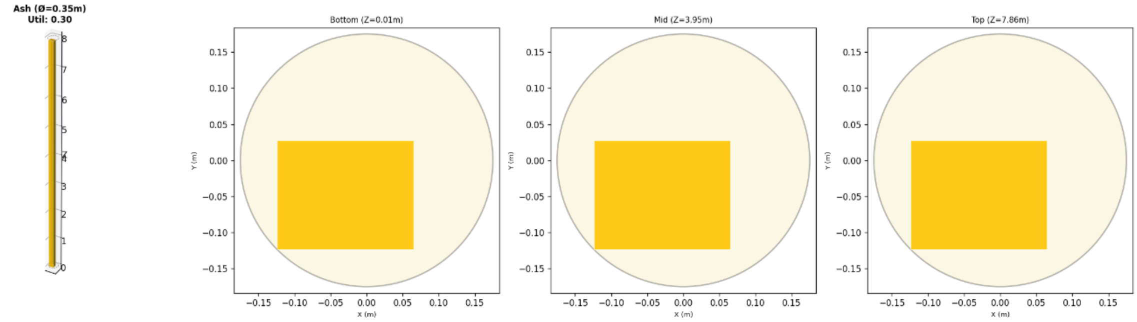

Ash (Ø=0.39m)
Util: 0.22
Bottom (Z=0.01m)
Mid (Z=4.38m)
Top (Z=8.70m)
X (m)
Y (m)

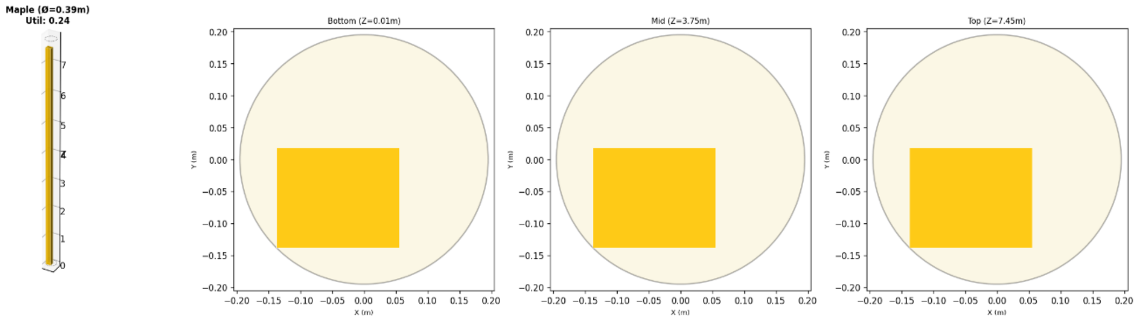

Ash (Ø=0.44m)
Util: 0.33
Bottom (Z=0.01m)
Mid (Z=4.88m)
Top (Z=9.71m)
X (m)
Y (m)

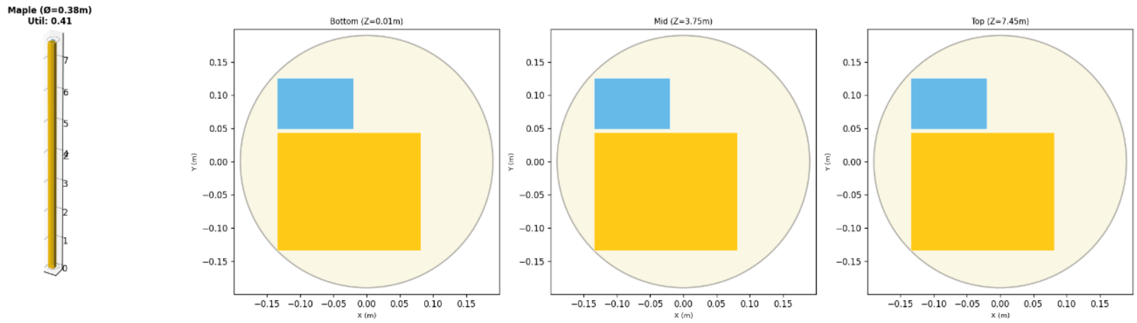

Robinia (Ø=0.36m)
Util: 0.37
Bottom (Z=0.01m)
Mid (Z=1.10m)
Top (Z=2.15m)
X (m)
Y (m)

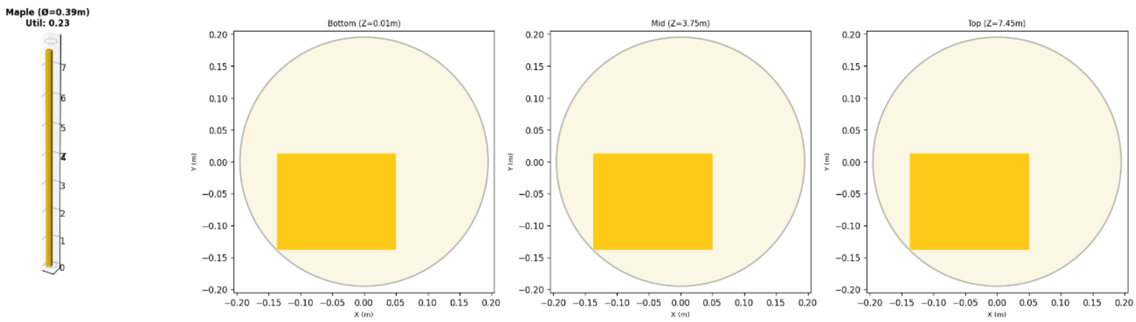

Maple (Ø=0.39m)
Util: 0.25
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
X (m)
Y (m)


Oak (Ø=0.39m)
Util: 0.38
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
X (m)
Y (m)


Ash (Ø=0.35m)
Util: 0.30
Bottom (Z=0.01m)
Mid (Z=3.95m)
Top (Z=7.86m)
X (m)
Y (m)

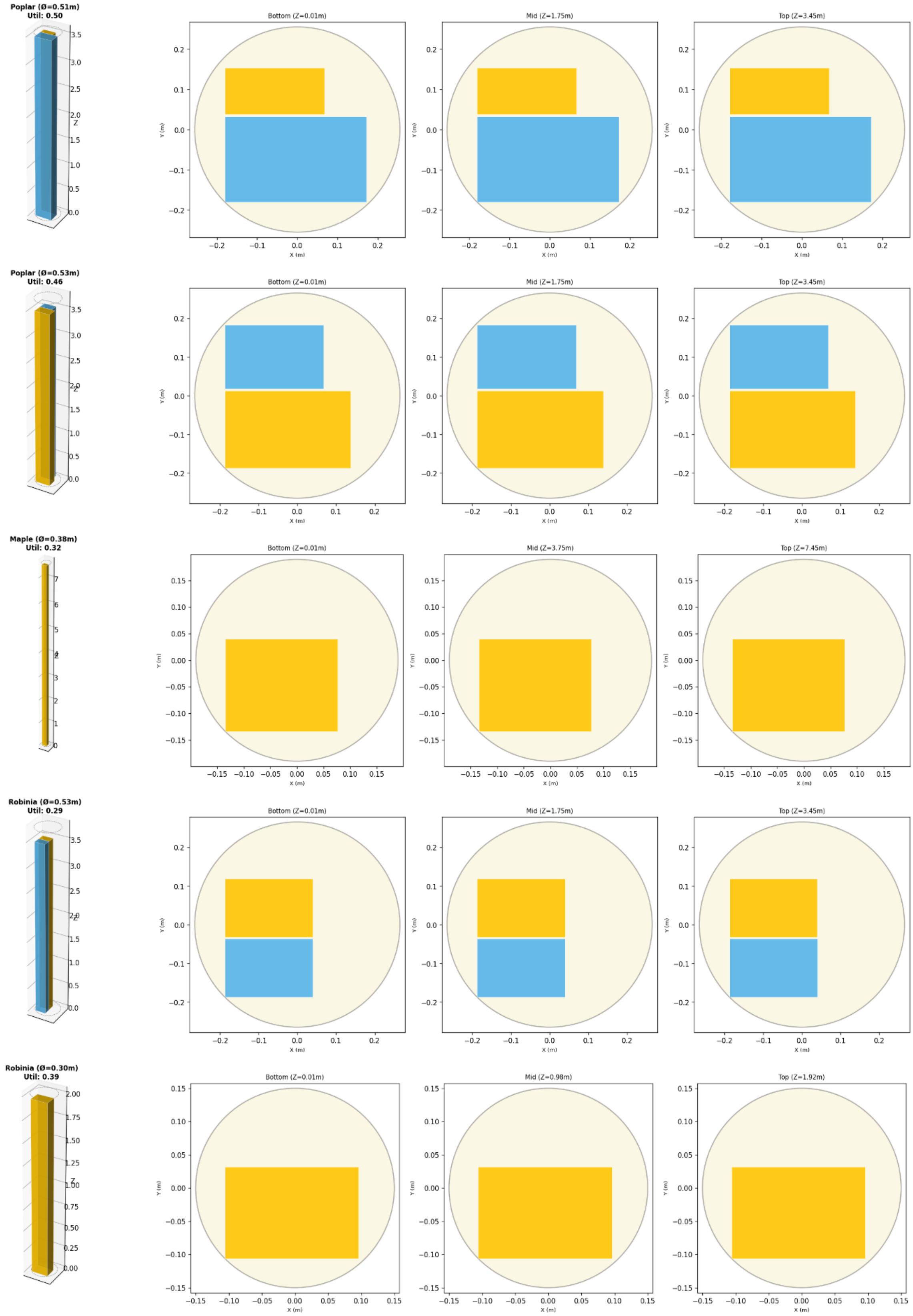

Maple (Ø=0.39m)
Util: 0.24
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
X (m)
Y (m)


Maple (Ø=0.38m)
Util: 0.41
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
X (m)
Y (m)

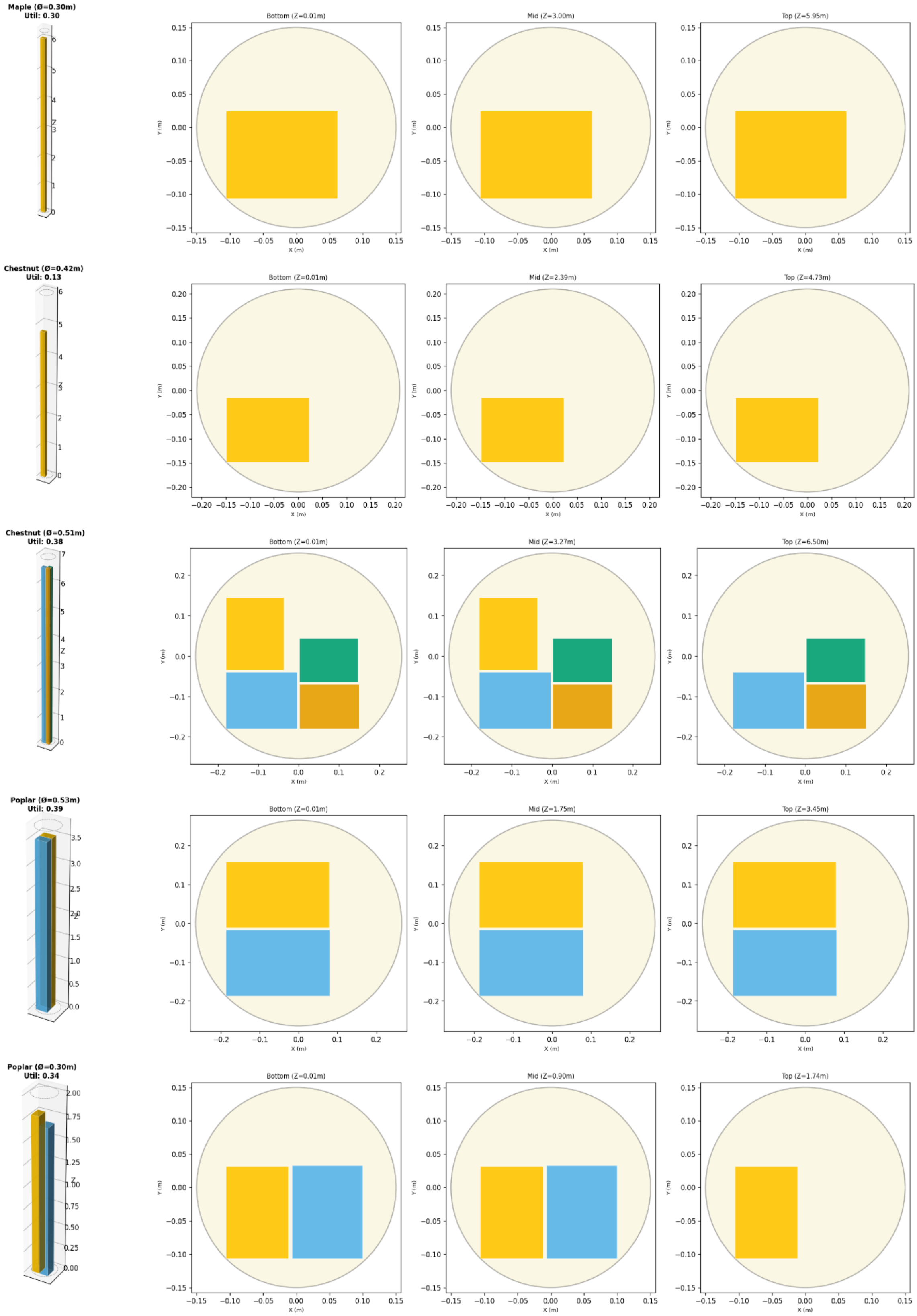

Maple (Ø=0.39m)
Util: 0.23
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
X (m)
Y (m)

Maple (Ø=0.40m)
Util: 0.36
Bottom (Z=0.01m)
Mid (Z=3.95m)
Top (Z=7.86m)
Maple (Ø=0.42m)
Util: 0.26
Bottom (Z=0.01m)
Mid (Z=4.10m)
Top (Z=8.15m)
Ash (Ø=0.35m)
Util: 0.40
Bottom (Z=0.01m)
Mid (Z=3.90m)
Top (Z=7.76m)
Poplar (Ø=0.39m)
Util: 0.45
Bottom (Z=0.01m)
Mid (Z=1.26m)
Top (Z=2.47m)
Poplar (Ø=0.48m)
Util: 0.28
Bottom (Z=0.01m)
Mid (Z=1.64m)
Top (Z=3.24m)
X (m)
Y (m)
Z


Poplar (Ø=0.52m)
Util: 0.47
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Poplar (Ø=0.51m)
Util: 0.57
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Robinia (Ø=0.40m)
Util: 0.27
Bottom (Z=0.01m)
Mid (Z=1.10m)
Top (Z=2.15m)
Poplar (Ø=0.51m)
Util: 0.54
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Poplar (Ø=0.40m)
Util: 0.23
Bottom (Z=0.01m)
Mid (Z=1.06m)
Top (Z=2.08m)
X (m)
Y (m)
Z


Poplar (Ø=0.51m)
Util: 0.50
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Poplar (Ø=0.53m)
Util: 0.46
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Maple (Ø=0.38m)
Util: 0.32
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
Robinia (Ø=0.53m)
Util: 0.29
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Robinia (Ø=0.30m)
Util: 0.39
Bottom (Z=0.01m)
Mid (Z=0.98m)
Top (Z=1.92m)
X (m)
Y (m)
Z

Oak (Ø=0.39m)
Util: 0.28
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
Maple (Ø=0.39m)
Util: 0.35
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
Robinia (Ø=0.52m)
Util: 0.45
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Ash (Ø=0.35m)
Util: 0.28
Bottom (Z=0.01m)
Mid (Z=3.95m)
Top (Z=7.86m)
Maple (Ø=0.39m)
Util: 0.19
Bottom (Z=0.01m)
Mid (Z=3.75m)
Top (Z=7.45m)
X (m)
Y (m)


Maple (Ø=0.30m)
Util: 0.30
Bottom (Z=0.01m)
Mid (Z=3.00m)
Top (Z=5.95m)
Chestnut (Ø=0.42m)
Util: 0.13
Bottom (Z=0.01m)
Mid (Z=2.39m)
Top (Z=4.73m)
Chestnut (Ø=0.51m)
Util: 0.38
Bottom (Z=0.01m)
Mid (Z=3.27m)
Top (Z=6.50m)
Poplar (Ø=0.53m)
Util: 0.39
Bottom (Z=0.01m)
Mid (Z=1.75m)
Top (Z=3.45m)
Poplar (Ø=0.30m)
Util: 0.34
Bottom (Z=0.01m)
Mid (Z=0.90m)
Top (Z=1.74m)
X (m)
Y (m)
Z

Cross-sections for FOR building, condensed