WORLD CONFERENCE ON TIMBER ENGINEERING 2025
BRISBANE, AUSTRALIA
Advancing Timber for the Future Built Environment

INVENTORY-CONSTRAINED DESIGN METHOD FOR WHOLE TREE USE IN BRANCHING CANOPY STRUCTURES

Yue FENG[1], Hengyi REN[2], Kim BABER[2,3], Canhui CHEN[4], Xingfei YUAN[1], Joseph M. GATTAS[*3]

ABSTRACT: Timber sourced from plantation forests is an in-demand resource, being natural, renewable, and possessing low embodied carbon. However, it is a finite resource, so the efficient material and structural use of timber in architecture is essential for maximising economic and carbon storage value. This paper introduces a digital design methodology for branching structures, with the aim of designing a branching structural system that supports a given canopy area and is constructed from a given inventory of whole trees. The proposed methodology encompasses three key stages: space-filling canopy distribution, branching topology generation, and inventory-constrained form-finding. The latter is based on the Combinatorial Equilibrium Method (CEM) and incorporates a material length constraint such that each structure uses a single entire tree from a forest inventory. The findings of this study support improved material design efficiency in wood and whole wood construction.

KEYWORDS: branching structures, space assignment, circle packing, inventory-constrained design, whole timber, Combinatorial Equilibrium Method

1 – INTRODUCTION

1.1 INVENTORY-CONSTRAINED DESIGN

Extensive research has been conducted on design processes for materially efficient building design, to minimise the environmental impact of structures. The majority of work has focused on volume or embodied carbon minimisation when using a range of typical building materials. However, emerging understanding of building circular design principles has led to recent focus on reuse of building materials [3]. In steel structures, Brütting et al. [4] introduced an inventory-constrained workflow designed to create structures from both reused and new elements, demonstrating that this combination significantly reduces embodied emissions.

Inventory-constrained design has also allowed building designers to develop a deeper understanding of material utilisation in the design and optimisation of wood structures [5, 7]. Baber et al. [1, 2] explored the design of an optimised branched timber structure that utilizes random length timber offcuts and minimally processed round logs in their near-to natural dimensions. Torghabehi et al. [20] proposed a computational workflow that uses the inventory of harvested tree crotches to inform the geometry of a dome, resulting in reduced use of raw materials. Several recent studies have considered design workflows for reuse of timber elements from deconstructed or demolished buildings [9, 10, 19, 24]. However, exploration of inventory-constrained design in building architecture and structural engineering remains limited, especially when considering the constraints of whole-timber consumption.

1.2 FORM-FINDING METHODS FOR BRANCHING STRUCTURES

Branching structures are a bio-inspired building form comprising the beautiful, forked and irregular arrangement of structural elements [16]. They are typically developed as compressive-only load-bearing systems [23] and as such, form-finding methods can be applied to their analysis. Developed approaches include the force density method [27], dynamic relaxation method [22], and inverse-hang recursive method [28]. Based on graphic statics, Xu et al. [25] presented a numerical form-finding and shape optimization method for bio-inspired branching structures. Research by Tu et al. [21] used potential energy as the evaluation indicator for structural efficiency, which was minimised from iterative update of the branching model shape. Du et al. [8] utilised machine learning to resolve the shape-finding problem of tree-like structures under a non-uniform load. Zhao et al. [26] introduced the double-element method for rapid, iterative form-finding analysis of tree structures.

In the domain of interactive form-finding, Combinatorial Equilibrium Modeling (CEM) [13, 15] is a theoretical framework based on graphic statics and graph theory, which allows for consideration and control of qualitative structural behavior in the conceptual design phase. CEM is well suited for the analysis of structures composing link elements with axial tension and compression forces, in-

[1]College of Civil Engineering and Architecture, Zhejiang University, Hangzhou, China
[2]School of Architecture, Design, and Planning, The University of Queensland, Brisbane, Australia. j.gattas@uq.edu.au
[3]School of Civil Engineering, The University of Queensland, Brisbane, Australia. j.gattas@uq.edu.au
[4]School of Design and Architecture, Swinburne University of Technology, Melbourne, Australia. j.gattas@uq.edu.au

https://doi.org/10.52202/080513-0487

cluding truss, cable-strut, and branching structures. CEM has also been applied to incorporate inventory constraint in the design of structures from reused elements [4], however there is still limited research on interactive form-finding incorporating inventory constraint.

1.3 RESEARCH CONTRIBUTION

This paper aims to develop a novel computational design framework for the interactive structural design of branching structures supporting a loaded roof canopy, strictly constructed from a given inventory of whole trees. A structure of this type is shown in Figure 1 [6], comprising three branching structures, each made from one whole tree, supporting a 61 m^2 irregular roof canopy.

The proposed framework arises from the novel integration of three processes: space-filling canopy distribution, branching topology generation, and inventory-constrained form-finding. A space-filling algorithm, Section 2, uses a circle packing method to solve the problem of uniform distribution of support points across a supported canopy of irregular area. A branch topological generation algorithm, Section 3, is then introduced to progressively aggregate the top points into supporting branches and tree fork connections, resulting in a finite set of tree structures providing complete canopy support. An inventory-constrained CEM form-finding process, Section 4, is finally applied to the tree structure geometries to find an equilibrated compressive-only structural form, while satisfying a whole-tree utilisation constraint from an input forest inventory. The integrated framework is summarised in Figure 2.

2 – SPACE-FILLING CANOPY DISTRIBUTION

2.1 SPACE-FILLING ALGORITHMS

Space-filling algorithms are methodologies used to efficiently cover or fill a given space, whether a 2D plane, 3D volume, or higher dimension. The current landscape of packing and filling algorithms spans various innovative techniques. The Hilbert curve [11] is a continuous fractal space-filling curve that can traverse all points in a square grid. It can be expressed by a parallel rewrite system, such as a Lindenmayer system (L-system) [12], for generative creation of patterns that fill regular spaces. The space colonization algorithm is a bio-inspired algorithm

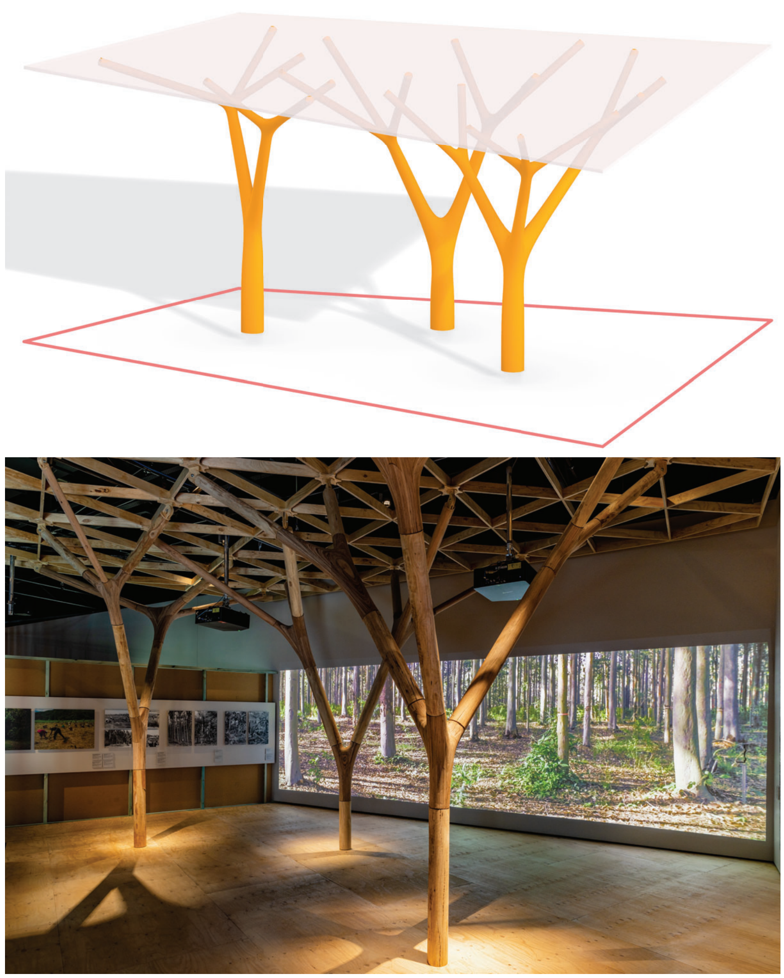

Figure 1: Forest to Fibre branching canopy pavilion. Design model (top) and photo of installed structure (bottom). State Library Queensland, 2023.

https://doi.org/10.52202/080513-0487

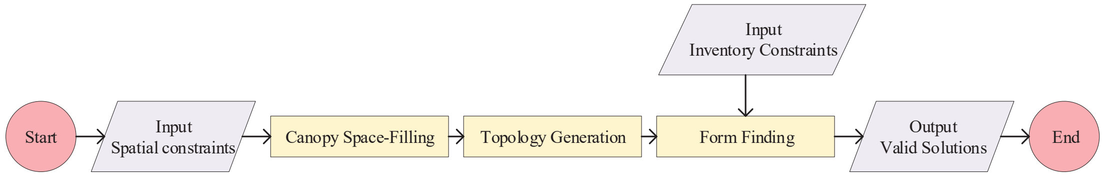

Figure 2: computational design framework for the interactive structural design of branching structures.

used to model branching structures [17]. It can describe an iterative method for growing networks of branches based on the placement of organic materials based on attraction points. Circle packing algorithms, developed primarily for packing problems [18], are also often adapted for space-filling purposes. It can fill a specific area by iteratively moving or adding circles, until they do not overlap and are tangent with other circles.

For the present context, if a roof canopy area is supported by an underlying branching structure at uniformly-distributed points, the vertical load distribution into the structure can be assumed as uniform to simplify structural design and optimisation tasks. As such, the branching structure design framework follows a top-to-bottom approach by first employing a circle packing algorithm to generate a uniform distribution of support points across any regular or irregular canopy area.

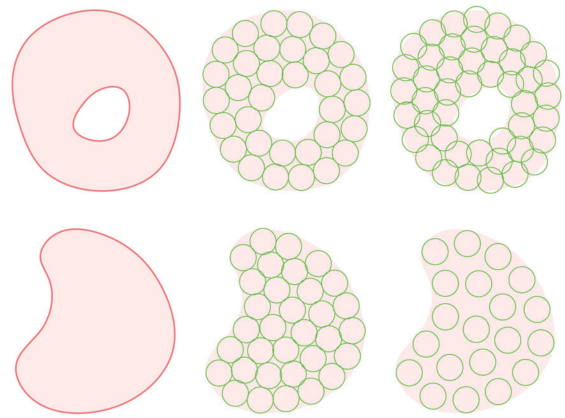

Figure 3: Irregular roof areas (left) with equidistant circle packing (middle). Over and under-filled circle packing (right).

2.2 CANOPY POINT CIRCLE PACKING

In practical design scenarios, irregular boundaries of a roof canopy may arise due to the influence of surrounding buildings or special design considerations. An irregular roof area in this study, defined in response to such constraints, is specified as being bounded within or between closed planar spline curves, Figure 3 (left). The maximum number of uniformly-distributed support points n_t which can be distributed across an irregular area can be expressed as:

$$n_t < \frac{A}{\pi r^2} \tag{1}$$

where A is the canopy area and r is influential range of each point, that is r inscribes a uniform circular area supported by each point.

The circle packing problem solver was implemented using the *BouncySolver* and *Collider* in Rhino/Grasshopper. In the solver program, circle packing solutions can be obtained by iteratively moving circles within the specified roof area boundary curves until the a best solution is obtained, when the irregular roof area is fully filled with equidistant circles, Figure 3 (middle). A proper pattern of packed circles could be found by adjusting n_t, to avoid over- or under-filling the roof area, Figure 3 (top right) and (bottom right), respectively.

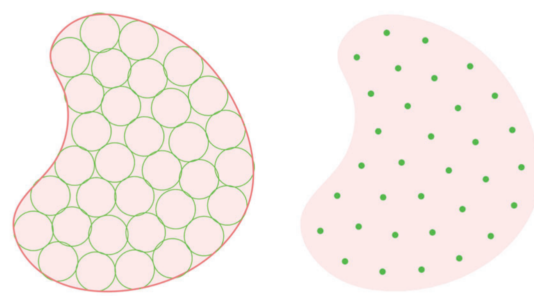

Figure 4: Irregular roof area filled with 33 equal circles (left) and top point distribution (right) after circle packing.

3 – BRANCHING STRUCTURE TOPOLOGY GENERATION

3.1 POINT CLUSTERING

Once the quantity and position of canopy support points are determined, the topology of the branch structure itself can be generated from the clustering of points into high-level supporting branches. These are then also recursively clustered until reaching support locations at the ground level. Such point clusters are normally nominated subjectively based on judgment and design, as computational clustering strategies quickly encounter combinatorial explosion if seeking to list all available point combinations. Consider the design scenario in Figure 4 as an example, containing 33 support points in the canopy area.

If these branches are clustered such that they support 3 points each, the number of potential combinations n_c can be obtained as:

$$n_c = C(n_t, n_b) = \frac{n_t!}{n_b!\,(n_t - n_b)!} = \frac{33!}{3!\,(33-3)!} \tag{2}$$

which is 5456 for the above case. Many of the generated combinations are highly unsuited as structural support arrangements, for example, widely spaced point groups, and overlapping, or unstable, near-linear groupings.

To obtain valid solutions and more efficiently understand the branched canopy support design space, a geometrical filtering process is developed. Filter conditions are

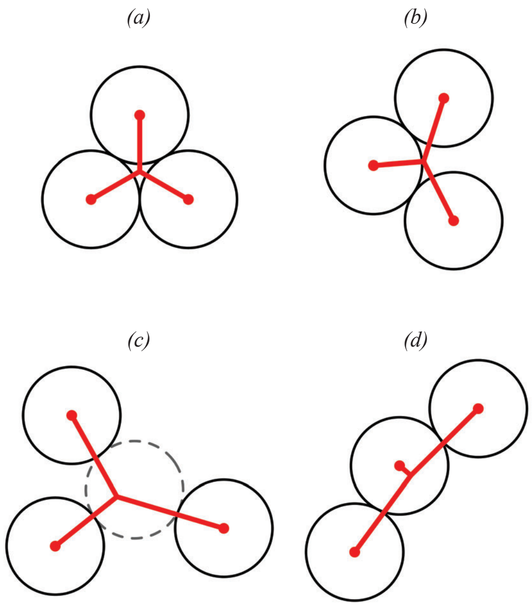

Figure 5: Valid (a, b) and invalid (c,d) 3-point clusters.

applied to generated combinations to eliminate the large proportion of invalid solutions. The first filter condition imposes a limit on maximum distance D between clustered points:

$$\sum_{i=1}^{3} d_i \leq D\,(D > 6r) \tag{3}$$

where d_i represents side lengths in the triangle formed by three clustered points. The second filter condition imposes a limit on angular dispersal Θ between clustered points:

$$\cos\left(\max\left\{\theta_j\right\}\right) > \Theta\,(-1 < \Theta < 1) \tag{4}$$

where θ_j represents internal angles in the triangle formed by three clustered points, with j = 1,2,3. Example three-point clusters are shown in Figure 5. Valid clusters are shown in (a) and (b) and invalid clusters are shown in (c) and (d), from Eq. (3) and (4), respectively.

A recursive algorithm was developed in Matlab to find valid point clusters, summarised as Algorithm (1). From a list of canopy support points, a matrix containing all valid 3-point clusters is generated through filtering all combinations according to the "(3)" and "(4)". The valid matrix will be input into the algorithm as $C_{n\times3}$, which represent n valid 3-point clusters. The output is a matrix S_{all} containing all valid solutions. For the above canopy case, with $D = 7.2r$ and $\Theta = 100°$, there are 77 valid solutions for grouping 33 points into 11 point clusters. Five valid solutions are shown in Figure 6. Three clustered points are connected to a fourth *anchor* point, located at their geometric centre.

3.2 BRANCHING STRUCTURE TEMPLATE

The anchor points created by the point clustering method correspond to the required support points for the next-lower layer of a branch structure topology. This is

Algorithm 1 Valid Solutions from Point Clusters

Input: $n_c, n_s, C, S_{current}, P_{unique}, row_{start}$
Output: S_{all}

n_c: number of combinations
n_s: number of combinations in each solution
$C_{n\times3}$: expression matrix for n 3-point clusters
$S_{current}$: an empty matrix to store current solutions
P_{unique}: an empty matrix to record point labels
row_{start} = the start coefficient of the recursion
S_{all} : all valid solutions of points clusters.
function RECURSION($C, n_s, n_c, S_{current}, row_{start}, P_{unique}$)
  if $n_s = 0$ **then**
    $S_{all} = S_{current}$
    return S_{all}
  end if
  S_{all} = null
  for $i = row_{start} \rightarrow n$ **do**
    if $C_i \cap P_{unique}$ = null **then**
      Add the points labels in current combination C_i to P_{unique}
      S_{new} = RECURSION(C, n_s - 1, n_c, [$S_{current}$ C_i], i + 1, P_{unique})
      Add the S_{new} to S_{all}
      Remove the last added points labels from P_{unique} to explore other solutions
    end if
  end for
end function

shown as a simplified 2D arrangement in Figure 7. Point clustering is applied through sequential layers of the structure until reaching ground level, to create a branching structure *template*. This is defined as a preliminary diagram of lines and points, storing the branching structure height and initial topology. As the template is only generated from geometrical point clustering, architects and designers can rapidly explore and compare alternative clustering arrangements, before it it is conveyed to the form-finding process developed in the next section.

To enrich the range of available topologies, the point clustering algorithm as applied to structure sub-layers is improved with two features. First, output solutions can be generated with varied numbers of points per cluster. Examples with 2-point and 3-point clusters are shown in Figure 8. Second, an additional filtering criteria is added to prevent overlapping between separate template groups. An example impermissible overlapping grouping is shown in Figure 8 (right).

Continuing the Figure 6 case study with 11 canopy point clusters, Figure 9 (top row) shows four potential sub-layer clusters: with [2,3,3,3] (left two) or [2,2,2,2,3] (right two) points per cluster. 3D templates are generated from height input, Figure 9 (bottom row), with 4 or 5 branching structures required to support the canopy. As there is no overlap between clusters as assigned in 2D, the structure branches will never collide or overlap in 3D.

4 – INVENTORY-CONSTRAINED FORM-FINDING BASED ON CEM

4.1 LOADING AND SUPPORT ASSUMPTIONS

In nature, the branches of a tree are rigidly connected to each other and capable of transmitting axial, shear, and flexural loads. However, for most constructed branching structures, the structural form is typically designed to carry axial forces in the components only, to both simplify con-

https://doi.org/10.52202/080513-0487

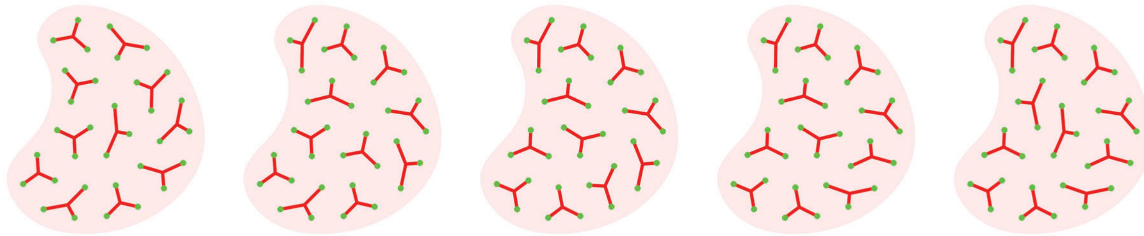

Figure 6: Valid solutions for canopy point clusters (top view).

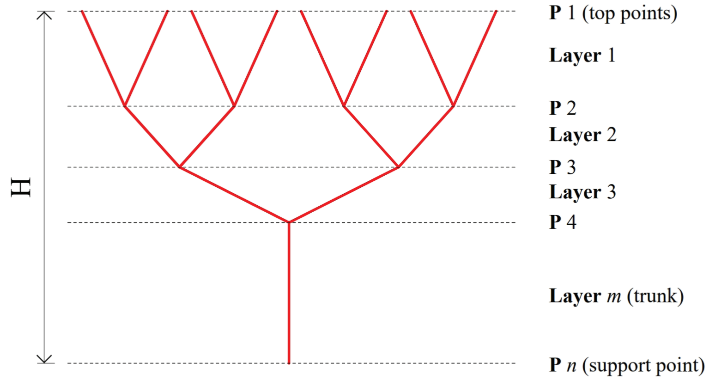

Figure 7: Branching structure template (elevation view).

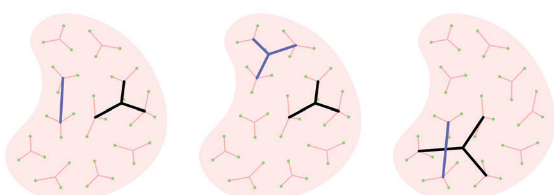

Figure 8: Valid (left and middle) and invalid (right) solutions of lower-layer points clustering (top view).

struction and enhance structural efficiency.

As the final stage of the framework, the branch structure templates are processed through a numerical form-finding tool to establish an optimised structural form for compressive-only load transfer. Branch components are considered as linear 3-dimensional compressive elements and point clusters are considered as pinned joints between connected members. The canopy is modelled as a single layer network of tensile elements, connecting and stabilising all canopy points in the roof plane. A homogeneous load is applied to the canopy points as vertical point loads. The lower end of the trunk is considered as a pinned support to the ground plane.

4.2 WHOLE-TREE INVENTORY CONSTRAINT

To maximize the efficiency of timber use in the branching structure, a whole timber length constraint is introduced before the form-finding analysis. The height of each layer in the branching structure template is changed such that total length of the structure matches a target length of a whole tree. Taking a template from Figure 9 as an example, the relationship between layer heights and total lengths with same total height is shown in Figure 10. The three axes of the figure represent the three layer heights (h_1, h_2, h_3) of the structure. A structure with fixed total height h_t occurs at any point on a plane P, expressed as:

$$\sum_{i=1}^{3} h_i = h_t \tag{5}$$

The total length l_t of the logs used in the structure is represented by any point on the surface S, expressed as:

$$\sum_{i=1}^{3} l_i = l_t \tag{6}$$

in which l_i represents the length of logs used in ith layer of the branching structure, calculated as a function of co-ordinates of intersection points.

Obviously, there are many factors that affect the total length of the template, including structural topology, form, layer height, total height, etc. For a fractal branching structure, the relationship between these factors and the total length is nonlinear, but the trend of these variables affecting the total length can be seen from Figure 10. Overall, a larger total height and upper layer heights most strongly influence total length. The line of intersection between plane P and surface S represents the structure that can meet the requirement of an inventory with a specific length and structure with a specific height.

https://doi.org/10.52202/080513-0487

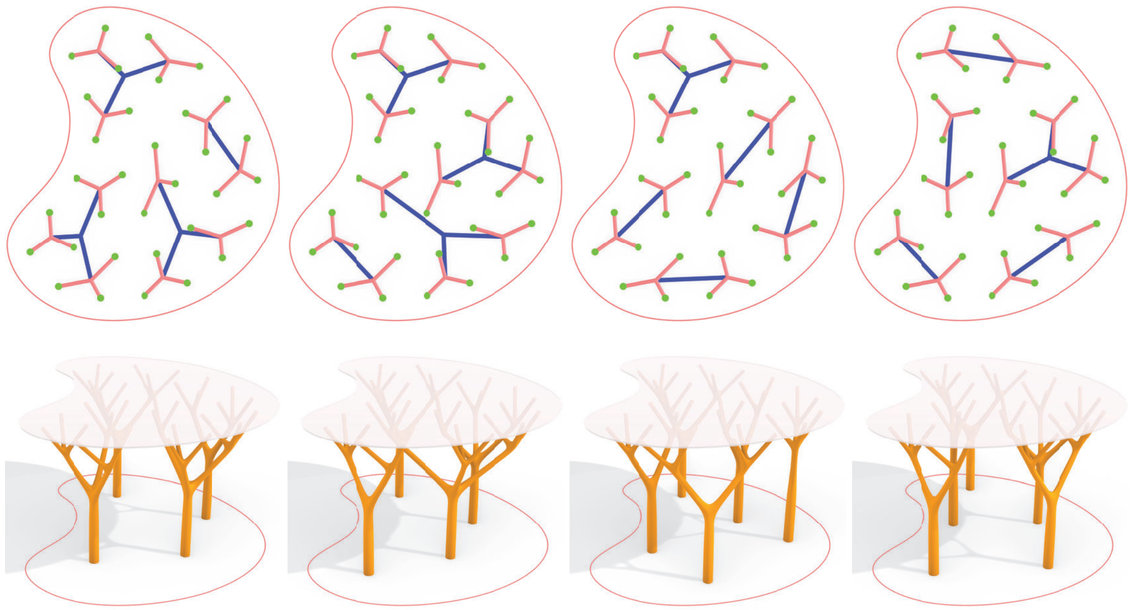

Figure 9: Top view (top) and 3D view (bottom) of 4-tree solutions (left two) and 5-tree solutions (right two).

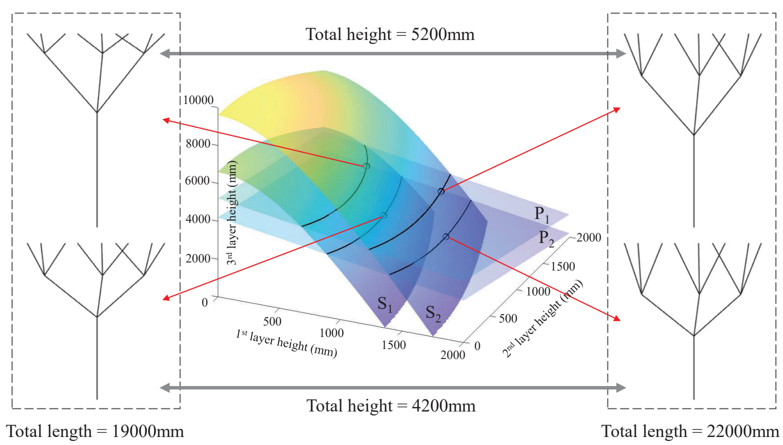

Figure 10: Relationship between total length and layer height for a single tree structure. P1 and P2 at h_t = 4.2 and 5.2 m, respectively. S1 and S2 at l_t = 19 and 22 m, respectively.

4.3 FORM-FINDING

As a structure composed only of line elements, the form-finding process of each individual branching structure can be analyzed using CEM. After completion of the preceding steps, the template that meets the inventory requirements can be finally input into the COMPAS CEM Rhino/Grasshopper package [14] for structural form-finding. CEM optimizes the coordinates of points in the structure sub-layers to satisfy the aforementioned compressive equilibrium and support conditions. Example input forms (in green) and optimized forms (in brown) of selected templates are shown in Figure 11. Optimized branch locations tend to move inward, to achieve an equilibriated state.

While visually, the branching forms change a lot from form finding, the total length of elements is not significantly affected. Table 1 summarises the length change in pre- and post-optimised templates shown in Figure 11 (tree1 to tree4, left-to-right). Values listed for shape1 to shape4 represent different initial total height conditions. Total length change in all cases is less than 10% and always shorter than the target length, which maintains the validity of the inventory constraint.

5 – CONCLUSION

This paper proposed an inventory-constrained design method for use of whole trees in branching canopy structures. The integration of a canopy space-filling circle packing algorithm, recursive clustering topology generation method, and inventory-constrained form finding method created a digital design workflow with multiple points for intuitive designer control over branching structure form and function. This supports wide design exploration while remaining within a valid, structurally efficient, and materially efficient design space.

https://doi.org/10.52202/080513-0487

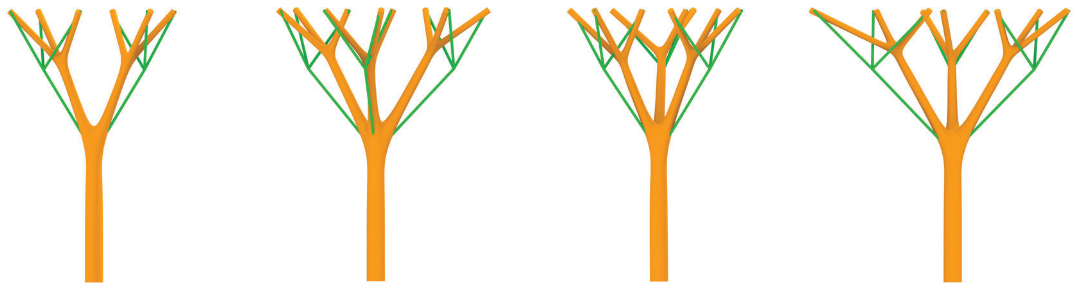

Figure 11: Single trees in previous forms (in green) and optimized forms (in yellow) from Figure 9 (left).

Table 1: Total lengths of trees in different shapes (mm).

height	items	tree1		tree2		tree3		tree4	
		shape 1	shape 2	shape 1	shape 2	shape 1	shape 2	shape 1	shape 2
5200	previous	12915	14836	13119	14980	12944	14800	13075	14947
	optimized	12561	14269	12868	14384	12823	14462	12770	14317
	difference	-354	-567	-251	-596	-121	-338	-305	-630
	rate	-2.7%	-3.8%	-1.9%	-4.0%	-0.9%	-2.3%	-2.3%	-4.2%
		shape 3	shape 4	shape 3	shape 4	shape 3	shape 4	shape 3	shape 4
4200	previous	12651	14581	12812	14697	12581	14459	12773	14666
	optimized	11973	13593	12117	13581	12149	13713	12103	13630
	difference	-678	-988	-695	-1116	-432	-746	-670	-1036
	rate	-5.4%	-6.8%	-5.4%	-7.6%	-3.4%	-5.2%	-5.2%	-7.1%

ACKNOWLEDGEMENTS

The first author gratefully acknowledges funding from the China Scholarship Council (CSC) to support his visiting research student placement at the University of Queensland. K. Baber and J. Gattas would also like to acknowledge research support provided by the Australian Research Council Research Hub to Advance Timber for Australia's Future Built Environment (project number IH220100016), funded by the Australian Government.

REFERENCES

[1] K. R. Baber, A. Barton, J. R. Burry, C. Chen, J. M. Gattas, C. Koch, and H. Ren. "Inventory-constrained design of a variable small diameter round timber structure." In: *Proceedings of IASS Annual Symposia*. Vol. 2023. 2. International Association for Shell and Spatial Structures (IASS). 2023, pp. 1–12.

[2] K. R. Baber, J. R. Burry, C. Chen, J. M. Gattas, and A. Bukauskas. "Inventory constrained design of a timber funicular structure." In: *SN Applied Sciences* 2 (2020), pp. 1–19.

[3] J. Brütting, J. Desruelle, G. Senatore, and C. Fivet. "Design of truss structures through reuse." In: *Structures*. Vol. 18. Elsevier. 2019, pp. 128–137.

[4] J. Brütting, P. O. Ohlbrock, J. Hofer, and P. D'Acunto. "Stock-constrained truss design exploration through combinatorial equilibrium modeling." In: *International Journal of Space Structures* 36.4 (2021), pp. 253–269.

[5] A. Bukauskas, P. Shepherd, P. Walker, B. Sharma, and J. Bregulla. "Inventory-constrained structural design: new objectives and optimization techniques." In: *Proceedings of IASS annual symposia*. Vol. 2018. 11. International Association for Shell and Spatial Structures (IASS). 2018, pp. 1–8.

[6] C. Chen, K. R. Baber, J. M. Gattas, C. A. Nygaard, and M. Burry. "From Wastewater to Building: Harnessing digital fabrication for circular economy innovations." In: *Proceedings of the International Conference for the Association for Computer-Aided Architectural Design Research in Asia 2025 (CAADRIA2025), Japan.* 2025.

[7] C. Cristescu et al. "Design for deconstruction and reuse of timber structures–state of the art review." In: (2020).

[8] W. Du, Y. Zhao, Y. Wang, H. Wang, and M. Yang. "Novel machine learning approach for shape-finding design of tree-like structures." In: *Computers & Structures* 261 (2022), p. 106731.

[9] S. M. Haakonsen, A. Tomczak, B. Izumi, and M. Luczkowski. "Automation of circular design: A timber building case study." In: *International Journal of Architectural Computing* (2024), p. 14780771241234447.

[10] Y. Huang, L. Alkhayat, C. De Wolf, and C. Mueller. "Algorithmic circular design with reused structural elements: method and tool." In: *Proceedings of the International fib Symposium on the Conceptual Design of Structures*. Vol. 55. Fédération Internationale du Béton (fib). 2021, pp. 457–468.

[11] H. V. Jagadish. "Analysis of the Hilbert curve for representing two-dimensional space." In: *Information Processing Letters* 62.1 (1997), pp. 17–22.

[12] A. Lindenmayer. "Mathematical models for cellular interactions in development I. Filaments with one-sided inputs." In: *Journal of theoretical biology* 18.3 (1968), pp. 280–299.

[13] P. Ole Ohlbrock and J. Schwartz. "Combinatorial equilibrium modeling." In: *International Journal of Space Structures* 31.2-4 (2016), pp. 177–189.

[14] R. Pastrana, P. O. Ohlbrock, P. D'Acunto, and S. Parascho. *COMPAS CEM: The Combinatorial Equilibrium Modeling framework for COMPAS*. 2021. doi: 10.5281/zenodo.5705740. url: https://arpastrana.github.io/compas%5C_cem.

[15] R. Pastrana, P. O. Ohlbrock, T. Oberbichler, P. D'Acunto, and S. Parascho. "Constrained Form-Finding of Tension–Compression Structures using Automatic Differentiation." In: *Computer-Aided Design* 155 (2022). issn: 0010-4485. doi: 10.1016/j.cad.2022.103435. url: https://www.sciencedirect.com/science/article/pii/S0010448522001683.

[16] I. M. Rian and M. Sassone. "Tree-inspired dendriforms and fractal-like branching structures in architecture: A brief historical overview." In: *Frontiers of Architectural Research* 3.3 (2014), pp. 298–323.

[17] A. Runions, B. Lane, and P. Prusinkiewicz. "Modeling trees with a space colonization algorithm." In: *Nph* 7.63–70 (2007), p. 6.

[18] W. Thurston. "The finite Riemann mapping theorem." In: *An International Symposium at Purdue University on the Occasion of the Proof of the Bieberbach Conjecture, 1985*. 1985.

[19] A. Tomczak, S. Haakonsen, and M. Łuczkowski. "Matching algorithms to assist in designing with reclaimed building elements." In: *Environmental Research: Infrastructure and Sustainability* 3.3 (2023), p. 035005.

[20] O. O. Torghabehi, P. von Buelow, S. Mankouche, and K. Vliet. "Limb: Inventory-constrained design method for application of natural tree crotches as heavy timber joinery." In: ().

[21] G. Tu and C. Chen. "A shape-finding approach of tree-like structure based on grouping strategy and generalized inverse matrix theory." In: *Structures*. Vol. 29. Elsevier. 2021, pp. 1960–1966.

[22] G. Tu, C. Chen, Z. Gong, and Y. Wang. "A Form-Finding Method for Branching Structures Based on Dynamic Relaxation." In: *Applied Sciences* 11.15 (2021), p. 7165.

[23] P. Von Buelow. "A Geometric Comparison of Branching Structures in Tension and in Compression versus Minimal Paths." In: (2007).

[24] J. Warmuth, E. X. Estrella Arcos, B. Kälin, J. Natterer, and C. Fivet. "Reuse of Load-Bearing Timber Elements-Case Study of a Lookout Tower in Lausanne, Switzerland." In: *World Conference on Timber Engineering (WCTE 2023). Timber for a livable future*. Curran Associates, Inc. 2023, pp. 3600–3607.

[25] C. Xu, Z. Wang, B. Li, and Q. Liu. "Form-finding and shape optimization of bio-inspired branching structures based on graphic statics." In: *Structures*. Vol. 29. Elsevier. 2021, pp. 392–407.

[26] Z. Zhao, B. Liang, H. Liu, and Q. Sun. "A novel numerical method for form-finding analysis of branching structures." In: *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 39 (2017), pp. 2241–2252.

[27] Z. Zhao, D. Yu, T. Zhang, and H. Gao. "Form-finding and optimization for free form grid structures supported by branching columns based on updated force density method." In: *Structures*. Vol. 44. Elsevier. 2022, pp. 1190–1203.

[28] Z. Zhao, D. Yu, T. Zhang, N. Zhang, H. Liu, B. Liang, and L. Xian. "Efficient form-finding algorithm for freeform grid structures based on inverse hanging method." In: *Journal of Building Engineering* 46 (2022), p. 103746.

 https://doi.org/10.52202/080513-0487