



Constructability-driven design of frame structures with state-space search methods

Yijiang Huang ^{a,b,*}, Caelan Garrett ^c, Caitlin Mueller ^b

^a ETH Zurich, Zurich 8092, Switzerland

^b Massachusetts Institute of Technology, Cambridge 02139, MA, USA

^c NVIDIA Research, Seattle 98105, WA, USA

ARTICLE INFO

Keywords:

Frame
Sequencing
Scaffold-free
Assembly
Heuristic search

ABSTRACT

In the design of frames and trusses, the relationship between the structural form, the construction sequence, and the structural behavior during construction is rarely systematically considered. This paper proposes a method to systematically consider constructability in design, specifically focusing on minimizing the maximum displacement or stress during construction. The paper presents the formulation of the optimal assembly sequencing problem as a state-space search with a unique design of the search heuristic and constraint bounding schemes. It proposes three variants of heuristic search algorithms for finding a feasible sequence, a diverse set of feasible sequences, and an optimal sequence, respectively. These algorithms are tested on a case study structure to showcase their ability to navigate the combinatorial space of assembly sequences, and provide means for designers to incorporate sequence-related performance into conceptual structural design.

1. Introduction

When designing discrete-element-based structures such as trusses and frames, it is common to consider mechanical behavior (stiffness and strength) and material efficiency to inform early-stage design decisions. However, another key aspect of a structure's performance is its constructability: how challenging is it to build this design in terms of cost, time, and logistical complexity? Generalized constructability metrics are notoriously challenging to define quantitatively because they depend heavily on system typology and other problem specifics, and because many aspects of traditional construction are unplanned during design or handled in the field by humans, and are therefore unpredictable. This means that many design concepts that perform well in terms of mechanical behavior and efficiency may in turn be challenging to construct, and the potential cost savings offered during design optimization are not captured in reality.

In response, this paper proposes a method to systematically consider constructability in design, specifically focusing on sequence-related performance of the assembly of the bar structure. Construction sequencing determines the order of building a structure from discrete components from a given kit of parts. The construction of trusses, frames, and more complex structural hierarchies, for example those

involving pre-cast elements, all require detailed sequence planning [1–3]. This is a universal and important aspect of the construction planning process, as it has a profound impact on the stiffness and stability of the partially built structure. This work develops algorithmic tools to (1) assist in the decision making of physics-driven construction sequencing and (2) propose a sequence-related performance metric to inform early-stage structural design.

Systematically considering performance early in the design process typically requires the development of quantification, calculation, and computer simulation of the desired performance metric. While finite-element-based simulation tools have been widely used to analyze and inform the design of structures in their completed stages (e.g., [4]), the analysis of sequential structural behaviors during construction has not been thoroughly understood. The coupling between combinatorial decision making and structural analysis in construction sequencing requires a fundamentally different set of mathematical formulations and computational tools.

We focus our attention on a simplified yet challenging type of construction process: monotonic assembly of discrete bar structures without scaffolding, where elements are added to the partial construction one at a time (defined precisely in section 3.1). In this work, we use the term “bar” in a general sense to refer to any linear structural element,

* Corresponding author at: ETH Zurich, Zurich 8092, Switzerland.

E-mail address: yijiang.huang@inf.ethz.ch (Y. Huang).

including truss elements that carry only axial forces but also frame elements that can carry bending and torsional loads. Scaffolding is defined as a temporary structure that supports the partially built structure and does not become part of the final structure. While not all structures are built without scaffolds in practice, many types, e.g., spanning systems like bridges or discrete shell structures, could benefit from the reduction of scaffolds to increase the operational and material efficiency of the construction process (see section 2.3 for more discussions). In this context, we aim to equip designers with computational tools for answering the following questions in the conceptual design stage:

- Is design in consideration buildable?
- What's the best way to build it?
- Are there other design options that are easier to build?

In this paper, we focus on finding assembly sequences whose structural behavior, such as maximum displacement or stress, during construction could be controlled. To help answer the questions above, we first propose a general framework to formulate a performance metric to describe the objectives and constraints of a nominated construction sequence of a given design. Then, we devise computational means to (1) find a feasible construction sequence, (2) find a diverse set of feasible sequences, and (3) find the optimal sequence. The proposed automated search routine allows us to find high-performing construction sequences when design complexity grows beyond human capacity to figure out the sequence by hand. The constructability score of a design can be computed by summarizing the performance of its construction sequences. This score can then be used to quantify the “buildability” and “easiness of building” of different options in a design catalog.

The core contributions of this work include:

- **Quantifying construction sequence's performance:** We present several formulations to evaluate sequence performance that involve the structure's material and mechanical behavior during construction (section 3).
- **Constructability score formulation:** We present two formulations of a constructability score, one for feasibility and the other for optimality, for a given design by summarizing the performance of its various construction sequences (section 4.1).
- **Computing feasible, diverse, and optimal sequences:** We present three search algorithms, two of them newly proposed, to find construction sequences for various purposes (section 4.2).
- **Constructability-driven design space exploration:** We present a case study that illustrates how we can use the proposed constructability score to inform design decisions on a discrete design catalog (section 5.2).

2. Related work

2.1. Construction scheduling

Research in automated construction scheduling aims to sequence job tasks and assign resources, while satisfying task precedence constraints and resource conflicts among tasks [5]. Some previous work investigates optimizing the assembly sequences of individual elements on a building scale, in the context of pre-cast concrete panels [6], combining crane operation planning and precast element sequencing [2], and the statically stable assembly of steel frames [7]. These methods use soft computing methods, such as the genetic algorithms to optimize assembly sequences while applying several geometric rules to ensure the stability of the partially constructed structure. Historically, it has been difficult for these algorithms to scale to structures with hundreds of elements and to have global optimality guarantees. Furthermore, previous work uses geometric rules to enforce structural safety, which can only serve as a proxy for detailed physics simulation. Although we focus on the construction of bar structures in this work, our algorithmic

framework is efficient, physics-based, and flexible. Thus, it can be extended to address construction planning problems in other contexts.

2.2. Assembly planning for manufacturing

Assembly planning aims to find sequences of operations to assemble parts of a structure (*assembly sequencing* [8]) and to determine the motions that bring each part to its target state (*assembly path planning* [9]). The literature in this domain mainly targets mechanical part manufacturing and focuses on three classes of constraints (also called predicates): liaison constraints that ensure contact between the joining parts and the existing subassembly, geometric constraints that ensure a collision-free insertion path, and stability constraints that check the rigid-body stability of the subassembly [10].

In contrast to the commonly considered stability constraint, we consider a new type of constraint, the stiffness constraint, which concerns a global elastic behavior on the sub-assembly, and thus cannot be encoded as a matrix *a priori*, as commonly done in the literature [11]. Our problem does not assume that the assembled parts are rigid, but can undergo elastic deformation due to self-weight. In addition, since the discrete-element system we consider is already in the form of a spatial line graph, the connectivity between parts can be directly inferred from the structure itself, and thus we do not need to generate or pre-specify the liaison predicates as in mechanical assembly sequence planning [12].

Finding a feasible assembly plan requires identifying movable parts and part groups in each intermediate assembly state. Existing work proposes graphical methods ([13]-chap 4.2, [14]), soft computing methods [10], and hierarchical methods [15]. We refer interested readers to [10] for a review of these efforts. However, the search space grows exponentially with the number of parts, and thus most work limits to design instances with at most a few dozen parts. One way to limit the large branching factor is the idea of assembly-by-disassembly; it has been widely used to limit geometric constraint violations, where an assembly plan is obtained by disassembling the completed assembly into parts and then reversing the order and path of disassembly [16–18]. Due to the challenges outlined above, most of the literature in assembly planning focuses on *sequential* and *monotonic* assembly that only allows one part to move at each step and parts to be removed once added to the sub-assembly. Our work addresses this category, but we aspire to plan sequences for structures with hundreds of elements and additionally focus on the stiffness constraint and explore the connection between the design variation and assembly plans. Recent work investigates non-monotonic planning [19] and uses physics simulation to help check the disassemblability of a sub-assembly [18,20].

Assembly planning can be formulated as an optimization to find the desired assembly plan. Typical optimization objectives include minimizing assembly time and cost [10], minimizing the usage of additional resources (e.g., fixtures or supports to maintain the stability of partial assemblies [21]), and maximizing part visibility for creating visual assembly instructions [22,23]. Due to the large search space, existing algorithms either require annotated precedence constraints to find optimal sequences [24,25], or they can only find suboptimal sequences, e.g., using a greedy algorithm [21,26,27], a heuristic search [22], or adaptive sampling followed by user editing [28]. Our work uses a heuristic search strategy, but our novel design evaluation framework for physics-constrained assembly problems can be used with other assembly planning algorithms.

2.3. Scaffold-free construction

Techniques for saving scaffolding material during the construction of spanning structures have attracted interest from ancient times to the present day. We review historical and recent efforts to build structures from a discrete set of elements or modules without scaffolding, including masonry vaults, precast modules shells, trusses, and frame structures.

Vaults and shells. Vaulted structures are historically built using reusable formwork or falsework, but their reuse is limited to structures with similar geometry ([29,30]-chapter 2.3.2). To save material and labor costs, Mexican builders developed a specific type of vault geometry to make it buildable with brick and mortar without formwork [31]. To facilitate the economical production of shell structures, overhang methods were developed in the former Soviet Union, which uses pre-cast free cantilever modules that are supported entirely by previously installed units [32]. Notable examples include the cupola roof with a diameter of 75 m for the Sport Palace in Tbilisi, Georgia, which uses an interlocking mechanism and bolts to allow the cantilevering [33],¹ and the square market roof with a span of 42 m in Belya Zerkov, Ukraine, which welds the cast-in reinforcement bars of neighboring modules [34].

Several algorithmic methods are proposed to aid in the planning of construction sequences for vaulted and shell structures. Beyeler et al. [35] propose a backward search algorithm to find stable deconstruction sequences of masonry structures. Decomposition algorithms have been proposed to first divide the structure into element groups and then use these groups as partial ordering to accelerate the search for the construction sequence in cable-assisted masonry construction [21].

Recently, interest in scaffold-free construction of vaulted structures has been revived by the development of robotic assembly technology. Robotic collaboration strategies, which usually involve one or more robots holding the partial construction for stabilization while other robots add new elements, have been proposed to assemble vaulted structures [36–39]. In [1], three cooperative robotic assembly strategies are compared for scaffold-free construction of a stable masonry arch, focusing on modeling and analyzing mechanical behavior during construction.

Trusses and frames. Historical studies on scaffold-free construction of trusses or frame structures are much sparser than those on vaulted or shell structures. Schukhov's hyperbolic lattice towers built with timber elements are designed to be assembled without scaffolding, due to a specially designed structural system and assembly sequence to ensure sufficient workspace for workers and structural stability during construction [40].

In recent work on algorithmic sequencing of bar structure construction, a graph decomposition algorithm is proposed to divide a frame structure into stiff layers to accelerate the search for sequences for robotic spatial extrusion, constrained by FEA-based stiffness analysis [41]. Hayashi et al. [42] combine graph embedding and reinforcement learning to train a sequence planning agent that can efficiently generate a stable assembly path for arbitrary 3D trusses. Their work focuses on the assembly of trusses and minimizes the total number of temporary supports throughout the process. Wang et al. [43] propose a topology optimization formulation with a new temporal coherency constraint to compute the optimal assembly sequence of frame structures. While the objective function is limited to compliance, the proposed algorithm can find a high-quality (sometimes optimal) sequence quickly and can be extended to handle multiple-bar-at-a-time assembly and assembly with static or dynamic scaffolding. In contrast, this work focuses on search-based methods for its flexibility in incorporating any objective and constraint, and investigates how the planned results impact design decisions.

Recent work on multi-robot assembly has also shed new light on scaffold-free construction of trusses and frames that require precise positioning of elements and stable support during construction. Collaborative robot assembly strategies were developed for timber [44] and steel [45] frames. Bruun et al. [3] uses graph rigidity theory to identify stable assembly and disassembly sequences for dual-robot

construction of trusses. In [46], a breadth-first search is used to operate on a support hierarchy graph to identify a stable disassembly sequence to reuse nominated elements from an existing timber stick frame.

Although current robots are not yet capable of construction skills that require dexterity, such as tying knots or making connections, research in collective human-robot construction aims to combine the in-situ decision-making and dexterity of humans with the precision and strength of robots [47]. Existing work aims to transcend the traditional linear workflow of pre-planned structures to allow humans to make design decisions on site, while robots can help stabilize and position elements [48,49].

2.4. Assembly-aware design

Prior work in assembly-aware design connects the design and the assembly sequence by searching assembly plans and varying parts geometry and/or layout. Significant amounts of work in computer graphics have used similar principles to design toys, furniture [50], and mechanical devices [51,52]. Interested readers are referred to [53] for a review of these efforts.

In architecture and construction, Parascho et al. [45] design a tetrahedra-based shape generation logic specifically for alternating build-and-hold task sequences for two robots. In [3], this design logic for the growth of tetrahedra cells is further enhanced by graph-based checks to ensure rigidity during construction. Zhang et al. [54] propose a template-based design synthesis method to generate corbel dwellings that only use limited types of interlocking blocks for construction without mortar and formwork. Motamedi et al. [55] develop a patching grammar to design the topology of multi-vaults for scaffold-free 3D printing of clay. Wang et al. [56] propose a topology optimization formulation that optimizes the structural density layout and the fabrication sequence for additive manufacturing. In [57], a disassembly search algorithm coupled with a collapse constraint is used to find a stable construction sequence for masonry shell structures, and this information is used to guide the design of the geometry and tessellation of the structure. Zargar et al. [58] demonstrates integrating a simplified robot simulation, which approximates the time of robotic pick and place of elements, into a multi-objective optimization, trading off constructability, structural, daylight, and energy goals for the design of a stand-alone classroom.

In this work, we propose a computational framework to assist in the decision-making about the design and planning of frame structures. Unlike historical studies that focus on a specific type of structural system with a specific building method [31,40], our framework is a planning, analysis, and evaluation tool for studying any frame structure's elastic structural behavior during construction.

3. Evaluating construction sequences

Given a bar structure design, our workflow allows the user to quantify the constructability of the design by searching for feasible or optimal assembly sequences under user-specified constraints and objectives. Designers could then use this score to compare different design options and make informed design decisions. The overview of the entire design workflow is shown in Fig. 1. The input bar structure is designed and sized in another process according to its complete state for nominated loads, which is not the focus of this work. Our proposed workflow focuses on the formulation and computation of constructability scores for this given design.

In this section, we first detail the assumptions about the structural system and the construction process considered (section 3.1). Then we discuss the structural constraints and objectives for evaluating a given construction sequence, which forms the foundation for the proposal of constructability scores in the next section (section 3.2).

¹ See

[https://taa.net.](https://taa.net.ge/en/archive-geo/tbilisi-sports-palace/)

ge/en/archive-geo/tbilisi-sports-palace/ for an Internet archive of this project.

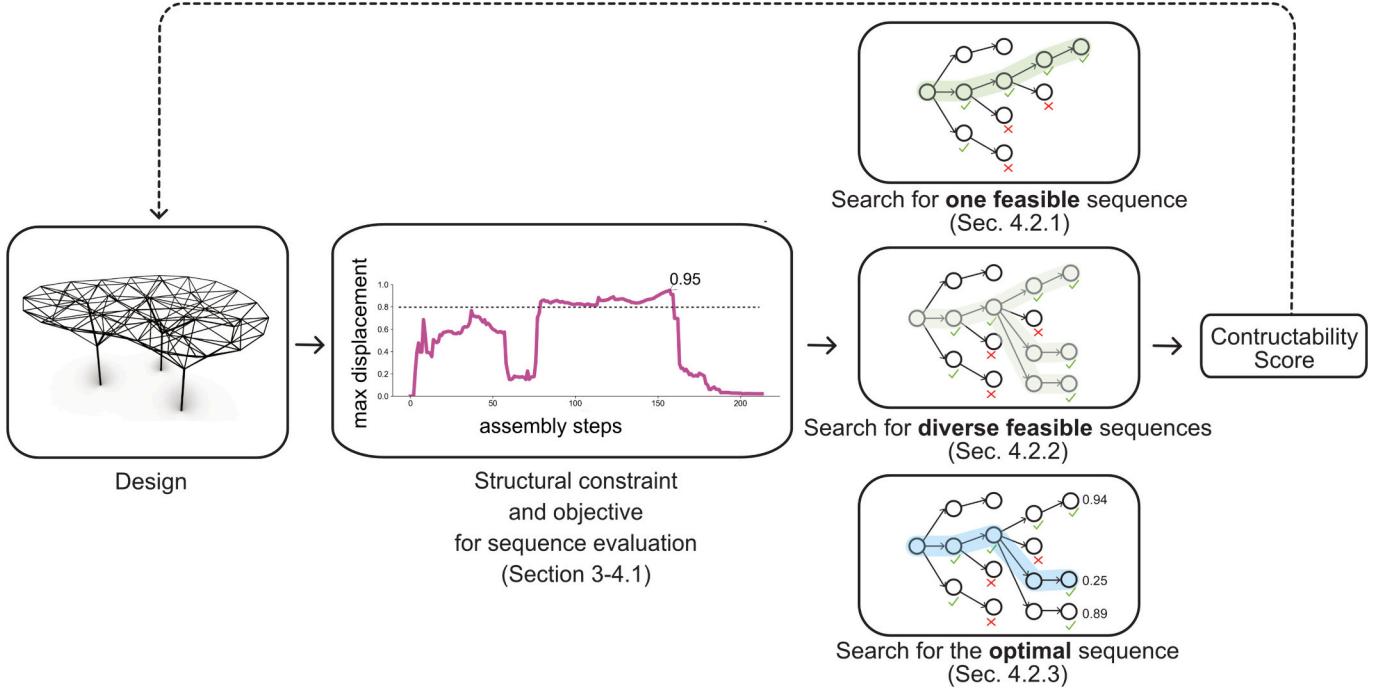


Fig. 1. Overview of the proposed constructability-driven design workflow.

3.1. Assumptions on the structural system and construction process

We focus on a broad class of building structures: bar structures, which are composed of discrete linear elements and used broadly and in great variety in the built environment. A bar system is a collection of n elements that form a set E , where each element corresponds to a bar and could have different material property and cross-section. The term bar system does not enforce a certain structural typology on the final structure, and thus when completed, the primary structural action could be dominated by axial forces, as in truss systems, or it could also include bending and torsion, as in frame systems.

In this work, we focus on scaffold-free construction, which means that only the elements in the final design can be introduced, and no scaffolding is allowed during assembly. Unlike previous work on scaffold-free assembly based on rigidity or equilibrium [3], we focus on stiffness-based assembly that allows the partially constructed structure to cantilever, and we aim to control the maximum displacement or stress during construction. We assume that all elements are connected by rigid connections, which can transfer moment and torsional load. In the case where the structure is designed as a truss in its completed state, rigid joints make the structure statically indeterminate, and internal forces will follow the stiffer possible load path, and thus its primary structural action will still be axial force as its truss counterpart with pin joints ([59]-chapter 1). The structure can be designed and sized according to specific nominated loads in its completed stage, but we focus on forces induced by self-weights during the construction.

We define a construction sequence as an ordering of elements in which they are added to the structure. We assume that elements are not allowed to be removed and later re-introduced once they are assembled in place. We call such a construction process a *monotonic, scaffolding-free* process because the structure strictly grows over time. Viewed from the context of the generation of mechanical assembly sequences [12], we consider a new type of constraint, the stiffness constraint, to control deformation, along with the traditional liaison predicate to ensure part connectivity.

3.2. Structural constraints and objectives on construction sequences

This work considers structural constraints related to displacement or stress to check whether a certain sequence is feasible. These structural considerations can be quantified by bounding a physical number obtained from a physics simulation at each construction step. Mathematically, the constraint evaluation function g that operates on a construction sequence can be expressed by the following inequality:

$$g(\psi[1:k]) \leq t, \quad \forall k \in [1, L, n], \quad (1)$$

where g performs the structural analysis on a partial construction and outputs the considered physical number (e.g., the maximum displacement or stress), and t is the given bounding tolerance. $\psi[1:k]$ is the partial construction specified by the first k steps of the nominated sequence ψ . An objective function f defined in a construction sequence can be expressed as a function that maps any sequence to a real number.

Solving the sequence planning problem has two modes: (1) feasibility problem and (2) optimization problem. Although any user-defined function can be used for g and f , we focus on two important metrics of mechanical behavior: displacement and stress.

Displacement. The max-displacement constraint enforces that a partially constructed structure's maximal nodal displacement is below a given tolerance:

$$g(\psi[1:k]) = \max_{i \in \{1, \dots, k\}} \text{displacement}(\psi[i]) \leq t, \quad \forall k \in [1, L, n] \quad (2)$$

Each bar element experiences a self-weight load due to gravity, which causes the structure to bend downward. Excess displacement is undesirable because it leads to alignment issues or material failure in the construction process. In this work, the displacement of all nodes is calculated using a first-order linear elastic finite element analysis (FEA) of a 2D or 3D frame structure. The Euclidean norm of the x, y (and z) translational displacements at each node is compared to the given maximum allowed tolerance.

To illustrate the difference in displacement behavior among different construction sequences, Fig. 2 shows two feasible construction sequences for a 2D bridge design. The 2D steel bridge is designed and sized in a separate process to support the gravity load and the in-service point

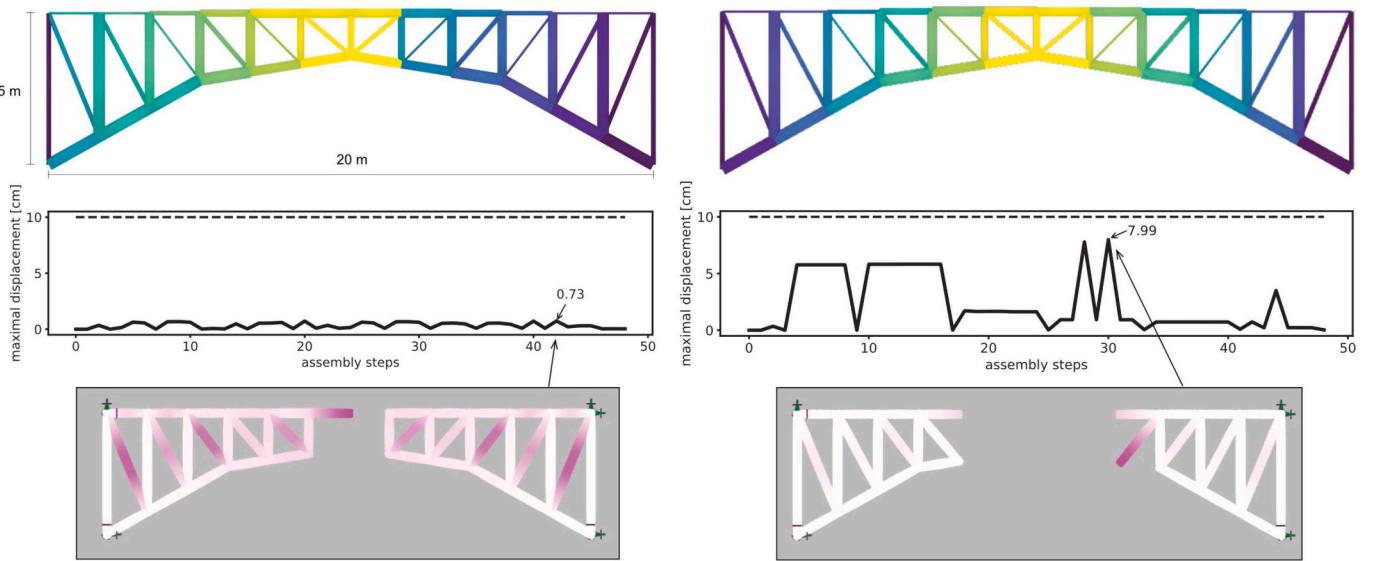


Fig. 2. Two construction profiles for a 2D bridge design under the displacement constraint. The construction sequence is color-coded, with purple built first and yellow built last. The dashed line shows the maximum allowed displacement tolerance (10 cm). The bottom images show the construction steps that have the maximum displacement, color-coded by the displacement value from white to pink. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

loads of 10 kN per node approximating vehicle live load, applied at the upper chord of the structure. In the design phase, all joints are modeled as rigid and the structure is rigidly supported at the four points on its horizontal boundary. Solid circular cross sections are used for all elements with standard structural steel (specific weight 78.5 kN/m^3), with optimized diameters ranging from 1 cm to 5 cm. The maximal displacement is set at 10 cm (shown as a dashed line in Fig. 2). In the maximum displacement constraint formulation, a feasible construction sequence must have the maximum displacement at each step bounded below this number. We can further plot the maximal nodal displacement against the construction steps. This plot, called *construction profile*, shows us the evolution of the physical attribute that we care about throughout construction and is an essential tool for us to quantify the performance of the construction. In practice, such construction profiles have been used to examine and predict the bridge construction process with nonstandard scaffolding systems [60]. Fig. 2 shows that the same design, with the same displacement tolerance, can have feasible construction sequences with very different construction profiles. We need to formulate a metric, that is, defining an objective function f , to compare these sequences and find an optimal sequence for a design with respect to this metric.

Depending on one's requirements, there are many formulations for evaluating a construction sequence. The simplest of all is optimizing for the worst step during the construction, i.e. using the maximum value of the graph to represent the corresponding sequence. Mathematically, this corresponds to having the objective function equal to the constraint function: $f(\psi) = g(\psi[1:n])$. Examples of worst-case partial constructions are shown in the bottom row of Fig. 2. There are other formulations for evaluating a sequence based on its construction profile, for example, a Dirichlet energy that measures the smoothness of displacement evolution by $\sum_i |\nabla f|^2$ or the total accumulated max displacement $\sum f$. For simplicity, we focus on the maximum-displacement formulation in the experiments in this work.

Stress. Load-induced stress can be another physical aspect that one might be concerned about during construction. Excessive stress can lead to material failure that is catastrophic for the construction process. Thus, bounding the maximum stress throughout construction is critical to the safety and success of the construction process. Similarly to the displacement constraint, the maximum stress can be calculated through

FEA by taking the maximum Von-Mises stress of each element across its length. Fig. 3 shows two feasible construction sequences under the stress constraint for the same 2D bridge design shown in the last section, with a stress tolerance set to 100 MPa, factored down from the yield stress of structural steel (250 MPa). We can apply all the sequence objective formulations that we discussed above for the displacement constraint to the stress constraint.

4. Constructability score

4.1. Formulating the constructability score

Based on the evaluation framework for a given construction sequence outlined in the last section, our aim is to derive a definition of the constructability score for a design. We propose two types of constructability scores, one focused on feasibility and the other focused on optimality.

The feasible constructability score. The feasibility score is proposed to answer the first question we asked in section 1: “Is this design buildable?”. Therefore, a feasible constructability score F_{feas} is simply a binary value: 0 when a feasible sequence exists and 1 when none exists.

The optimal constructability score. To answer the other questions we raised in section 1: “What is the best way to build a design?” and “Are there other design options that are easier to build?”, we define an optimality score based on the best-performing construction sequence concerning any user-defined objective we discussed in section 3.2. For example, if we use the maximum displacement as the objective function, the optimal score is the minimum of the maximum displacement of all feasible sequences. Later in section 4.2.3, we will develop an algorithm to find the optimal sequence for this objective. Mathematically, the optimal score is defined as:

$$F_{\text{opt}}(X) = f(\psi_*) = f\left(\underset{\psi}{\operatorname{argmin}} f(\psi)\right)$$

where F_{opt} is the optimal constructability score function defined on a design X and ψ_* is the optimal sequence that minimizes objective function f .

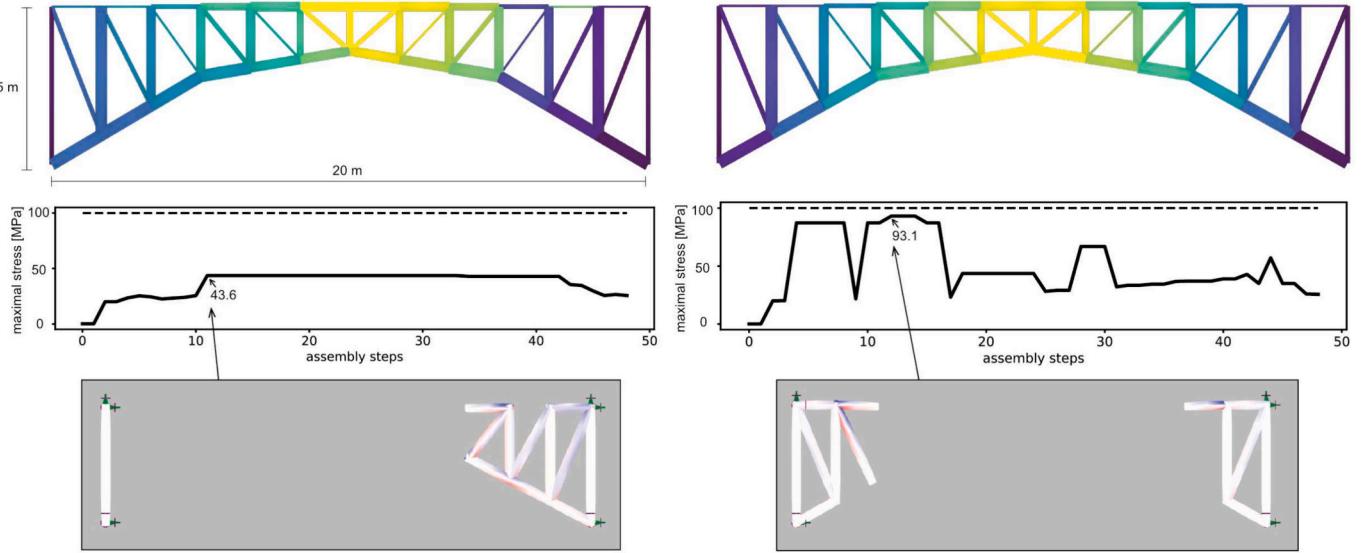


Fig. 3. Two construction sequences and their profiles for a 2D bridge under the stress constraint. The bottom row shows the construction steps that have the maximum stress, color-coded by stress (compression in red, tension in blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.2. Computing the constructability score

To compute the sequences needed to evaluate the constructability score, we adapt the standard greedy search algorithm with backtracking ([61]-chapter 6.3) to find the construction sequences. In the context of this work, a state includes the partially constructed structure, and the transition action is simply adding an element. Thus, the set of possible states is finite but can be large, containing permutations of all the elements.

In this section, we describe three new variants of the greedy best-first search algorithms to find (1) a single feasible sequence (section 4.2.1), (2) a diverse set of feasible sequences (section 4.2.2), (3) the optimal sequence (section 4.2.3).

4.2.1. Deterministic search for finding a feasible sequence

If the goal is to determine whether there exists a feasible sequence for a design, the planning algorithm should terminate as soon as it finds a feasible solution without wasting time on further exploration. The algorithm should also be algorithmically complete, meaning that it will find a solution if one exists.

We adapt greedy backtracking search ([61]-chapter 6.3) to solve the constraint satisfaction problem of searching for a feasible sequence. A flowchart of this algorithm can be found in Fig. 4. The search algorithm progresses from the initial state $P = \emptyset$ to the goal state $P = E$. An open queue is used to store candidate search nodes, which contains the search node that represents the partial construction P and the next element e to be considered. The search nodes in the queue are sorted by a priority function $k(P, e)$ defined on the previous state P and the next element e . At each iteration, the search node P in the open queue that minimizes $k(P, e)$ is expanded to form the new state $P' = P \cup \{e\}$ (Fig. 4-(1)). Then, it is checked with the constraint function $g(P') < t$ (Fig. 4-(2)). If the constraint is satisfied, the algorithm could terminate and return the complete sequence ψ , or continue to expand the search node P' by adding all unbuilt elements e to the search queue, where e is topologically connected to P or supported (Fig. 4-(3)).

We propose a priority function $k(P, e) = \langle n - |P|, h(e) \rangle$ that first orders search nodes by the number of unbuilt elements $n - |P|$, and then breaks the ties using a heuristic function $h(e)$ defined on individual element e . By prioritizing search nodes with fewer elements that remain to be planned, the search greedily explores the state space in a *depth-first*

manner. Because all successor states P' of P share the same number of remaining unbuilt elements, the heuristic tiebreaker decides the order in which successors are considered. This local ordering can have strong global effects on the sequence of partial construction considered. Many heuristic formulations can be used, e.g., based on randomness, Euclidean, and graph distance, and we refer the readers to [17] for an in-depth treatment of the impact of the heuristic choice on the search algorithm's performance on trading off satisfying the stiffness constraint, same as the structural constraint discussed in this work, and the collision constraint to ensure robot reachability.

In this work, we adopt the EuclideanDist heuristic for $h(e)$, i.e., ordering the elements by the z-coordinate of the midpoint of each element. This means that elements that are closer to the ground are prioritized, which is a widely used, intuitive heuristic used in many construction practices.

To save computational overhead from duplicated FEA simulations in the same partial construction P , we maintain a global cache of the performance score of P for reuse whenever possible.

4.2.2. Diverse search for finding a feasible sequence set

Extending the search algorithm for finding one feasible sequence, we propose an effective technique for finding feasible sequences that vary substantially from each other, which is also a valuable use case in design and construction. If a design has many good possible sequences, as identified through this method, this may indicate that it is robust against unmodeled construction challenges and flexible in accommodating such inputs from human experts later in the process.

In general, diverse search techniques usually involve using previously found solutions to bias or deflate the search away from existing solutions (e.g., the deflation technique in nonconvex optimization [62]). Simply letting the search continue exploring the remaining search queue after finding the first feasible solution will cause the search to converge to similar solutions, because it explores in a depth-first manner guided by the priority function $\langle n - |P|, h(e) \rangle$. After finding the first solution $\psi_1 = [e_1, L, e_n]$, the search will continue from the search node that corresponds to the partial construction state $P = \{e_1, L, e_{n-1}\}$ by taking a step back from ψ_1 . Then, the restarted search will converge to the same solution ψ_1 , and restart from the state $P = \{e_1, L, e_{n-2}\}$. This means that the search will spend most of its time exploring local permutations of the last few elements of the solution ψ_1 and will converge to local minima around ψ_1 .

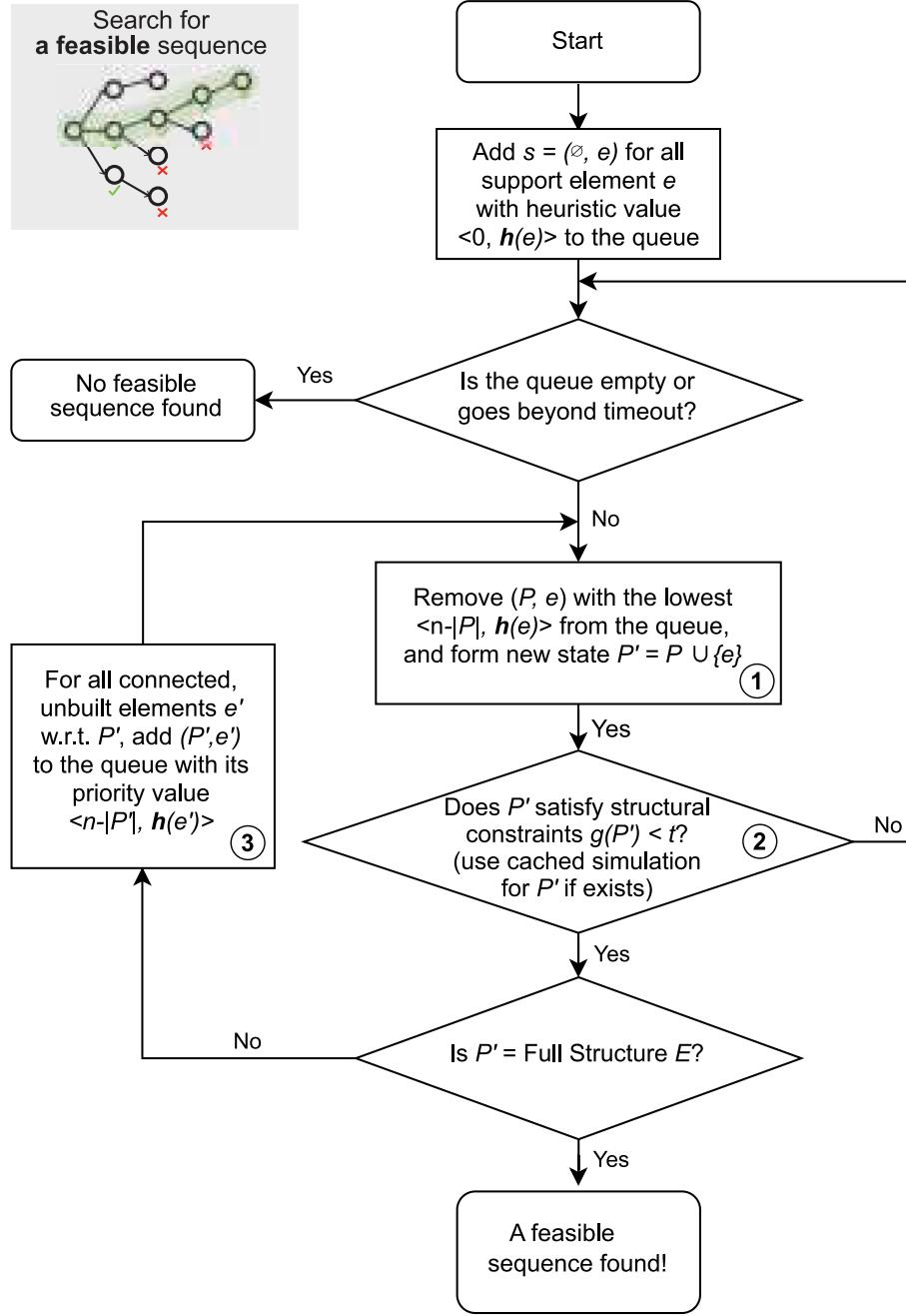


Fig. 4. Algorithm flowchart of the adapted greedy backtracking search algorithm for finding a feasible sequence.

To find a more diverse set of solutions, we modify the feasible search in the last section so that it (1) restarts the search by clearing the queue whenever a new solution is found and (2) uses a new, diversity-driven heuristic function. The flowchart of this diverse search algorithm is provided in Fig. 5, with the added steps highlighted in gray.

The diversity-driven heuristic function can be written as:

$$h(e, P, \Psi) = - \sum_{\psi \in \Psi} \sum_{e_k \in P \cup \{e\}} |I_\psi(e_k) - I_{\psi_p}(e_k)|, \quad (3)$$

where Ψ is the stored set of construction sequences found. P is the partial construction of the current search node and e is the candidate element considered. $I_\psi : e \rightarrow N$ is a mapping function that maps an element to its index in the sequence ψ . ψ_p is the sequence that leads to the partial construction P . Intuitively, this heuristic function computes the weighted average of the accumulated permutation difference between

the uncompleted sequence ψ_p and each found sequence $\psi \in \Psi$. This diversity score penalizes the new sequence ψ_p from having too many local permutations compared to the sequences found, in which case $h(\cdot)$ will be less negative and less prioritized. Thus, it encourages the search to prioritize search nodes with more negative values that potentially lead to more diverse solutions. Note that changing h in the priority function $k(P, e) = \langle n - |P|, h(\cdot) \rangle$ only affects the order in which the search explores the successors of a search node, and the search still explores in a depth-first manner, which maintains the completeness of the search. To maintain the speed of finding the first feasible sequence, we use the EuclideanDist heuristic for the first solution and switch to the diversity-driven heuristic function after the first solution is found (Fig. 5-(1)). Every time a new sequence is added to the solution pool Ψ , the diversity heuristic calculation will be updated due to the change in Ψ , and the search is restarted by clearing the open queue (Fig. 5-(2)).

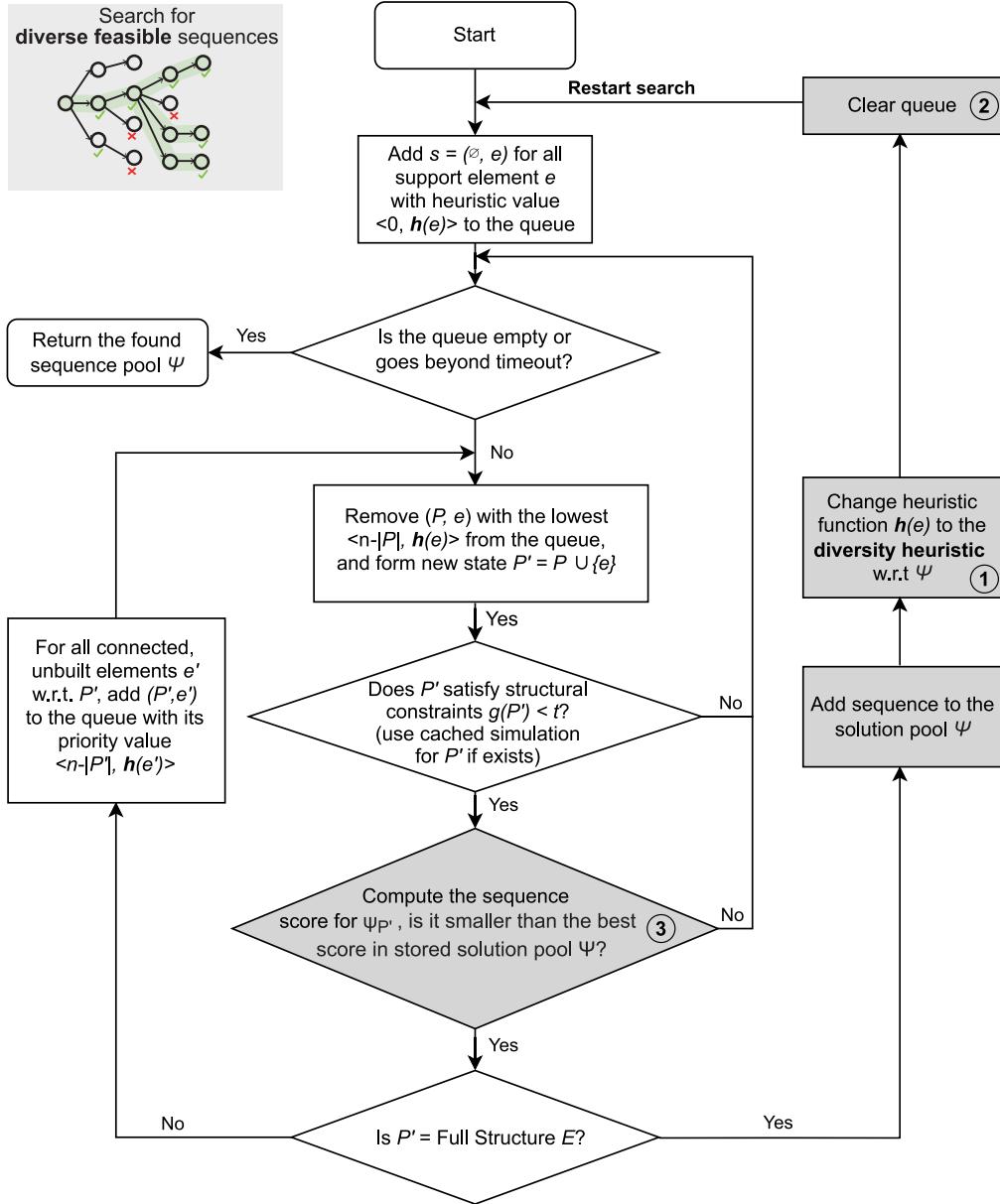


Fig. 5. Algorithm flowchart of the search algorithm for finding a diverse set of feasible sequences. Components that are different from the feasible search algorithm (Fig. 4) are highlighted in gray.

In addition, we are interested in identifying sequences that perform better than the ones found so far, so we apply branch-and-bound pruning if the current search node's partial structure P' has a higher objective value (worse) than the best solution found in the solution pool Ψ (Fig. 5-(3)). If so, the search node is discarded and we consider the next candidate search node on the open queue.

The diversity heuristic formulation in eq. (3) could be extended to other diversity metrics. In its current form, the absolute value in the second summation in eq. (3) computes the difference between the ordered position of e_k in the sequence ψ_P and its ordered position in ψ , for each element e_k in the partial construction $P \cup \{e\}$. This formulation for computing a distance measure between two permutations is called position distance [63]. Many other distance measures are used in the stochastic local search literature to compare two solutions [64,65]. Experimenting with other permutation distances is left for future work.

4.2.3. Optimal search with iterative constrained search

In the general case, the sequence optimization problem with arbitrary objective and constraints can be written as:

$$\begin{aligned} \min_{\psi} \quad & f(\psi) \\ \text{s.t.} \quad & g(\psi[1:k]) \leq t, \quad \forall k \in [1, Ln] \end{aligned} \quad (4)$$

where t is a fixed constraint tolerance, f and g are nominated structural objective and constraint functions, as discussed in section 3.2. Solving this combinatorial problem in its entirety is challenging and induces a large computational overhead. We instead focus on the following, more restricted case:

$$\min_{\psi} \quad g(\psi[1:n]) \quad (5)$$

This is equivalent to enforcing the constraint and the objective to be the same function.

We propose an iterative approach by performing a feasible search with gradually tightened constraint tolerance t_k . A flowchart of this algorithm is given in Fig. 6. For each iteration k , we perform a feasible search (same as in section 4.2.1) with a tolerance t_k for the structural

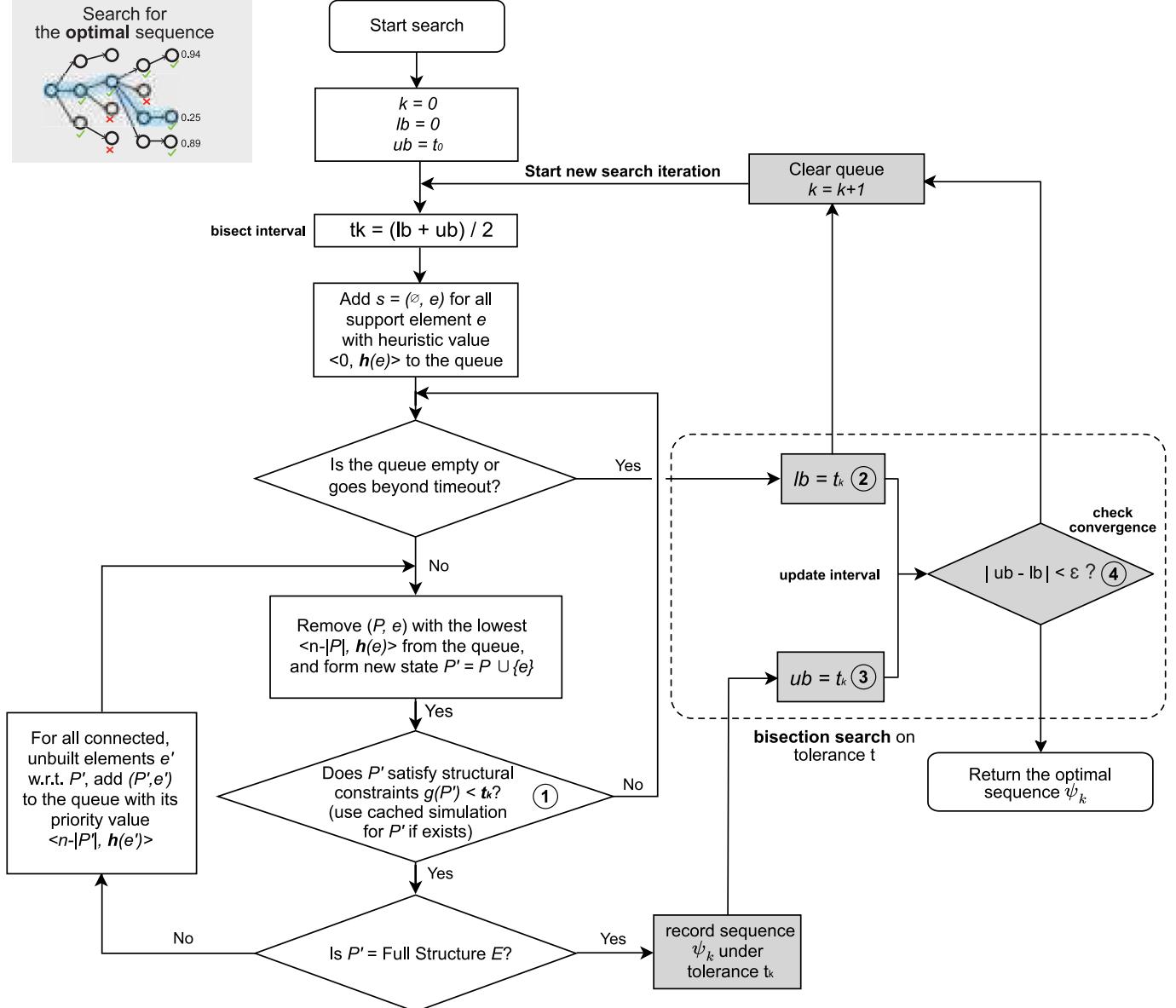


Fig. 6. Algorithm flowchart of the search algorithm for finding the optimal sequence with the smallest tolerance value bounding the constraint. Components that are different from the feasible search algorithm (Fig. 4) are highlighted in gray.

constraint $g(P) \leq t_k$ (Fig. 6-(1)). Across iterations, we use a bisection search on an interval $[lb, ub]$ to tighten the constraint tolerance t_k by a factor of 2: if no sequence is found within a timeout, we set the lower bound to the current tolerance t_k (Fig. 6-(2)); if a sequence is found, we set the upper bound to t_k (Fig. 6-(3)).

A convergence tolerance δ is used to control the termination of the bisection search (Fig. 6-(4)). If the difference between the current lower bound lb and the upper bound ub is less than δ , the algorithm terminates and returns the sequence ψ_k found in the last iteration. This algorithm is guaranteed to find the optimal t in a finite number of iterations under convergence tolerance δ , assuming the timeout for the inner-loop feasible sequence search is large enough to maintain its completeness.

5. Results

We demonstrate our proposed strategies in a parametrically designed small-scale double-layer roof structure (Fig. 7). The topology of the truss structure is generated by connecting nodes randomly populated from three layers of regions enclosed by user-prescribed 2D curves. We

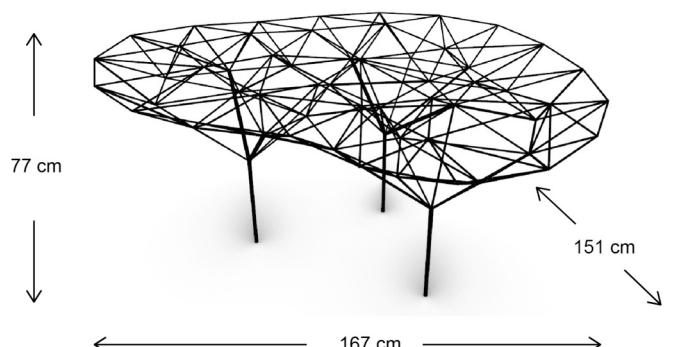


Fig. 7. Overview of the double-layer roof design.

parameterize the design by controlling the number of nodes sampled in each layer. The elements within each layer are connected by a Delaunay triangulation, and elements between layers are connected by a 3-

nearest-neighbor with a distance threshold. This design is inspired by dentriform or tree-like spanning structures, which have a long history in natural morphology [66], architecture [67,68] and light-weight structures [69]. The roof support structure in Stuttgart Airport, designed by GMP Architekten, is a well-known built example of this type of structure.²

This roof structure is scaled down to the size of a dining table or desk and consists of 214 elements and 62 nodes, with a physical dimension of 167 cm × 151 cm × 77 cm (length, width, height). Based on estimation of the weight of a table deck and live load, the design loads considered for the completed structure are distributed self-weight load and point loads of 4 N per node, applied at the 33 nodes on the top. The support condition is fixed at the bottom three nodes. The structure is designed and chosen because of its interesting geometric and structural configuration, which makes scaffold-free construction challenging. The structure is stiff in its completed stage because all parts of the system work together: the tension ring, the double-layer webs, and very few support columns. When built without scaffolds, because those parts are not there at once, the structure is very flexible and prone to large deformation during construction. Although the structural service mechanical behavior in its completed stage can be predicted using standard FEA tools, the in-construction behavior is much harder to estimate without the physically constrained automated planning techniques proposed in this work. The design is complex enough that it is hard for a designer to have an intuition on whether this design can be built without scaffolding or not. Thus, it provides an interesting test case for the sequencing algorithms proposed in this work.

Hollow, cellulose tubes with a circular cross-section and a wall thickness of 0.16 cm are used for all elements. The material properties of cellulose are presented in Table 1. The diameters of the cross section are optimized for these designed loads using the Karamba3D's cross section optimizer [70], with optimized diameters ranging from 0.6 cm to 1.6 cm. After sizing, the completed structure is expected to have a total mass of 2.16 kg and a service displacement of 0.28 cm. Detailed design data, including nodal positions and connectivity, are included as supplementary data for future benchmarking purposes.

In this section, we demonstrate two use cases: (1) for a given design, comparing construction sequences found by various configurations of the search algorithm (section 5.1) and (2) comparing design options using the optimal constructability score (section 5.2). Furthermore, we provide detailed data on search behavior for the diverse and optimal search algorithms in section 5.3. We focus on constraining and minimizing the maximum displacement in all experiments for the structural constraint and the objective. First-order linear elastic FEA is used for all experiments, with no buckling check performed. All the experiments presented in this section are performed using an IronPython implementation inside the GHPython environment in Grasshopper3D [71]. The structural analysis used to check the structural constraint and objective is based on Karamba3D [70]. All experiments were performed on a consumer-grade laptop without parallelization or GPU acceleration. The Grasshopper script can be found online,³ which includes the

design geometry, the structural sizing and analysis, and a GHPython implementation of the three search algorithms discussed in section 4.2. Although the Python script for the search is only executable in the IronPython environment within Rhino3D because of the dependency for the Karamba3D API for analysis, it could be adapted to use other FEA packages straightforwardly. Thus, we include them as separate files in case readers want to adapt them to their own use cases or the Grasshopper file becomes obsolete in the future.

5.1. Comparing construction sequences

In this section, we show three sequences found by the sequence search algorithm under various configurations, with a target maximum displacement of 0.8 cm. This rather generous displacement tolerance is set to be the radius of the largest cross-section used in the design, to ease the alignment of elements during construction. However, we note that in practice, a more stringent displacement tolerance could be used according to the design code, for example, 0.3 cm according to the commonly used rule of thumb $L/180$ for the longest cantilever distance 47 cm. Key search statistics are summarized in Table 2.

An infeasible sequence. As an example, we consider an infeasible sequence found by a forward state-space search, without checking the structural constraint (Fig. 8). At first glance, the sequence obtained seems like a viable solution, since it is biased to build the structure layer by layer from the bottom to the top, due to the EuclideanDist heuristic used (section 4.2.1). However, the displacement construction profile reveals that the sequence is not feasible and has about 40 steps in the sequence with maximum displacements above the specified tolerance. Thus, this proves that without explicitly incorporating the constraint into the search, the sequence found by a naive search algorithm can be infeasible, even using a seemingly reasonable heuristic. Furthermore, the quantitative analysis of the construction sequences using the construction profile is important to reveal detailed structural behavior during construction.

A feasible sequence. Next, we look at a feasible sequence found by the same search algorithm as before, with the only difference being that the structural constraint checking with a displacement tolerance of 0.6 cm is included. The displacement profile illustrated in Fig. 9 shows that the maximum displacement is successfully suppressed under the given tolerance, although with the compromise of increased maximum stress during construction. This shows that incorporating the constraint into the search, as expected, will help find the feasible sequence under the prescribed tolerance. Now, it is natural to ask: What is the smallest value of displacement tolerance for a feasible sequence?

The optimal sequence. Finally, we look at the optimal construction sequence that minimizes the maximum displacement, computed by the optimal algorithm in section 4.2.3. Snapshots of this sequence are shown in Fig. 10. For demonstration purposes, we set the initial tolerance to be 1.7 cm, which is about twice the target tolerance of 0.8 cm, so that the initial guess is far away from the minimum and the algorithm has to perform some iterations to find the optimal sequence. A step-by-step illustration of the sequence is shown in Fig. A.13. The construction profiles suggest that the maximum displacement of the sequence is 75% below the infeasible sequence while maintaining the same maximum stress value. This result demonstrates the power of the proposed automated search technique, which is capable of finding the construction sequence that performs the best and is otherwise difficult to obtain. Detailed statistics of the optimal search algorithm for finding this sequence are discussed in section 5.3.

5.2. Comparing design options

We use the optimal search to compute the optimal constructability score to compare design options. Fig. 11 shows the three designs that we consider, including the original design as above and two design variations generated using the same parametric design model described in

Table 1
Material property of cellulose.

Young's modulus E (MPa)	In-plane shear modulus G12 (MPa)	Transverse shear modulus G3 (MPa)	weight (kN/m ³)	Tensile strength (MPa)	Compressive strength (MPa)
2070	828	828	12.7	43.4	43.4

² <https://www.gmp.de/en/projects/548/stuttgart-airport>

³ https://github.com/yijiangh/sequence_dse

Table 2

Statistics of three construction sequences found by three search configurations. The first two rows are run with the feasible search. The optimal search algorithm finds the optimal sequence with an initial tolerance of 1.7 cm. The complete search statistics are given in section 5.3.

Sequence	Score (cm)	Displacement tolerance (cm)	Timeout (s)	Num. state evaluation	Num. structural constraint failure	Runtime (s)
an infeasible sequence	infeasible	N/A	180	214	N/A	0.2
a feasible sequence	0.6	0.6	180	615	401	15.2
the optimal sequence	0.2	1.7	N/A	9354	8029	147.5

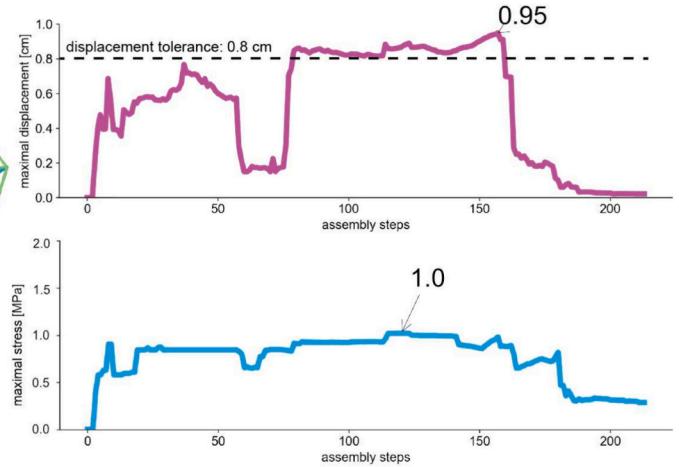
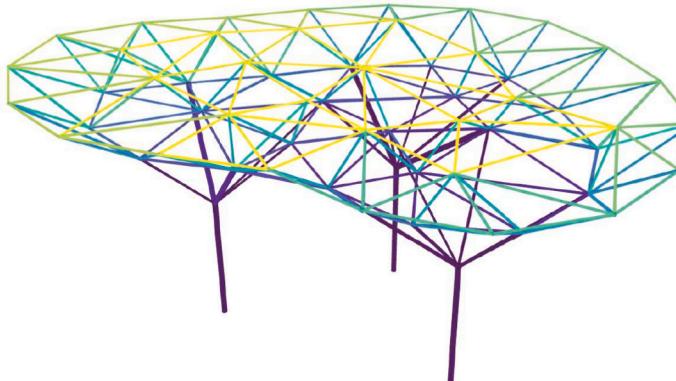


Fig. 8. Infeasible sequence found by the search algorithm without considering the structural constraint. The construction sequence is color-coded, with purple built first and yellow built last. The construction profiles on the right (top displacement, bottom stress) show that the maximum displacement of the sequence is above the given tolerance of 0.8 cm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

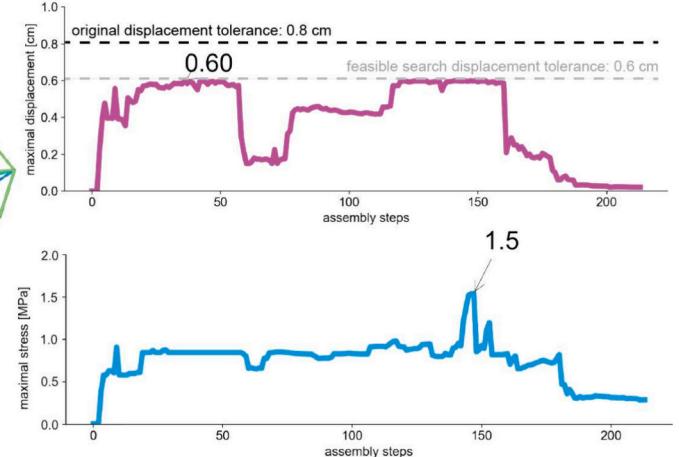
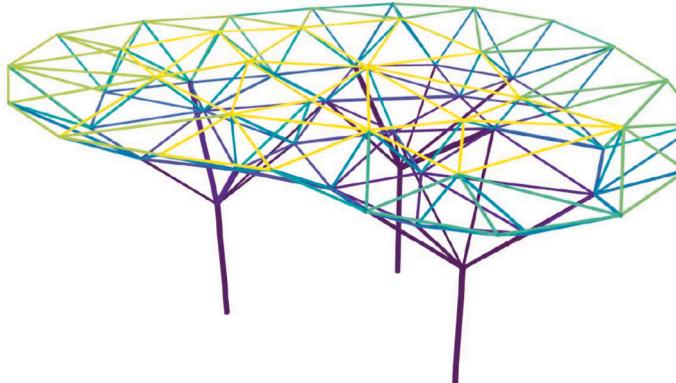


Fig. 9. Feasible sequence found by the search algorithm with a maximum displacement tolerance of 0.6 cm. The displacement profile (top right) shows that the sequence obtained remains in the given tolerance throughout the construction. The stress profile (bottom right) shows an increase in maximum stress, compared to the sequence shown in Fig. 8.

section 5. To diversify the design catalog, the connectivity, the number of support columns, and the number of points in the top chord are all varied by changing the number of sampled nodes in each layer. Table 3 shows the optimal search statistics for the three design instances. Illustrations of the two design variations' optimal sequences are provided in the appendix. Design variations 1 and 2 have more support columns supporting the roofs, 6 and 5 respectively, and are heavier than the original design. Design variation 1 dominates the other two regarding the optimal constructability score and has the lowest maximum stress value. However, this better constructability score comes at the cost of having more elements and consuming more materials, which is a trade-off that designers should consider.

5.3. Comparing the diverse and optimal search algorithms

Finally, we provide some search statistics to compare the optimal and diverse search algorithms by having them find sequences under the maximum displacement constraint on the original double-layer roof design (top row of Fig. 11). Although the two algorithms solve different types of problem, one for finding an optimal sequence and the other for finding a set of feasible sequences, this comparison provides us some additional algorithmic insights.

The optimal search performs 7 iterations to find the optimal sequence that minimizes the maximum displacement in 147.5 s. The search statistics for each iteration are given in Table 4. We observe that when the tolerance is approaching the optimum, the search time

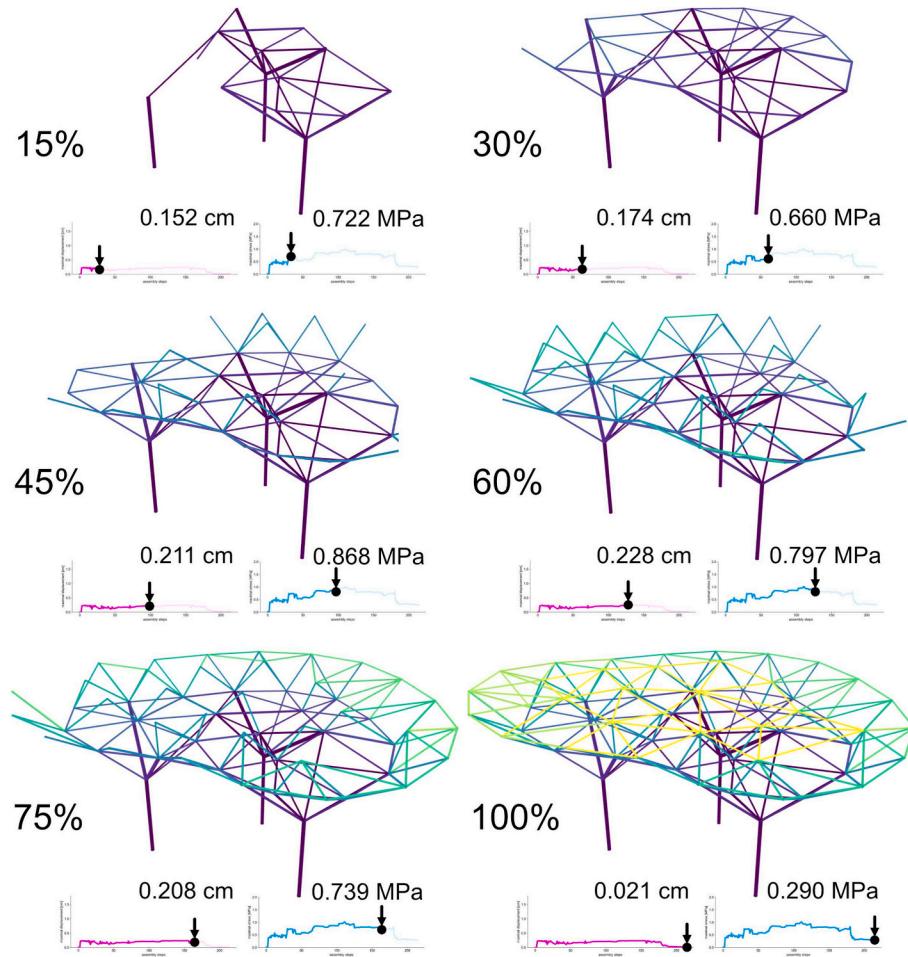


Fig. 10. Sequence that minimizes the maximum displacement found by the optimal search algorithm. A maximum displacement tolerance of 1.7 cm is used as the initial tolerance for the optimal algorithm. The construction profile shows a 70% reduction in the maximum displacement under the given tolerance. The sequence is illustrated step by step in Fig. A.13.

increases because the search encounters more constraint violations. Notably, iteration 7 has the most structural constraint failures while having a relatively short runtime. This happens because the search encounters many structural constraint failures that share a small set of partial constructions. In these scenarios, the FEA caching technique helps avoid redundant FEA computation. Although the optimal search only outputs the best-performing sequence in the last iteration, looking at the interim sequences helps us visualize the convergence process. Fig. 12 shows that the sequences obtained from the first, fourth, and sixth (optimal) iteration, with a score of 0.85 cm, 0.32 cm, and 0.24 cm, respectively. Due to the gradually tightened constraint tolerance, we can observe that the search avoids building long cantilevering elements early in the process and delay building them until neighboring, lighter elements are built and form a stiffer sub-structure locally. This is especially evident in the optimal sequence, where the element that causes the largest displacement in the previous iterations is delayed until the end (marked with a solid square on the right).

The diverse search algorithm has a one-hour budget to compute the best performing sequences and finds 20 feasible sequences. The search statistics are presented in Table 5. The histogram shows that the sequences found have similar performances, around 0.94 cm. The six best-performing sequences shown in Fig. 12 reveal the diversity of sequences found by the algorithm. These results show that the diverse search algorithm works well to find a diverse set of sequences, but performs poorly in optimizing the score.

As expected, Table 5 shows that the optimal search outperforms the

diverse search in the best quality of solution and total runtime. This resonates with a general observation in optimization: unconstrained optimization is usually easier than surveying the feasible subspace of a constrained problem.

6. Discussion

6.1. Summary of results

The proposed feasible search algorithm is demonstrated to automatically find a feasible construction sequence for the given double-layer roof design, by constraining maximum displacement to be below a given tolerance of 0.8 cm. In contrast, a naive search algorithm without considering the structural constraint is shown to find an infeasible sequence that has maximum displacement that goes beyond the tolerance.

The optimal search algorithm takes longer to converge, but it can find the optimal construction sequence that minimizes the maximum displacement, with a score of 0.2 cm. By analyzing the partial structures that have the largest displacement found by the search iterations, we show that the algorithm delays the construction of long cantilevering elements until neighboring, lighter elements are built, and thus minimizes the sequence score. We also show that the optimal search algorithm can be used to quantitatively compare design options by computing the optimal constructability score for each design. Finally, compared to the optimal algorithm, the diverse search algorithm takes

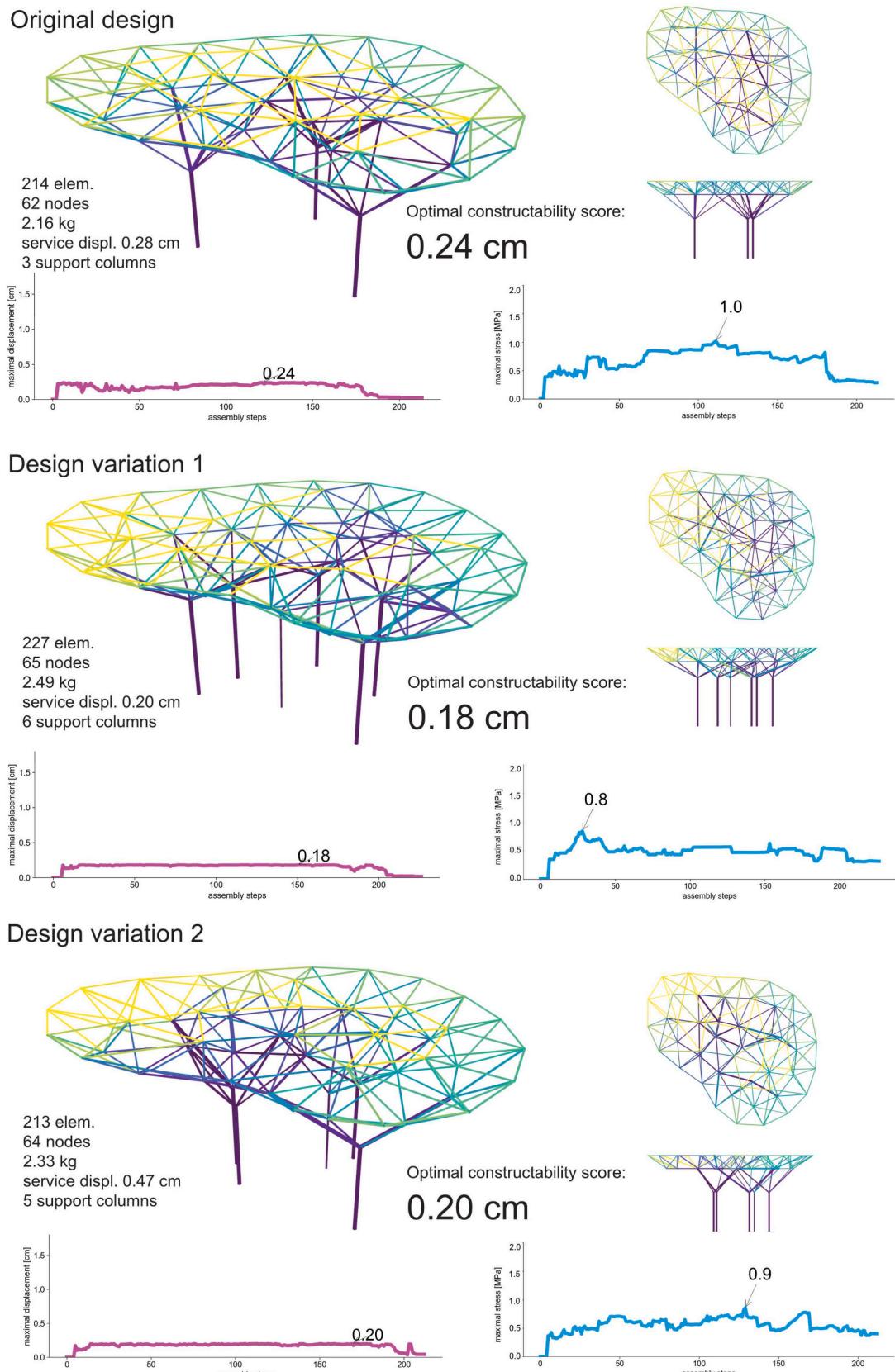


Fig. 11. Comparing three design options using the optimal constructability score.

Table 3

Search statistics of running the optimal search algorithm on the three design options. All three instances are run with an initial maximum displacement tolerance of 1.7 cm, the EuclideanDist heuristic, and a one-hour timeout.

Design	Iters	Num. of state evaluation	Num. of structural constraint failure	Num. of solutions found	Sequence score average	Sequence score std.	Sequence score minimum	Runtime (s)
Original design	7	9354	8029	5	0.42	0.22	0.24	147.50
Design variation 1	8	516,787	503,215	5	0.35	0.21	0.18	701.39
Design variation 2	7	817,633	727,922	4	0.42	0.26	0.20	561.10

Table 4

Detailed search statistics of the optimal search algorithm, with an initial displacement tolerance of 1.7 cm. At each iteration, the feasible search algorithm is given a 180-s timeout.

Iters	Displacement tolerance (cm)	Num. of state evaluations	Num. of structural constraint failures	Min. remaining elements	max. Num. of backtracking	Runtime (s)	Sequence score (cm)
1	0.85	283	69	0	0	8.7	0.84
2	0.43	588	374	0	0	11.6	0.43
3	0.21	1153	1092	208	6	3.5	infeasible
4	0.32	985	771	0	0	20.8	0.32
5	0.27	1331	1117	0	0	25.7	0.27
6	0.24	1211	997	0	0	23.4	0.24
7	0.23	3803	3609	207	7	5.6	infeasible

longer to compute and the resulting sequences have higher scores, but it is capable of finding a diverse set of feasible sequences.

Previously proposed assembly sequence planning methods focus mostly on optimizing other aspects of the assembly, such as assembly time, cost, and tool changes, and have limitations in scalability to large design instances with more than a few dozens of elements [10]. In contrast, our results show that the proposed search methods can solve for design instances with hundreds of elements, incorporate constraints and objectives based on elastic structural behavior that is rarely considered before, and be easily extended to find feasible, optimal, and diverse solutions. Moreover, albeit search-based methods are often labeled as a “brute-force” approach, we show that with proper state-space formulation, heuristic design, and constraint bounding schemes, the search can be efficient and used as a plug-and-play tool for evaluating the constructability of a given design.

6.2. Limitations and future work

For search-based assembly-driven design, we see a number of limitations and opportunities for future work. First, we have only demonstrated the evaluation of a discrete design catalog under the optimal constructability score, without involving a formal optimization algorithm in the outer design loop to minimize the score over a continuous design space. The proposed optimal search routine is still not fast enough to be used in a formal optimization loop. Future directions might include (1) improving the implementation and utilizing parallel computing to accelerate search node feasibility evaluation; (2) richer feedback from the planner to the outer loop optimizer to guide the optimization more efficiently.

Second, the optimal search proposed in section 4.2.3 only solves the unconstrained version (eq. (5)) of the sequence optimization problem. For general constrained optimization, as in eq. (4), a promising direction is to incorporate the constraints and objectives discussed in this work into an anytime beam search algorithm (e.g., [72,73]). However, challenges remain in converting the objective function into an admissible heuristic to maintain the optimality guarantee of the algorithm.

Third, while only structural constraints and objectives are discussed and experimented with in this work, the search algorithm is modularized and can be easily extended to include other constraints. While first-order linear elastic FEA is used for all experiments since it suffices as a

fast check for controlling excessive displacement and stress, if needed, it is straightforward to include more advanced structural analysis techniques, e.g., second-order analysis and buckling check. Other constraints can also be incorporated into the search framework, including, for example, operational-space constraints that ensure enough space for human or robot operators to assemble the element [74]. However, including these additional constraints exacerbates the computational overhead involved in the search. Furthermore, the completeness of the search algorithm could be broken due to the nature of subroutines used to plan or evaluate these constraints. For example, the sampling-based motion planning algorithm used to evaluate robot reachability could make the overall algorithm probabilistically complete instead of complete [17].

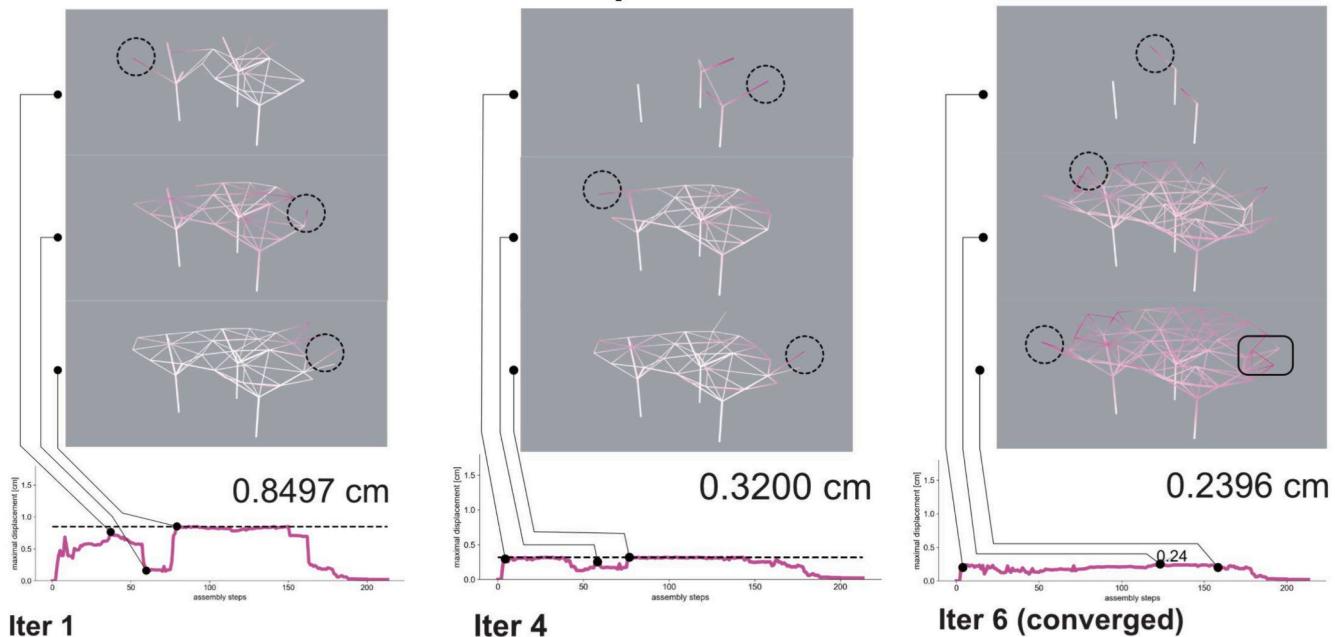
Finally, while scaffold-free construction eliminates any waste from the scaffolding system, such a “minimalist” construction process might not be practical for many structures. Introducing a few scaffolding elements [43] or using robots to support [3,45] can stabilize the partial construction and increase the robustness of the process. However, planning for an optimal construction sequence with minimal scaffolding elements makes the planning horizon infinite, and thus requires new algorithms to evaluate and compute such plans.

7. Conclusions

This paper has presented a way to characterize performance related to the construction sequence that can be used to inform design decisions. We focus on the context of the monotonic scaffolding-free assembly of discrete-element structures assuming rigid connections, and our goal is to control the displacement/stress behavior of the partially built structures during construction. In order to quantify constructability and compare many ways of building a given design in terms of construction sequences, we propose a constructability score formulation and state-space search algorithms to find high-performing sequences regarding the score. Specifically, our contributions include:

- Formulations to evaluate a construction sequence’s performance regarding its in-construction displacement and stress profiles.
- Score formulation to quantify the constructability of a given design by summarizing its sequence performance.

Sequences found in inner iterations of the optimal search



Sequences found by the diverse search

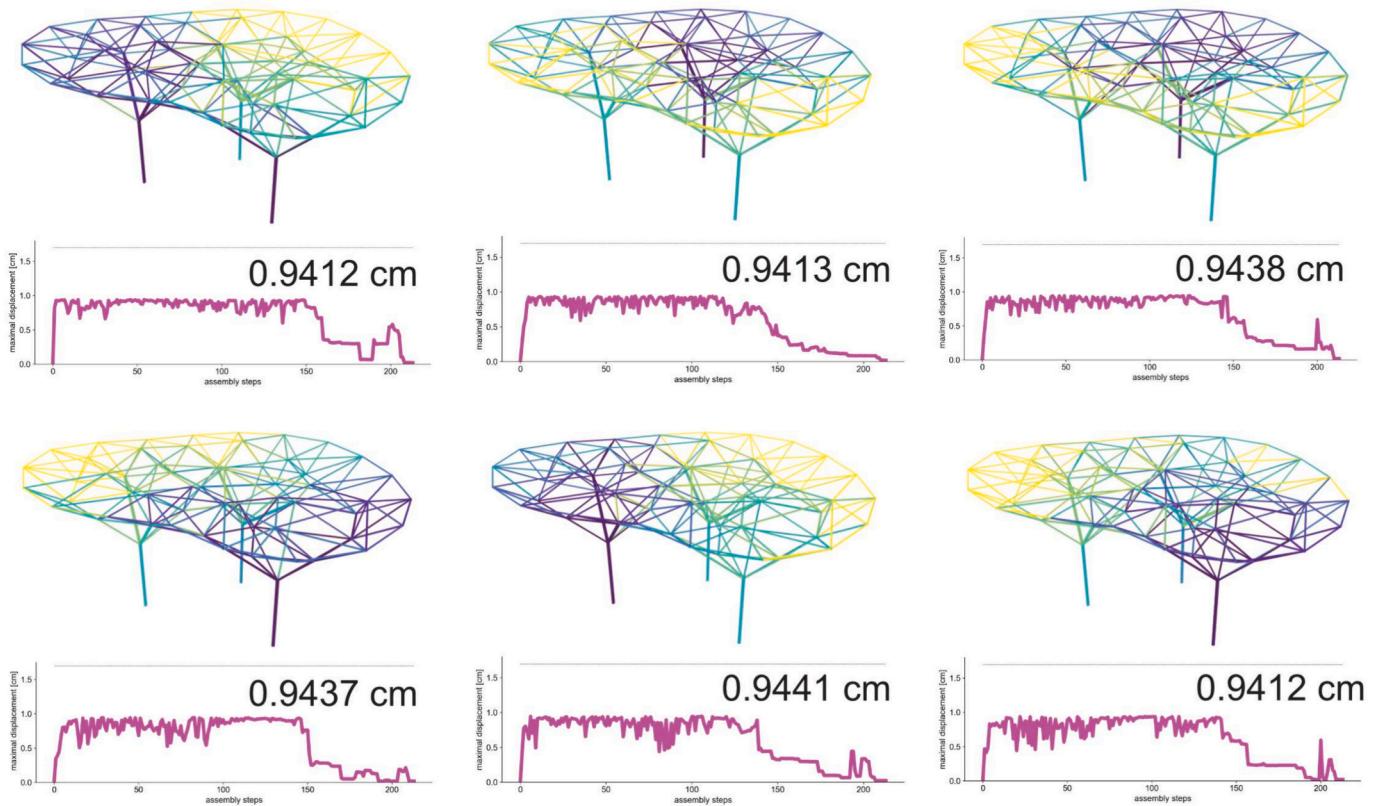


Fig. 12. Top row: sequences found in the inner iterations of the optimal search, with the final, converged sequence on the right. The elements that cause the largest displacement are marked with dashed circles. The square on the right highlights the element that causes the largest displacement in the previous iterations are delayed until a stiffer sub-structure is formed in the optimal sequence. Bottom two rows: top six sequences out of the 20 sequences found by the diverse search.

Table 5

Search statistics comparing the optimal and diverse search algorithms. Both algorithms run with a displacement tolerance of 1.7 cm, the EuclideanDist heuristic, and a one-hour timeout.

Search algorithm	Num. of state evaluation	Num. of structural constraint failure	Num. of solutions found	Sequence score average (cm)	Sequence score std. (cm)	Sequence score minimum (cm)	Runtime (s)
Optimal search	9354	8029	7	0.42	0.22	0.24	147.5
Diverse search	36,429	11,929	20	0.95	0.003	0.94	3600.0

- Three variants of heuristic search algorithms to find feasible, diverse, and optimal sequences regarding the constructability score, by combining heuristic design and constraint bounding techniques.
- Demonstrations on using search algorithms to evaluate the sequence score of a parametrized double-layer roof design and compare design options.

The results show that the evaluation framework and automated search techniques can help designers compare and explore the nuances of sequences and find the best-performing construction sequence for a given design. The constructability score can be used to compare design options and trade-off with other design considerations. We also show that the search scheme can find a feasible sequence for a design with more than 200 elements in a few seconds and can find the optimal sequence in less than 15 min.

The tools presented in this paper thus provides a formal and practical framework to help construction engineers to evaluate and plan scaffold-free assembly sequence. They also equip structural designers with tools to drive their design decisions with sequence-related constructability performance, in addition to other traditionally considered measures. The proposed search framework is flexible and invites further research to integrate other construction-related constraints, such as ensuring sufficient operating space, into the framework. The presented search results and the design evaluation study can serve as a baseline for future work to improve planning speed, as well as integration into a formal design optimization to find the optimal design with respect to the proposed constructability score.

As construction practices advance toward a tighter integration between design and execution, the ability to quantify certain aspects of construction and the use of algorithms to assist decision making becomes increasingly important. The methods presented in this paper represent a key step toward this future.

CRediT authorship contribution statement

Yijiang Huang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Conceptualization. **Caelan Garrett:** Writing – review & editing, Methodology, Formal analysis. **Caitlin Mueller:** Writing – review & editing, Visualization, Supervision, Resources, Methodology, Data curation, Conceptualization.

Declaration of competing interest

Yijiang Huang acknowledges the support of an ETH Zurich Postdoctoral Fellowship. Caelan Garrett reports a relationship with NVIDIA Corp that includes: employment and equity or stocks. There is no additional relationship, patent, and activities to report.

Yijiang Huang reports financial support from ETH Zurich and Massachusetts Institute of Technology. Caitlin Mueller reports financial support from Massachusetts Institute of Technology. Caelan Garrett reports financial support was provided by NVIDIA Corp. Caelan Garrett reports a relationship with NVIDIA Corp that includes: employment and equity or stocks. There is no additional relationship, patent, and activities to report.

Data availability

We have uploaded the design and benchmarking data in the attach file step.

Acknowledgement

The authors thank Suwan Kim for her contributions to the case study design presented in this article.

Appendix A. Optimal construction sequencing benchmarks

The node positions, element connectivity, cross-sections, and the optimal sequence (Fig. A.13) of the original design are provided as CSV files in the supplementary materials. In addition, Fig. A.13, Fig. A.14 and Fig. A.15 present detailed, step-by-step illustration for the optimal sequences computed for the original design, variation 1, and variation 2 in Section 5.2.

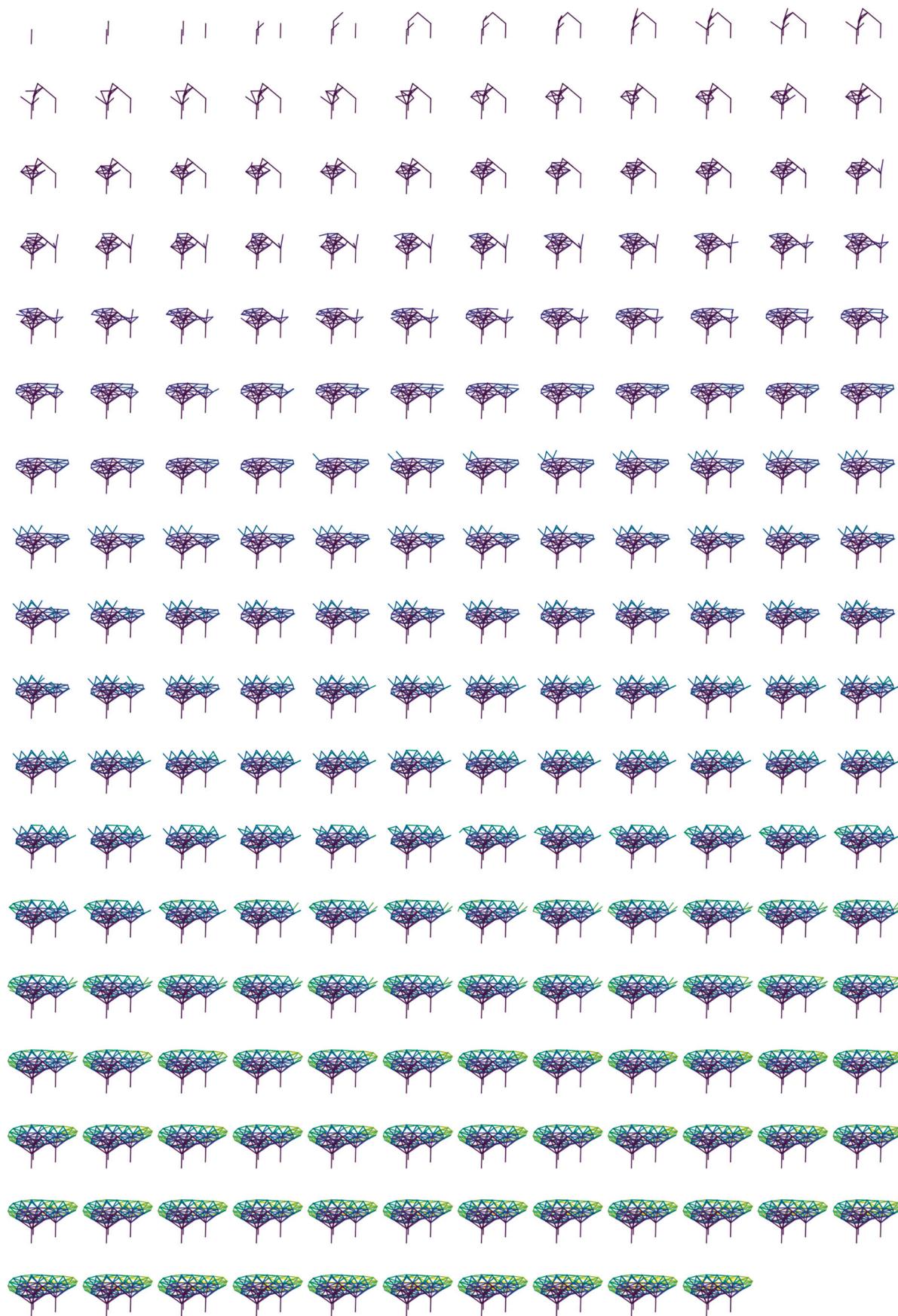


Fig. A.13. The optimal construction sequence for the original design (Fig. 10), found by the optimal search algorithm.

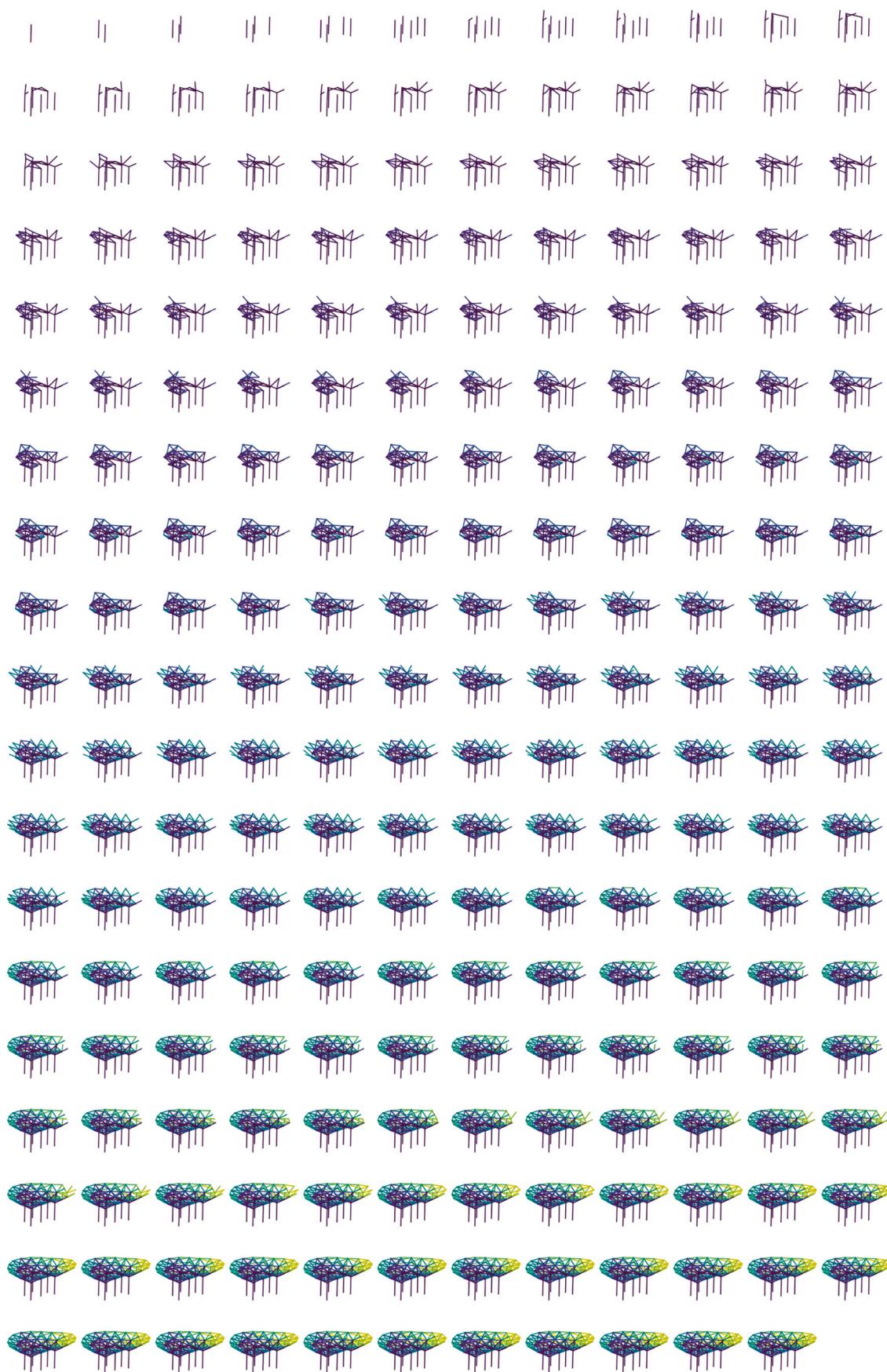


Fig. A.14. Optimal sequence for design variation 1 (see Fig. 11), found by the optimal search algorithm.

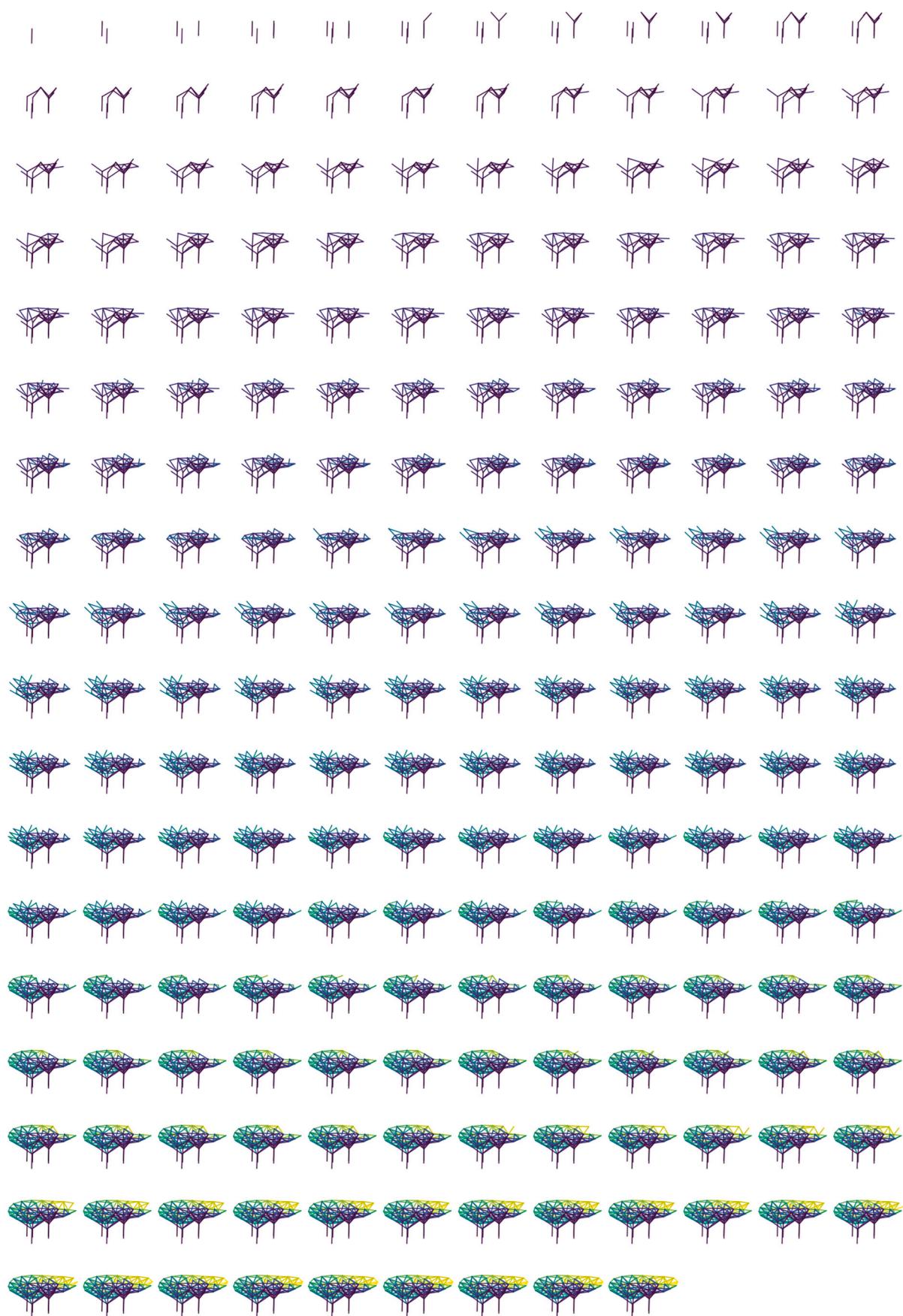


Fig. A.15. Optimal sequence for design variation 2 (see Fig. 11), found by the optimal search algorithm.

Appendix B. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.autcon.2024.105711>.

References

- [1] E.P. Bruun, R. Pastrana, V. Paris, A. Beghini, A. Pizzigoni, S. Parascho, S. Adriaenssens, Three cooperative robotic fabrication methods for the scaffold-free construction of a masonry arch, *Autom. Constr.* 129 (2021) 103803, <https://doi.org/10.1016/j.autcon.2021.103803>.
- [2] L. Huang, R. Pradhan, S. Dutta, Y. Cai, BIM4D-based scheduling for assembling and lifting in precast-enabled construction, *Autom. Constr.* 133 (2022) 103999. URL: <https://www.sciencedirect.com/science/article/pii/S0926580521004507>, <https://doi.org/10.1016/j.autcon.2021.103999>.
- [3] E.P. Bruun, S. Adriaenssens, S. Parascho, Structural rigidity theory applied to the scaffold-free (dis)assembly of space frames using cooperative robotics, *Autom. Constr.* 141 (2022) 104405, <https://doi.org/10.1016/j.autcon.2022.104405>.
- [4] C.T. Mueller, J.A. Ochsendorf, Combining structural performance and designer preferences in evolutionary design space exploration, *Autom. Constr.* 52 (2015) 70–82, <https://doi.org/10.1016/j.autcon.2015.02.011>.
- [5] V. Faghghi, A. Nejat, K.F. Reinschmidt, J.H. Kang, Automation in construction scheduling: a review of the literature, *Int. J. Adv. Manuf. Technol.* 81 (2015) 1845–1856, <https://doi.org/10.1007/s00170-015-7339-0>.
- [6] Y. Wang, Z. Yuan, C. Sun, Research on assembly sequence planning and optimization of precast concrete buildings, *J. Civ. Eng. Manag.* 24 (2018) 106–115. URL: <https://journals.vilniustech.lt/index.php/JCEM/article/view/458>.
- [7] V. Faghghi, K.F. Reinschmidt, J.H. Kang, Construction scheduling using genetic algorithm based on building information model, *Expert Syst. Appl.* 41 (2014) 7565–7578. URL: <https://www.sciencedirect.com/science/article/pii/S0957417414003340>, doi:.
- [8] P. Jiménez, Survey on assembly sequencing: a combinatorial and geometrical perspective, *J. Intell. Manuf.* 24 (2013) 235–250, <https://doi.org/10.1007/s10845-011-0578-5>.
- [9] S. Ghandi, E. Masehian, Review and taxonomies of assembly and disassembly path planning problems and approaches, *Comput. Aided Des.* 67 (2015) 58–86, <https://doi.org/10.1016/j.cad.2015.05.001>.
- [10] M.R. Bahubalendruni, B.B. Biswal, A review on assembly sequence generation and its automation, *Proc. Inst. Mech. Eng. C J. Mech. Eng. Sci.* 230 (2016) 824–838, <https://doi.org/10.1177/0954406215584633>.
- [11] M.V.A.R. Bahubalendruni, A.K. Gulivindala, S.S.V.P. Varupala, D.K. Palavalasa, Optimal assembly sequence generation through computational approach, *Sādhanā* 44 (2019) 174, <https://doi.org/10.1007/s12046-019-1157-2>.
- [12] M.V.A.R. Bahubalendruni, B.B. Biswal, M. Kumar, R. Nayak, Influence of assembly predicate consideration on optimal assembly sequence generation, *Assem. Autom.* 35 (2015) 309–316, <https://doi.org/10.1108/AZ-03-2015-0022>.
- [13] R.H. Wilson, On Geometric Assembly Planning, Ph.D. thesis, Stanford University, 1992. URL: <https://apps.dtic.mil/st/citations/ADA325963>.
- [14] M. Shpitlani, G. Elber, E. Lenz, Automatic assembly of three-dimensional structures via connectivity graphs, *CIRP Ann.* 38 (1989) 25–28. URL: <https://www.sciencedirect.com/science/article/pii/S0007850607626444>.
- [15] Q. Su, A hierarchical approach on assembly sequence planning and optimal sequences analyzing, *Robot. Comput. Integr. Manuf.* 25 (2009) 224–234. URL: <https://www.sciencedirect.com/science/article/pii/S0736584507001305>.
- [16] D. Halperin, J.-C. Latombe, R.H. Wilson, A general framework for assembly planning: the motion space approach, *Algorithmica*, *Int. J. Comp. Sci.* 26 (2000) 577–601, doi: 10/b2rq5w.
- [17] C.R. Garrett, Y. Huang, T. Lozano-Pérez, C.T. Mueller, Scalable and probabilistically complete planning for robotic spatial extrusion, in: *Robotics: Science and Systems (RSS)*, 2020. URL: <https://roboticsconference.org/2020/program/papers/92.html>.
- [18] Y. Tian, J. Xu, Y. Li, J. Luo, S. Sueda, H. Li, K.D.D. Willis, W. Matusik, Assemble them all: physics-based planning for generalizable assembly by disassembly, *ACM Trans. Graph.* 41 (2022) 1–11, <https://doi.org/10.1145/3550454.3555525>.
- [19] E. Masehian, S. Ghandi, ASPPR: a new assembly sequence and path planner/replanner for monotone and nonmonotone assembly planning, *Comput. Aided Des.* 123 (2020) 102828, <https://doi.org/10.1016/j.cad.2020.102828>.
- [20] Y. Tian, K.D.D. Willis, B.A. Omari, J. Luo, P. Ma, Y. Li, F. Javid, E. Gu, J. Jacob, S. Sueda, H. Li, S. Chitta, W. Matusik, Asap: Automated Sequence Planning for Complex Robotic Assembly with Physical Feasibility, [arXiv:2309.16909](https://arxiv.org/abs/2309.16909), 2023.
- [21] M. Deuss, D. Panozzo, E. Whiting, Y. Liu, P. Block, O. Sorkine-Hornung, M. Pauly, Assembling self-supporting structures, *ACM Trans. Graphics (TOG)* 33 (2014) 214, <https://doi.org/10.1145/2661229.2661266>.
- [22] M. Agrawala, D. Phan, J. Heiser, J. Haymaker, J. Klingner, P. Hanrahan, B. Tversky, Designing effective step-by-step assembly instructions, in: *ACM Transactions on Graphics (TOG)* 22, ACM, 2003, pp. 828–837, <https://doi.org/10.1145/882262.882352>.
- [23] J. Heiser, D. Phan, M. Agrawala, B. Tversky, P. Hanrahan, Identification and validation of cognitive design principles for automated generation of assembly instructions, in: *Proceedings of the Working Conference on Advanced Visual Interfaces, AVI '04*, Association for Computing Machinery, New York, NY, USA, 2004, pp. 311–319, <https://doi.org/10.1145/989863.989917>.
- [24] G. Chen, J. Zhou, W. Cai, X. Lai, Z. Lin, R. Menassa, A framework for an automotive body assembly process design system, *Comput. Aided Des.* 38 (2006) 531–539, <https://doi.org/10.1016/j.cad.2006.01.012>.
- [25] Y. Qin, Z. Xu, Assembly process planning using a multi-objective optimization method, in: *2007 International Conference on Mechatronics and Automation, 2007*, pp. 593–598, <https://doi.org/10.1109/ICMA.2007.4303610>.
- [26] N. Mellado, P. Song, X. Yan, C.-W. Fu, N.J. Mitra, Computational design and construction of notch-free reciprocal frame structures, in: P. Block, J. Knippers, N. J. Mitra, W. Wang (Eds.), *Advances in Architectural Geometry 2014*, Springer International Publishing, Cham, 2015, pp. 181–197, https://doi.org/10.1007/978-3-319-11418-7_12.
- [27] M.K. Gelber, G. Hurst, R. Bhargava, Freeform assembly planning, *IEEE Trans. Autom. Sci. Eng.* 16 (2018) 1315–1329, <https://doi.org/10.1109/TASE.2018.2878670>.
- [28] B. Kerbl, D. Kalkofen, M. Steinberger, D. Schmalstieg, Interactive disassembly planning for complex objects, *Comp. Graphics Forum* 34 (2015) 287–297, <https://doi.org/10.1111/cgf.12560>.
- [29] J. Fitchen, *Building Construction before Mechanization*, Mit Press, 1989. URL: <https://mitpress.mit.edu/9780262560474/building-construction-before-mechanization/>, ISBN: 9780262560474.
- [30] M. Rippmann, *Funicular Shell Design: Geometric Approaches to Form Finding and Fabrication of Discrete Funicular Structures*, Doctoral thesis, ETH Zurich, 2016, <https://doi.org/10.3929/ethz-a-010656780>. URL: <https://www.research-collection.ethz.ch/handle/20.500.11850/116926>. accepted: 2017-11-24T16:15:32Z.
- [31] A. Ramfrez Ponce, R. Ramírez Melendez, Curves of Clay: Mexican Brick Vaults and Domes, 2015, pp. 309–324, https://doi.org/10.1007/978-3-319-00137-1_21.
- [32] M. Sanchez-Arcas, *Form Und Bauweise der Schalen*, Verlag für Bauwesen, 1961.
- [33] D. Kajaia, Precast spherical cupola roof for sport palace in Tbilisi and its erection by overhang method, in: *Symposium on Problems of Interdependence of Design and Construction of Large-Span Shells for Industrial and Civil Buildings (IASS)*, Leningrad, 1966.
- [34] E.N. Kaplunovich, C. Meyer, Shell construction with precast elements, *Concr. Int.* 4 (1982) 37–43.
- [35] L. Beyeler, J.-C. Bazin, E. Whiting, A graph-based approach for discovery of stable deconstruction sequences, in: P. Block, J. Knippers, N.J. Mitra, W. Wang (Eds.), *Advances in Architectural Geometry 2014*, Springer International Publishing, Cham, 2015, pp. 145–157. URL: http://link.springer.com/10.1007/978-3-319-11418-7_10.
- [36] K. Wu, A. Kilian, Robotic equilibrium: Scaffold free arch assemblies, in: *Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, 2018, p. 8, <https://doi.org/10.52842/conf.acadia.2018.342>.
- [37] S. Parascho, I.X. Han, S. Walker, A. Beghini, E.P.G. Bruun, S. Adriaenssens, Robotic vault: a cooperative robotic assembly method for brick vault construction, *Construction, Robotics.* 4 (2020) 117–126. URL: <http://link.springer.com/10.1007/s41693-020-00041-w>, doi: 10/gkh58b.
- [38] S. Parascho, I.X. Han, A. Beghini, M. Miki, S. Walker, E.P.G. Bruun, S. Adriaenssens, Lightvaul: A design and robotic fabrication method for complex masonry structures, in: *Proceedings of Advances in Architectural Geometry 2020*, 2020. URL: <https://oai.princeton.edu/handle/88435/pr1s17ss4d>.
- [39] J. Wang, W. Liu, G.T.-C. Kao, I. Mitropoulou, F. Ranaudo, P. Block, B. Dillenburger, Multi-Robotic Assembly of Discrete Shell Structures, De Gruyter, 2023, pp. 261–274. URL: <https://www.degruyter.com/document/doi/10.1515/9783111162683-020/html>.
- [40] E. Nozhova, Assembled without scaffolding: the construction of schukhov timber lattice hyperboloids, in: *Proceedings of the Fifth International Construction History Congress (SICCH)*, Chicago June 2015, volume 3, 5th International Congress on Construction History (SICCH), 2015, pp. 67–74.
- [41] Y. Huang, J. Zhang, X. Hu, G. Song, Z. Liu, L. Yu, L. Liu, Framefab: robotic fabrication of frame shapes, *ACM Trans. Graphics (TOG)* 35 (2016) 224, <https://doi.org/10.1145/2980179.2982401>.
- [42] K. Hayashi, M. Ohsaki, M. Kotera, Assembly sequence optimization of spatial trusses using graph embedding and reinforcement learning, *J. Int. Assoc. Shell Spatial Struct.* 63 (2022) 232–240, <https://doi.org/10.20898/j.iss.2022.016>.
- [43] Z. Wang, F. Kennel-Maushart, Y. Huang, B. Thomaszewski, S. Coros, A temporal coherent topology optimization approach for assembly planning of bespoke frame structures, *ACM Trans. Graph.* 42 (2023), <https://doi.org/10.1145/3592102>.
- [44] A. Thoma, A. Adel, M. Helmreich, T. Wehrle, F. Gramazio, M. Kohler, Robotic fabrication of bespoke timber frame modules, in: J. Willmann, P. Block, M. Hutter, K. Byrne, T. Schork (Eds.), *Robotic Fabrication in Architecture, Art and Design 2018*, Springer International Publishing, Cham, 2019, pp. 447–458, https://doi.org/10.1007/978-3-319-92294-2_34.
- [45] S. Parascho, T. Kohlhammer, S. Coros, F. Gramazio, M. Kohler, Computational design of robotically assembled spatial structures: A sequence based method for the generation and evaluation of structures fabricated with cooperating robots, in: *AAG 2018: Advances in Architectural Geometry 2018*, Klein Publishing, 2018,

- pp. 112–139. URL: <https://www.research-collection.ethz.ch/handle/20.500.11850/298876>.
- [46] E. Bruun, E. Besler, S. Adriaenssens, S. Parascho, Zerowaste: Towards computing cooperative robotic sequences for the disassembly and reuse of timber frame structures, in: Proceedings of the 42nd Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA), 2022. URL: <https://oar.princeton.edu/handle/88435/pr1wp9t657>.
- [47] I.X. Han, F. Meggers, S. Parascho, Bridging the collectives: a review of collective human–robot construction, *Int. J. Archit. Comput.* 19 (2021) 512–531, <https://doi.org/10.1177/14780771211025153>.
- [48] D. Mitterberger, L. Atanasova, K. Dörfler, F. Gramazio, M. Kohler, Tie a knot: human–robot cooperative workflow for assembling wooden structures using rope joints, *Constr. Robot.* 6 (2022) 277–292, <https://doi.org/10.1007/s41693-022-00083-2>.
- [49] I.X. Han, S. Parascho, Improv-structure: Exploring improvisation in collective human–robot construction, in: A. Gomes Correia, M. Azenha, P.J.S. Cruz, P. Novais, P. Pereira (Eds.), Trends on Construction in the Digital Era, Lecture Notes in Civil Engineering, Springer International Publishing, Cham, 2023, pp. 233–243, https://doi.org/10.1007/978-3-031-20241-4_16.
- [50] Z. Wang, P. Song, M. Pauly, Desia: a general framework for designing interlocking assemblies, *ACM Trans. Graph.* 37 (2018), <https://doi.org/10.1145/3272127.3275034>.
- [51] R. Desai, J. McCann, S. Coros, Assembly-aware design of printable electromechanical devices, in: The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18, ACM Press, Berlin, Germany, 2018, pp. 457–472. URL: <http://dl.acm.org/citation.cfm?doid=3242587.3242655>. doi: [10/gg55jt](https://doi.org/10/gg55jt).
- [52] S. Huber, R. Poranne, S. Coros, Designing actuation systems for animatronic figures via globally optimal discrete search, *ACM Trans. Graph.* 40 (2021) 1–10, <https://doi.org/10.1145/3450626.3459867>.
- [53] Z. Wang, P. Song, M. Pauly, State of the art on computational design of assemblies with rigid parts, *Comp. Graphics Forum* 40 (2021) 633–657, <https://doi.org/10.1111/cgf.142660>.
- [54] Y. Zhang, L. Tatarintseva, T. Clewlow, E. Clark, G. Botsford, K. Shea, A template design and automated parametric model for sustainable corbel dwellings with interlocking blocks, *Develop. Built Environ.* 14 (2023) 100148, <https://doi.org/10.1016/j.dibe.2023.100148>.
- [55] M. Motamedi, R. Mesnil, R. Oval, O. Baverel, Scaffold-free 3d printing of shells: introduction to patching grammar, *Autom. Constr.* 139 (2022) 104306, <https://doi.org/10.1016/j.autcon.2022.104306>.
- [56] W. Wang, D. Munro, C.C.L. Wang, F. van Keulen, J. Wu, Space-time topology optimization for additive manufacturing: concurrent optimization of structural layout and fabrication sequence, *Struct. Multidiscip. Optim.* 61 (2020) 1–18, <https://doi.org/10.1007/s00158-019-02420-6>.
- [57] G.T. Kao, A. Körner, D. Sonntag, L. Nguyen, A. Menges, J. Knippers, Assembly-aware design of masonry shell structures: A computational approach, in: Proceedings of IASS Annual Symposia, International Association for Shell and Spatial Structures (IASS), 2017, <https://doi.org/10.3929/ethz-b-000554318>.
- [58] S.H. Zargar, R.M. Leicht, A.R. Wagner, N.C. Brown, Integrating early assessment of robotic constructability into design optimization of a standalone classroom, *Autom. Constr.* 157 (2024), <https://doi.org/10.1016/j.autcon.2023.105175>. URL: <https://www.sciencedirect.com/science/article/pii/S0926580523004351>.
- [59] J. Ochshorn, Structural Elements for Architects and Builders: Design of columns, beams, and tension elements in wood, steel, and reinforced concrete, third edition ed, 2022. Independently published. URL: <https://jonochnoshorn.com/structuralelements/index.html>.
- [60] J. Fraser, D. Seel, J. Chadwick, K. Valambhan, A. Offiler, Cannon Place, London: Design and construction over a live railway station, in: Proceedings of the Institution of Civil Engineers - Civil Engineering 165, 2012, pp. 74–81. URL: <https://www.icevirtuallibrary.com/doi/10.1680/cien.2012.165.2.74>, <https://doi.org/10.1680/cien.2012.165.2.74>.
- [61] S.J. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, third edition ed, Pearson Education Limited, Malaysia, 2010 (ISBN: 0134610997).
- [62] I.P.A. Papadopoulos, P.E. Farrell, T.M. Surowiec, Computing multiple solutions of topology optimization problems, *SIAM J. Sci. Comput.* 43 (2021) A1555–A1582, <https://doi.org/10.1137/20M1326209>.
- [63] T. Schiavinotto, T. Stützle, A review of metrics on permutations for search landscape analysis, *Comput. Oper. Res.* 34 (2007) 3143–3153. URL: <https://www.sciencedirect.com/science/article/pii/S0305054805003746>.
- [64] K. Sörensen, M. Reimann, C. Prins, Permutation distance measures for memetic algorithms with population management, in: Proceedings of MIC, 2005, 6th Metaheuristics International Conference, Vienna, Austria, August 22–26, 2005, 2005, https://doi.org/10.1007/978-3-319-10762-2_37.
- [65] M. Zaefferer, J. Stork, T. Bartz-Beielstein, Distance measures for permutations in combinatorial efficient global optimization, in: T. Bartz-Beielstein, J. Branke, B. Filipič, J. Smith (Eds.), *Parallel Problem Solving from Nature – PPSN XIII*, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2014, pp. 373–383, https://doi.org/10.1007/978-3-319-10762-2_37.
- [66] L.B. Leopold, Trees and streams: the efficiency of branching patterns, *J. Theor. Biol.* 31 (1971) 339–354. URL: <https://www.sciencedirect.com/science/article/pii/0022519371901925>.
- [67] I. Md Rian, M. Sassone, Tree-inspired dendriforms and fractal-like branching structures in architecture: a brief historical overview, *Front. Architect. Res.* 3 (2014) 298–323. URL: <https://www.sciencedirect.com/science/article/pii/S2095263514000363>.
- [68] C. Bañón, F. Raspall, Architecture meets organic matter: Sombra verde and white spaces, in: 3D Printing Architecture: Workflows, Applications, and Trends, Springer, Singapore, 2021, pp. 53–69, https://doi.org/10.1007/978-981-15-8388-9_5.
- [69] B. Jiang, J. Zhang, M. Ohsaki, Optimization of branching structures for free-form surfaces using force density method, *J. Asian Architect. Build. Eng.* 21 (2022) 1458–1471, <https://doi.org/10.1080/13467581.2021.1928509>.
- [70] C. Preisinger, Linking structure and parametric geometry, *Archit. Des.* 83 (2013) 110–113, <https://doi.org/10.1002/ad.1564>.
- [71] Robert McNeel & Associates, Rhino3D® and Grasshopper3D®, 2022. URL: <http://www.rhino3d.com>.
- [72] D. Furcy, S. Koenig, Limited discrepancy beam search, in: Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI'05, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 2005, pp. 125–131. URL: <https://dl.acm.org/doi/10.5555/1642293.1642313>.
- [73] R. Zhou, E.A. Hansen, Beam-stack search: Integrating backtracking with beam search, in: ICAPS, 2005, pp. 90–98. URL: <https://cdn.aaai.org/ICAPS/2005/ICAPS05-010.pdf>.
- [74] Y. Huang, C.R. Garrett, I. Ting, S. Parascho, C.T. Mueller, Robotic additive construction of bar structures: unified sequence and motion planning, construction, *Robotics* 5 (2021) 115–130, <https://doi.org/10.1007/s41693-021-00062-z>.