



Structural rigidity theory applied to the scaffold-free (dis)assembly of space frames using cooperative robotics

Edvard P.G. Bruun^{b,*}, Sigrid Adriaenssens^b, Stefana Parascho^a

^a School of Architecture, Civil and Environmental Engineering, EPFL, Lausanne, Switzerland

^b Civil and Environmental Engineering Department, Princeton University, Princeton, NJ, USA

ARTICLE INFO

Keywords:
space frame
robot
fabrication
topology
assembly
disassembly
scaffold-free

ABSTRACT

This paper presents a fabrication-informed design method for triangulated space frame structures that remain stable during all phases of their robotic assembly and disassembly without requiring external scaffolding. A graph theoretic framework, based on rigidity theory, is developed to allow the structure, its support conditions, and the impact of robotic support constraints to be simultaneously represented in a single topological framework. The structural system is sequentially designed with an assembly logic based on Henneberg graph-construction steps, which are executed with two robots through a cooperative rigidity-preserving sequence. Ensuring planarity of the resulting graph during these construction steps is shown to lead to intrinsic disassembly potentials within the system. A graph-based algorithm is presented to locate, isolate and remove locally rigid tetrahedral cells formed in the structural system. This algorithm is then utilized to compute a rigidity-preserving robotic disassembly sequence. The design method is demonstrated in the case study design of a wooden space frame arch structure that is robotically (dis)assembled.

1. Introduction

Traditionally, architects and structural engineers place emphasis on the preliminary design of a structure in its finished state [1], but rarely account for the relationship between structural form and the resulting construction sequence that is necessary to reach this finished state. This can result in an expensive, time-consuming, or materially wasteful construction process. Construction costs can be substantial – for example, formwork amounts to about 40% of the total cost for reinforced concrete thin shells [2]. Looking at the other end of the building life-cycle, a structure is rarely designed with considerations for its efficient disassembly and potential reuse as a means to mitigate the large amount of waste construction activities currently contribute to landfill volumes [3]. Such a wasteful single-use design philosophy serves as a negative multiple on the high embodied energy associated with resource extraction and material processing requirements for building components [4]. Technological developments to improve construction efficiency and material usage can help address some of these environmental impacts. The challenge is that the Architecture, Engineering and Construction (AEC) industry lags other industries when it comes to leveraging contemporary automation techniques and the associated

productivity and environmental benefits [5]. The motivation behind this research is thus to aid in addressing the waste generation and productivity gap in the construction industry [6], specifically when considering geometrically complex truss and space frame structures. This is accomplished through the development of a structural design framework for efficient construction using a cooperative robotic fabrication setup. Efficiency, in the context of fabrication, is herein defined as maintaining stability without requiring external formwork or scaffolding during all stages of assembly or disassembly. This framework thus addresses the AEC industry challenges by both improving construction productivity through automation, reducing construction waste by eliminating the need for temporary support material, and planning for disassembly of the structure. Planning for disassembly as a core consideration in the design process provides opportunities for future structural reconfiguration and reuse [7–9].

Industrial robots are applicable to architectural fabrication setups for discrete element structures due to their application versatility [10] and spatial precision in picking and placing material [11]. They are also experiencing growing adoption in both industry and academic contexts [12] and are promising tools to tackle the AEC industry's lagging productivity and labor efficiency [13,14]. This paper presents a *fabrication-*

* Corresponding author.

E-mail address: ebruun@princeton.edu (E.P.G. Bruun).

<https://doi.org/10.1016/j.autcon.2022.104405>

Received 30 January 2022; Received in revised form 1 June 2022; Accepted 2 June 2022

Available online 17 June 2022

0926-5805/© 2022 Elsevier B.V. All rights reserved.

informed design framework, which is defined as a framework where the structural design directly accounts for how the structure will be built and taken apart, in this case utilizing the capabilities of a robotic setup (specifically with respect to sequencing). This constitutes a departure from a more traditional linear approach to construction, where a robot would act as a generic tool at the end of the workflow to materialize a structure designed in a separate design phase. On the other hand, a *fabrication-informed* framework specifically developed for a cooperative robotic setup, in this paper incorporates the multiple complimentary functions possible when several robots are working together: either placing members (during assembly) or removing parts of the structure (during disassembly), while also supporting the structure in its temporary state [15]. The proposed design method utilizes a graph-theoretic approach, based on rigidity theory, to link cooperative robotic construction with structural topology. This approach allows for the design of structures that have intrinsic scaffold-free assembly and disassembly potentials built directly into their preliminary design formulation.

1.1. Paper organization

Section 2 begins with a literature review on cooperative robotic fabrication and graph theory for the analysis of structural rigidity, followed by a more detailed explanation in Section 3 of the specific concepts from rigidity theory that are necessary when representing bar and joint frameworks (i.e., trusses and space frames). Section 4 presents a topology-driven design method for structures that can be assembled with cooperating robots in a rigidity-preserving manner. Section 5 then presents a graph-based algorithm that is used to plan a rigidity-preserving cooperative robotic disassembly sequence for the same structure. The design method is demonstrated in the case study design of a wooden space frame arch structure, with a discussion of the results from its physical assembly and disassembly presented in Section 6. The paper concludes in Section 7.

2. Literature review

2.1. Cooperative robotic fabrication in construction

In the AEC industry, industrial robotic arms were first applied to the construction of modular homes [16,17] and then in single-purpose automation systems [18–21]. Since their first adoption, robotic systems have continued to improve their functionality in unstructured environments [22,23], which has substantially improved their viability in the construction of structures with complex geometries [24]. Growth in the field of digital fabrication (DFAB) saw the first large-scale explorations in an architectural context of robots applied to the construction of geometrically complex structures [25–27]. At the outset, robots were used to build prefabricated load-bearing but non-standardized undulating walls made of discrete volumetric elements [28–30], followed by the DFAB house [31] as an example of how digital design and robotic fabrication processes can be used to build different non-standardized components and assembled in a structure [32–34]. Yet despite these advancements, when specifically applied to the construction of discrete element structures, robotic fabrication is still predominantly utilized for the construction of vertical layer-based structures [28–30,35–38]. A review of recent trends in multi-agent fabrication, applied in a collaborative (human-robot) or cooperative (robot-robot, cobot) framework, shows how multi-agent approaches can help to expand the feasible design space of robotic fabrication [39–41]. Maintaining stability and mitigating unwanted forces and deflections during construction can prove a significant challenge when using additive fabrication processes [42–44]. This can be a limitation to the design freedom of discrete element structures realized with robotic fabrication methods. One solution to the challenge of maintaining stability is to work with specially designed discrete elements that allow some level of interlocking to provide local stability during construction [45,46]. Another approach is

to utilize a cooperative robotic fabrication strategy (i.e., multiple robots working together to perform a task that cannot be performed with one robot alone), which can lead to new self-supporting construction processes, a concept proposed for space frame construction by Parascho [15,47,48]. In Parascho work, structural stability during construction was achieved without scaffolding by designing the structure specifically for fabrication using two robots sequenced in a cooperative manner. The robots take turns performing either the function of: (1) picking up and accurately placing structural components, or (2) holding and providing temporary support over indefinite periods of time to a partially completed structure. These are tasks that a human would struggle with but are well-suited to a robotic agent. Thus, these structures were only realizable in a self-stable way (i.e., without external scaffolding) when explicitly leveraging the use of multiple robots.

The support/place approach has been used in a number of subsequent cooperative robotic fabrication projects to construct complex discrete element structural forms in a range of materials and scales. For example, the fabrication of non-planar timber modules, where cooperating robots were used as a way to minimize the need for scaffolding in the intermediate stages of fabrication [49]. Several geometrically complex spanning structures have also been constructed without scaffolding using a cooperative robotic strategy: a bifurcating arch structure built out of foam blocks [50], and a doubly-curved brick shell [51–53], and a branching tree structure [54]. In general, the trend of moving from single to multiple agents for collaborative processes increases the complexity of the relationship between fabrication sequence and structural design. It is thus necessary to develop a design framework that can handle the increasing complexity of setups with multiple robots working together.

2.2. Graph theory and structural isomorphism

Designing a discrete element structure for stability during assembly and disassembly requires a numerical formulation that characterizes this objective. The goal is to integrate the fabrication sequence with the structural design, while considering the interaction of the multiple robotic agents without an overly prescriptive method that stifles geometric exploration and creativity in the preliminary design. These objectives are formulated here as a topological problem: relating the connection between the structural elements and the structural stability during construction, while taking into account the potential of utilizing robotic support. A topological problem can be modeled as a graph, $G = (V, E)$, which is a discrete mathematical structure made up of vertices, V , connected by edges, E , that represent pairwise relations between objects. The foundations of graph theory (i.e., the study of such structures) were laid by Euler in his solution to the *Seven Bridges of Königsberg* problem [55]. Graph theory has since seen broader application in the study of topology and in applied science applications to model how objects are distributed and connected in complex systems including circuits [56,57], elastic networks [58], and structures [59–61]. More recently, topological representations have also been used in geometric form-finding methods [62–64]. A graph is isomorphic to a physical system if it has a one-to-one mapping with the elements in the system it represents. Given such a representation, one can determine properties of the underlying system with graph theoretic approaches based only on topological information [65]. The general principle is that certain properties of the graph represent the intrinsic physical behavior of the system it represents. Isomorphic graphs have thus been used to show parallels in the analysis and formulation of a broad class of physical systems such as: trusses, dynamic mass-spring-damper oscillator systems, and planetary gear systems [65–67]. In the context of bar and joint frameworks (i.e., trusses and space frames), this relationship between a graph and its isomorphic structure can be used to compute force and displacement quantities [66,68,69] or generic properties such as rigidity and stability [65].

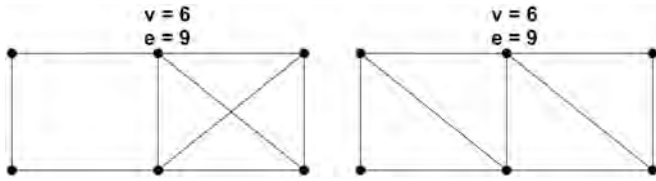


Fig. 1. Two trusses that both satisfy the planar Maxwell count condition, but only the right is infinitesimally rigid.

2.3. Infinitesimal rigidity and the Laman count condition

For a bar and joint framework, a necessary condition for infinitesimal rigidity was first formulated by Maxwell as a counting rule relating the number of nodes, v , to the number of members, e , in the framework. A structure needs a minimum of $e = 2v - 3$ members in $\mathbb{R}^2$, and a minimum of $e = 3v - 6$ members in $\mathbb{R}^3$ to be infinitesimally rigid [70]. An infinitesimally rigid structure is one where every infinitesimal motion is a Euclidean one (i.e., a rigid body motion); a structure is infinitesimally rigid if and only if it is statically rigid, meaning that every equilibrium load has a resolution [71]. The relationship between the number of nodes and members was later refined by Calladine to characterize the number of mechanisms present in a framework [72,73]. However, the overall Maxwell count condition is a necessary, but not sufficient, condition for infinitesimal rigidity. This is demonstrated with the two trusses in Fig. 1, where both satisfy the count condition, but only the right one is infinitesimally rigid. Thus, a more generalized approach, based on rigidity theory (a branch of graph theory), is necessary to determine if a bar and joint framework is rigid based on the topology of its isomorphic graph.

If for a graph, $G = (V, E)$, where the number of edges is $|E| = 2|V| - 3$, every subgraph, $G' = (V', E')$, satisfies the count condition, $|E'| \leq 2|V'| - 3$, then the graph is minimally rigid (i.e., infinitesimally rigid with the least amount of edges possible). Theorem 5.6 of Laman's paper [74] proved that this condition is not only necessary, but also sufficient, for a graph to be generically minimally rigid in the plane. The term *generic*, in reference to rigidity, refers to a graph that has at least one isomorphic realization (i.e., a specific mapping of vertices to physical node locations corresponding to a truss structure) that is statically rigid [71]. This theorem, when applied to the cases shown in Fig. 1, correctly identifies the 6-noded truss on the right as statically rigid with the minimum required 9 members. A sub-graph counting condition, with $|E'| \leq 3|V'| - 6$ instead, can also be used to determine if a graph is generically minimally rigid with respect to realizations in $\mathbb{R}^3$. These principles can be used to build generically rigid graphs, which thus have at least one isomorphic realization as a space frame in $\mathbb{R}^3$ that is isostatic (i.e., statically rigid and infinitesimally rigid). This procedure and certain caveats are further explained in Section 3.

2.4. Research contributions

This research extends past work on the scaffold-free cooperative robotic fabrication of space frames [15] by redefining the design process and assembly logic through a novel graph theoretic framework based on rigidity theory. A spatial structure is represented as its isomorphic graph [65], where the structural topology and influence of robotic support is simultaneously evaluated from the perspective of preserving rigidity in the graph. The topology-driven design method developed using this framework is able to unify the generation of a space frame geometry with the criteria that the resulting structure has a feasible scaffold-free assembly and disassembly sequence. The design method is also considered *fabrication-informed* since the resulting structure is directly designed based on what is possible when sequencing two robots in a cooperative manner. The following is a summary of this paper's main contributions:

- Presentation of the isomorphic *stability graph* as a unified framework to represent a structure and its supports, and the impact of external support (robotic or otherwise) on the rigidity of a graph.
- Recontextualization of an existing cooperative robotic assembly sequence for space frames [15] through the lens of rigidity-preserving Henneberg assembly steps used for the mathematical construction of rigid graphs.
- Development of a sequential topology-driven design method that links physical assembly and disassembly with the properties of graph planarity and minimal rigidity. These properties, resulting in a structure comprised of locally rigid tetrahedral cells, are used to guarantee that a scaffold-free assembly (member-by-member) and disassembly (cell-by-cell) sequence exist for the structure.
- Demonstration of the proposed method in the case study design of a space frame arch structure, and implementation of a cooperative robotic sequence for both its scaffold-free assembly and disassembly.

3. Graph representation for space frame structures

This section expands the general principles of rigidity theory and structural isomorphism introduced in Section 2.2 and 2.3 with specific concepts that are required to represent a space frame structure as a graph. This representation forms the basis for the design method that guarantees a rigidity-preserving assembly and disassembly sequence. The following topics are covered: Henneberg assembly steps to build generic minimally rigid graphs isomorphic to 3D bar and joint frameworks (Section 3.1), representing a minimally rigid tetrahedral building block (or cell) as a graph (Section 3.2), and accounting for structural supports and evaluating their impact on rigidity through a stability graph representation (Section 3.3). The terms *vertex* and *edge* are used when referring to a graph, and *node* and *element* are used when referring to a graph's *representation*, *realization* or *physical embedding* (used interchangeably) as a structure in $\mathbb{R}^2$ (i.e., a truss) or $\mathbb{R}^3$ (i.e., a space frame). The term *rigidity* will also henceforth be used to refer to *generic infinitesimal rigidity* unless noted otherwise.

3.1. Constructing rigid graphs: Henneberg assembly steps

While the Laman condition provides a definitive way to check rigidity in the plane ($\mathbb{R}^2$), its naïve algorithmic implementation results in an exponential time algorithm [65]. Modern algorithms exist that reduce the complexity of verifying minimal rigidity to polynomial [75,76], quadratic [77,78], or even sub-quadratic [79] time. However, these algorithms can be complex to implement. But rigidity can for large and complex graphs be more easily checked through a particular graph's sequential assembly sequence (if this information is available), since every minimally rigid graph can be built up starting from a single edge using only a simple set of rules. This process is named after Henneberg, who first introduced the idea of preserving rigidity in the plane by sequentially aggregating a low-valent vertex to the existing graph [80]. Thus, if a larger graph is found to be the result of a series of rigidity-preserving Henneberg steps then by inductive reasoning it is guaranteed to be rigid itself. Extending this concept to graphs that are to be embedded in space ($\mathbb{R}^3$), starting from a C_3 graph (i.e., 3-vertex cycle), every minimally rigid graph can be built up through a sequence of the following three Henneberg assembly steps [71], illustrated in Fig. 2:

- Type 1: (i) add a new vertex, (ii) connect the new vertex to any 3 vertices in the graph.
- Type 2: (i) delete any edge, (ii) add a new vertex, (iii) connect the new vertex to the 2 vertices incident to the deleted edge, (iv) connect the new vertex to any 2 other vertices in the graph.
- Type 3a: (i) delete any 2 edges, (ii) add a new vertex, (iii) connect the new vertex to the 4 vertices incident to the deleted edges, (iv) connect the vertex to any other vertex in the graph.

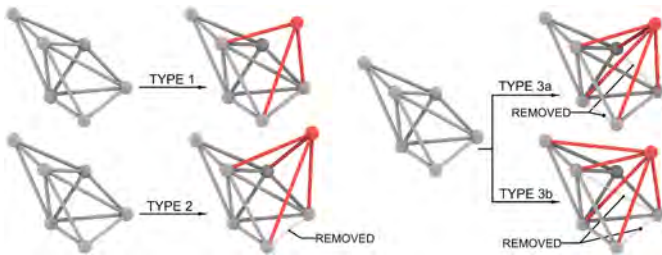


Fig. 2. Henneberg assembly steps for the construction of a graph with a generic rigid embedding in space ($\mathbb{R}^3$).

Type 3b: (i) delete any 2 edges, (ii) add a new vertex, (iii) connect the new vertex to the 3 vertices incident to the deleted edges, (iv) connect the vertex to any 2 other vertices in the graph.

Only after executing a Type 1 or Type 2 step is the resulting graph guaranteed to be a generically minimally rigid (i.e., it has at least one possible rigid realization in space). A Type 3 step has been shown to result in a graph without a rigid realization in some special cases [71,81,82]. The failure of Type 3 steps to definitively preserve rigidity is related to the fact that the Laman condition is a necessary, but not sufficient condition, for a minimally rigid graph embedded in space, as shown by the *double banana* counter-example in the literature [83,84]. Providing a combinatorial characterization of rigidity in space, as exists in the plane, is a major open problem in graph theory [85]. This issue can be mitigated through imposing additional topological and geometric constraints on the resulting physical structure that the graph is isomorphic to. For example, explicitly avoiding the formation of an axis of rotation in the *double banana* structure. Another way to preserve the property of rigidity in the overall graph is to construct the graph through the aggregation of smaller sub-graphs (referred to here as cells) with proven rigid realizations. Thus, the design method in this paper is based on a strictly Type 1 assembly logic, which is guaranteed to preserve rigidity. But using Type 2 or 3 Henneberg steps for assembling rigid structures is a possible foundation for the development of bidirectional (i.e., both adding and removing elements) fabrication sequences for more topologically varied structures.

3.2. The tetrahedral cell as a rigid building block

The tetrahedron, which can be represented with a skeletal framework of 4 nodes and 6 elements, is the simplest Platonic solid and corresponds to a minimally rigid cell from which larger rigid structures can be assembled. Due to their geometric simplicity, they are used as the fundamental building block for space frame structures designed to be assembled and disassembled with the cooperative robotic method described in this paper. More complex rigid cells, composed of more vertices, could also be used in this method, but their stability-preserving assembly feasibility is contingent on the number of robotic agents available in the setup. The graph representing the topology of a tetrahedron is minimally rigid, and has rigid realizations in $\mathbb{R}^3$ under the geometric constraint that no three nodes are collinear and that all four nodes are not coplanar [71]. These constraints, when coupled with the Laman count condition, are sufficient to guarantee rigidity in a structure that is built up from a collection of rigid tetrahedral cells. Fig. 3 illustrates the isomorphic graph representation of such a cell as a planar realization (i.e., no edges crossing) of the K_4 graph (i.e., complete graph on 4 vertices), which is built from, $G_0 = C_3$, with a Henneberg Type 1 step: $G_1 = (V_0 \cup \{3\}, E_0 \cup \{e_2, e_3, e_4\})$. The resulting graph satisfies the Laman count requirements, has a valid Henneberg construction sequence characterization, and known rigid realizations in $\mathbb{R}^3$ under the geometric constraints mentioned above. The representation of a

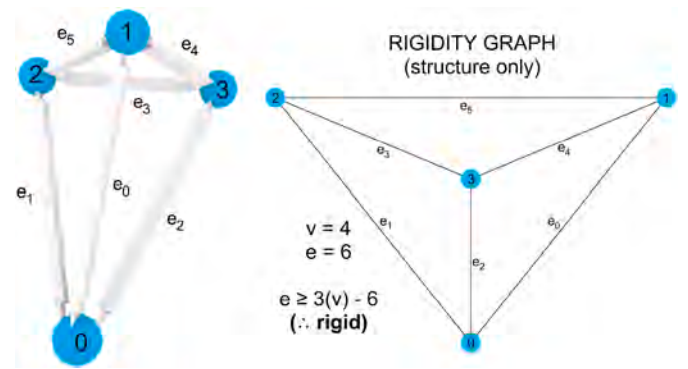


Fig. 3. Rigidity graph (right) and rigid realization (left) of a minimally rigid tetrahedral cell ($|V| = 4$, $|E| = 6$).

structure without support conditions, as in Fig. 3, is defined as a rigidity graph. Planarity is another important property of the graph of the tetrahedral cell, which will be discussed in the context of the assembly logic (Section 4.4) and the disassembly sequence algorithm (Section 5.2).

3.3. Structural supports represented in a graph

When representing a structure including its support conditions, its isomorphic graph is herein defined as a stability graph. For example, a structure might be rigid based on its topology but still undergo rigid-body motion based on its support conditions. One of the strengths of a topological approach is that the same methodology to assess rigidity, can be extended to evaluate stability based on the degrees of freedom (DOFs) being constrained at vertices in the graph. Thus, if the stability graph is found to be rigid (i.e., through the Laman count and Henneberg sequence characterization), then the structure it is isomorphic to is stable. This process is described for trusses ($\mathbb{R}^2$) in [65] and extended to space frames ($\mathbb{R}^3$) in this paper. A stability graph is built by representing the DOFs in the system as additional graph vertices, which for a graph realized in $\mathbb{R}^3$ are the three translations in Euclidean space. Edges are then added between these support vertices and the vertices that represent structural nodes restrained in translation by a support. Fig. 4 shows two different support conditions: (1) a single pin at the base of the structure, and (2) a base pin, roller, and directional pin (X & Z). Condition (1) is unstable as the structure, one rigid tetrahedral cell, is free to rotate about its base. This rigid body motion can equivalently be proved by analysis on the graph representation: each supported DOF is shown as an edge in the graph, $E_{\text{support}} = \{a, b, c\}$, hence the global count for this graph is now 12, which does not satisfy the required global Laman count, 15. Condition (2) on the other hand is stable, with $E_{\text{support}} = \{a, b, c, d, e, f\}$, hence the total count is 15, which satisfies the global Laman count, 15. But for a stability graph to be generically rigid it must also satisfy certain constraints when considering realizations in $\mathbb{R}^3$, similar to how the rigidity graph of the tetrahedral cell must satisfy certain geometric constraints as discussed in Section 3.2. If adjacent support edges are not independent this has the effect of removing a supported DOF from the system and destabilizing it. For example, in Fig. 4, if edges $\{e, f\}$ (i.e., $\{z, 2\}$ and $\{z, 3\}$) represent supports that are in the axis of member $\{2, 3\}$ (i.e., the support forces are collinear), then the system would not be stable. Additionally, a minimum of three nodes in the structure must at any time have at least one DOF constrained, to prevent an axis of rotation forming. This result corresponds to the principles of statics, where every equilibrium load can only be resolved through the global equations of force and moment equilibrium when precisely 6 independent support reactions are acting in at least one plane. Using this topological representation, unifying structure and support, the structure can be said to be stable if its stability graph is rigid. Minimal rigidity of

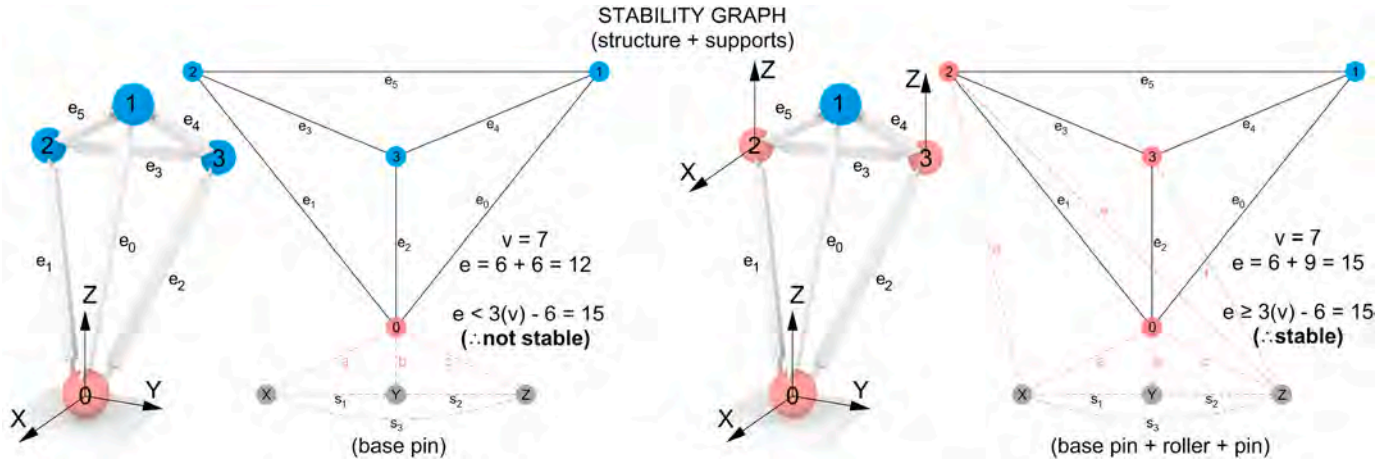


Fig. 4. Stability graphs representing two different sets of support conditions on the same minimally rigid space frame structure. Condition 1 (left) is unstable since $|E| = 12 < 3|V| - 6$, while condition 2 (right) is stable since $|E| = 15 \geq 3|V| - 6$, a Henneberg assembly sequence exists to build this graph, and the geometric constraints specified in Sections 3.2 and 3.3 are satisfied. Vertices/nodes in red are supported.

the stability graph is verified through the same process as the for the graph of the structure, by showing that there exists a sequence, S , of Henneberg Type 1 or 2 steps (h_1 or h_2) to build the final graph, G_n , starting from a rigid graph, $G_0 = (V_0, E_0)$:

$$\exists S\{H_1 \dots H_n\} \text{ s.t. } G_0 \rightarrow G_n \mid H \in \{h_1, h_2\} \quad (1)$$

There may exist numerous valid Henneberg sequences, S . The stability graph, G_3 , referring to support condition (2) in Fig. 4, can be built from the starting graph, G_0 , referring to the graph of the tetrahedral cell in Fig. 3, through one such sequence:

- $H_1 = h_1: G_1 = (V_0 \cup \{Z\}, E_0 \cup \{c, e, f\})$
- $H_2 = h_1: G_2 = (V_1 \cup \{Y\}, E_1 \cup \{b, s_2, (Y, 2)\})$
- $H_3 = h_2: G_3 = (V_2 \cup \{X\}, E_2 \setminus (Y, 2) \cup E_2 \cup \{a, d, s_1, s_3\})$

4. Designing a space frame structure for cooperative robotic assembly

4.1. Robotic support and Henneberg assembly

In this section, the principles of graph rigidity, and their characterization through Henneberg assembly steps, are implemented to develop a topology-driven design method for a space frame structure that can be assembled without scaffolding. In other words, the structure is to remain rigid without scaffolding during construction. The main objective is to design the structure explicitly considering a cooperative fabrication sequence with two robots [15]. Thus, the structure's stability graph must remain rigid throughout the full assembly process by accounting for the support contributions possible from the robotic fabrication setup itself. In addition, planarity must be preserved in the graph isomorphic to the structure, which is shown to result in the formation of locally rigid

cells arranged linearly in the overall structure. This property will be crucial in ensuring that the design method also embeds the existence of a scaffold-free disassembly sequence for the final structure. Thus, the design method is introduced through the lens of rigidity-preserving assembly, but has explicit considerations for disassembly in its formulation, which will be discussed further in the disassembly section of the paper (Section 5).

4.2. Cooperative robotic assembly strategy

The structural design method is based on a cooperative fabrication sequence with two robots (R_1 and R_2) that alternate their role in placing members and providing temporary support to the structure, as described in [15, p.74-77]. After placing a member, a robot is left supporting the structure, while the other robot is free to release the structure and retrieve and add the following member. The axial forces in individual members are resolved at concurrent points at the structural joints, illustrated for a system of three members in Fig. 6. There is only one possible physical arrangement of members at the joint level for each spatial mapping of the vertices. As a result, there exists a true one-to-one relation between the graph and the load path in the realized structure. A translational restraint shown as an edge in the graph, also represents the resulting force equilibrium in this pin-joint when a support is acting on it. This relation only exists for an idealized bar and joint structure where axial forces are resolved at concurrent points, because a graph representation only carries information about the connection topology. Capturing more complex equilibrium conditions, such as moments in the connections, requires information about offsets and distances not present in a topological representation. For example, the joint-level load paths cannot be fully characterized in a reciprocal frame style connection system where numerous relative member placements are possible for the same joint (as in previous work [47,48]).

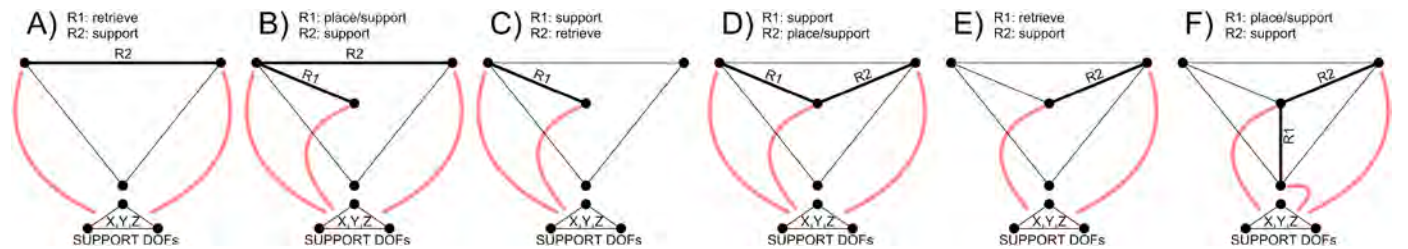


Fig. 5. The stability graphs at each of the 6 stages (A-F) for a Henneberg Type 1 step executed with two robots (R_1 and R_2) either supporting a member or retrieving the next member to add to the structure. Each edge highlighted in red represents three independent edges connecting a supported vertex in the structure to each of the three vertices representing the X, Y, Z translation constraints. For clarity, the graph of the rest of the structure and its non-robotic supports are not shown.

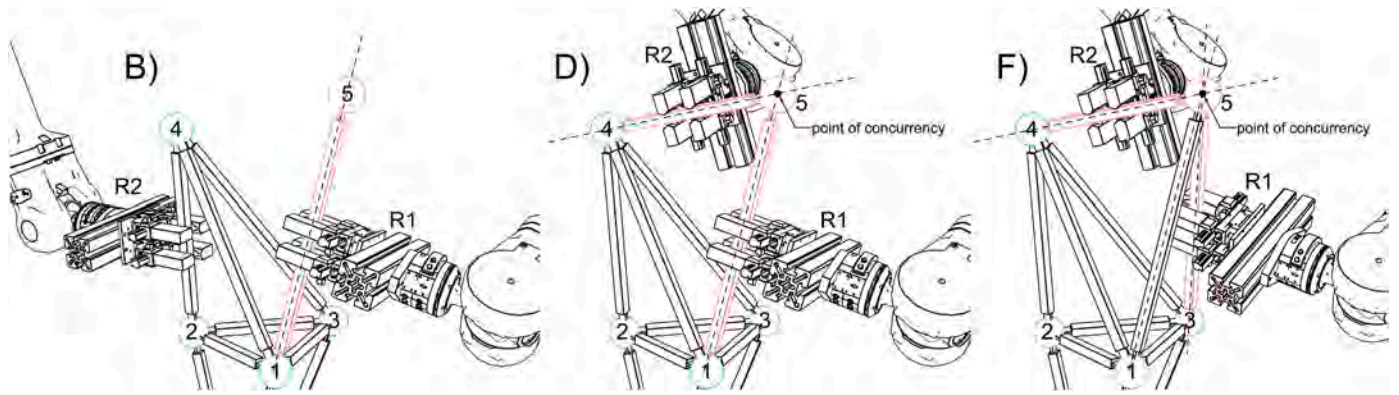


Fig. 6. Cooperative robotic sequence to execute a 3-member rigidity-preserving Henneberg Type 1 step for members with concurrent lines of action at the added node. Stage B (left): R1 = Place, R2 = Support; Stage D (middle): R1 = Support, R2 = Place; Stage F (right): R1 = Place, R2 = Support (adapted [15, Fig. 3.19])

In a structure constructed from concurrent members, according to a cooperative assembly strategy, the action of a temporary robotic support on the structure can be represented in the stability graph. The two vertices incident to the member a robot is supporting, are connected to the three graph vertices representing the translational constraints using the methodology described in Section 3.3 and shown during the execution of a Henneberg Type 1 step in Fig. 5. The robot thus affects the property of rigidity in the stability graph by altering the edge counts in the overall graph and its subgraphs. However, from the perspective of rigidity, despite utilizing the support provided by two robots during fabrication, there are limitations to the feasible sequence of adding members. For example, in a situation where elements are built off an existing rigid structure, but are placed without being incident to the same end node: one member can initially be placed and supported by either robot, but would later become unstable since it is only connected at one pinned end once the robot releases it to retrieve the next element. In the graph representation, this can be verified with a failure to achieve the global Laman count. One way to overcome this failure, is to add new members to the structure in mutually reinforcing ways (i.e., incident to the same new node) so that they assemble towards states of rigidity in the graph isomorphic to the structure. Building towards stages of rigidity can be evaluated through the lens of Henneberg assembly steps. A Type 1 step requires an additional vertex to be connected to the graph at three distinct vertices to maintain rigidity. In an applied context, during physical construction, such a step cannot happen instantaneously, but is instead executed through the sequential placement of three separate members. This process is shown in Fig. 5 through the stability graph representation at each of the 6 stages (placing and retrieving members) required to execute a full Henneberg sequence with two robots. The sequence starts with a generic rigid structure supported by a robot in stage A; for clarity the graph of the rest of the structure and its non-robotic foundation supports are not shown in Fig. 5. Each vertex that is supported by a robot at a stage is connected to the vertices representing the X, Y, Z translational support DOFs. During the first two member placements (stages B and D), there must be at least one robot supporting the partially completed Henneberg structure (i.e., the substructure of members connecting to the newly added node) for it to be rigid. Once the third and final member in the sequence is placed (stage F), the Henneberg step is complete, and the graph of the structure (in this case representing a simple tetrahedral cell) becomes minimally rigid without requiring robotic support. The aggregation process can then be repeated with the addition of another vertex to the existing graph, which is realized through the execution of another 3-member cooperative sequence with two robots. To complement the graph representation in Fig. 5, the physical representation of such a sequence is illustrated in Fig. 6 for the member placement stages (B, D, F) in the process of adding a node to an existing rigid structure. Depending upon the type of external non-robotic foundation support, a robotic support to the

structure might still be required to guarantee stability of the overall structure, even at the completion of a Henneberg step. It is possible for the rigidity graph to be rigid while the stability graph is not, as is the case for a structure supported on a single external base pin (i.e., Condition 1 in Fig. 4). In Section 3.3 it was explained that a total of 6 DOFs, represented as 6 edges in the graph, must be constrained on a minimum of 3 vertices (i.e., embedded as a plane) for the stability graph to remain rigid. This requirement is satisfied in a 3-member Henneberg sequence as there is always at least one robot supporting the structure. Since the robot constrains the movement of two vertices incident to the member it supports, this provides more support than is strictly necessary for the minimally rigid criteria. For example, the stability graph in Fig. 4 has 3 additional edges, $E = \{(Y, 2), (Y, 3), (X, 3)\}$, if member e_3 was supported by a robot (i.e., $V_{\text{support}} = \{2, 3\}$). But although such a structure would result in an over-constrained system (i.e., statically indeterminate), this over-constraining does not negatively impact the stability in the structure. Adding edges representing supports to a minimally rigid stability graph does not undermine its underlying property of rigidity; if a structure is stable under certain set of support conditions, additional supports only add to the structural redundancy of the system.

4.3. Assembly logic

A structure is designed sequentially to reflect its order of assembly, where each new node is connected back to three existing nodes in the structure, as per a Henneberg Type 1 step, forming a triangulated configuration as in [15, pg. 52-53]. This assembly logic has been recontextualized and developed in this paper through the graph-based analysis approach described in Section 4.1 and 4.2, to show that the properties of rigidity can be preserved at both the local and global level when utilizing a cooperative robotic place/support strategy as illustrated in Fig. 6. However, the assembly logic used here differs from previous work [15] as a new node in the structure is further specified to always connect to three nodes on the most recently formed tetrahedral cell. This approach guarantees that a rigid tetrahedral cell is formed after each 3-member aggregation phase. Thus, each added member is guaranteed to be part of at least one rigid cell. This is equivalent to specifying that a new vertex must be added to the current graph such that the resulting graph has a 1) planar realization (i.e., can be drawn with no edges crossings), and 2) linear partial order (i.e., a collection of rigid cells in series). The process of adding new nodes to three nodes on the most recently completed cell in the existing structure is not strictly necessary when just considering rigidity during assembly. But this aggregation logic becomes important when planning for a disassembly sequence as it guarantees the topological uniformity of the graph. The property of planarity and the linear partial order in the nested graph that is produced when utilizing this assembly logic guarantees that an efficient stability-preserving cooperative robotic sequence can later be

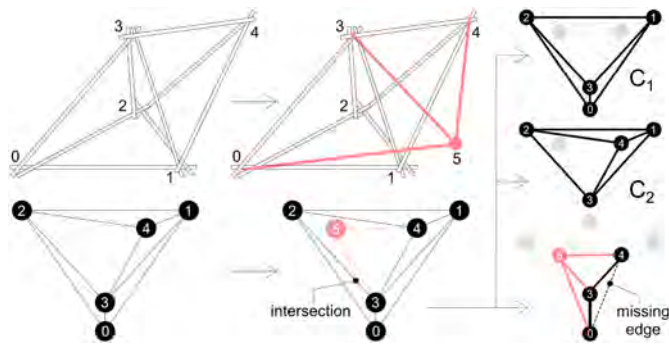


Fig. 7. An assembly approach where a new node can connect to any three nodes can result in a non-planar graph where a subgraph isomorphic to a tetrahedral rigid cell is not formed.

found for disassembly. Each member is guaranteed to be part of at least one rigid cell, which allows for an algorithmic process to locate, isolate, and remove these self-contained rigid cells during disassembly. This algorithm is explained in detail in Section 5.2. Thus, a design process utilizing the assembly logic described in this section is considered fabrication-informed as it has considerations for not just efficient assembly, but also for ease of disassembly, built directly into its formulation. In contrast, if an assembly logic based on connecting to any three nodes is used, the structure can still be assembled without requiring support but is now not guaranteed to be composed of rigid cells. This case is illustrated in Fig. 7, where node 5 is connected to the existing structure with $E = \{(0, 5), (3, 4), (4, 5)\}$. The result is a supergraph of the utility graph (UG), and is therefore non-planar [86]. It is possible to find two K_4 subgraphs, which correspond to the existence of two tetrahedral cells, C_1 and C_2 , in the original structure. But there is no such subgraph formed with the new edges in the graph, so no valid cell is formed when adding these members. A cell can be formed given the existing configuration by adding a single edge, $e = (0, 4)$, or by swapping $e = (0, 5)$ for $e = (1, 5)$, which can be verified by studying the case illustrated in Fig. 7. But in general, if such an assembly logic, allowing connection to any three nodes, is applied for a large structure with numerous assembly steps, there is no guarantee that the resulting topology can be characterized as a collection of elements that form tetrahedral cells.

4.4. Design method

The cooperative robotic support strategy and structural assembly logic can be implemented to design space frame structures that can be efficiently assembled and disassembled. The following steps are a schematic implementation of the concepts discussed thus far in this paper meant to guide the preliminary design of such structures:

1. Specify a curve, or a collection of curves, that represent the overall target form or backbone of the target structure.
2. Choose the target number of nodes that the structure is to be built from.
3. Specify geometric constraints on the realization of graph vertices. For example, those necessary in the formation of rigid realizations of tetrahedral cells Section 3.2.
4. Connect a new vertex to a region of the existing graph ensuring a tetrahedral cell topology while maintaining the property of planarity.
5. Calculate the physical realization of the new graph vertex on the basis of the geometric constraints specified.

Steps 4 and 5 occur in a loop executed as many times as specified in Step 3: sequentially updating the current isomorphic graph as each new vertex is added based on the physical realization of the structure from the previous stage. Step 4 is equivalent to assigning a valid topological configuration in the graph representation, and Step 5 maps this new graph vertex to a feasible structural node (i.e., physical embedding in space) - $N : V \rightarrow \mathbb{R}^3$ s.t. $n_a \neq n_b \mid (a, b) \in G(E)$. As per the assembly logic introduced in Section 4.3, each new vertex only has three potential regions it can connect to, with each corresponding to a face on the most recently completed tetrahedral cell in the structure. Once the region is selected, the new vertex is connected to each of the three adjacent vertices with an edge, as per a Henneberg Type 1 step, to preserve rigidity. This process can be thought of as sequentially building a graph with vertices that connect as a series of nested tetrahedral subgraphs. Choosing which of the three regions to connect to is determined by calculating which face in the most recently calculated cell has a normal vector closest to the tangent vector at a point on the target geometry curve closest to the cell. The process for selecting this face and choosing the region on the graph to add the next vertex to is shown schematically

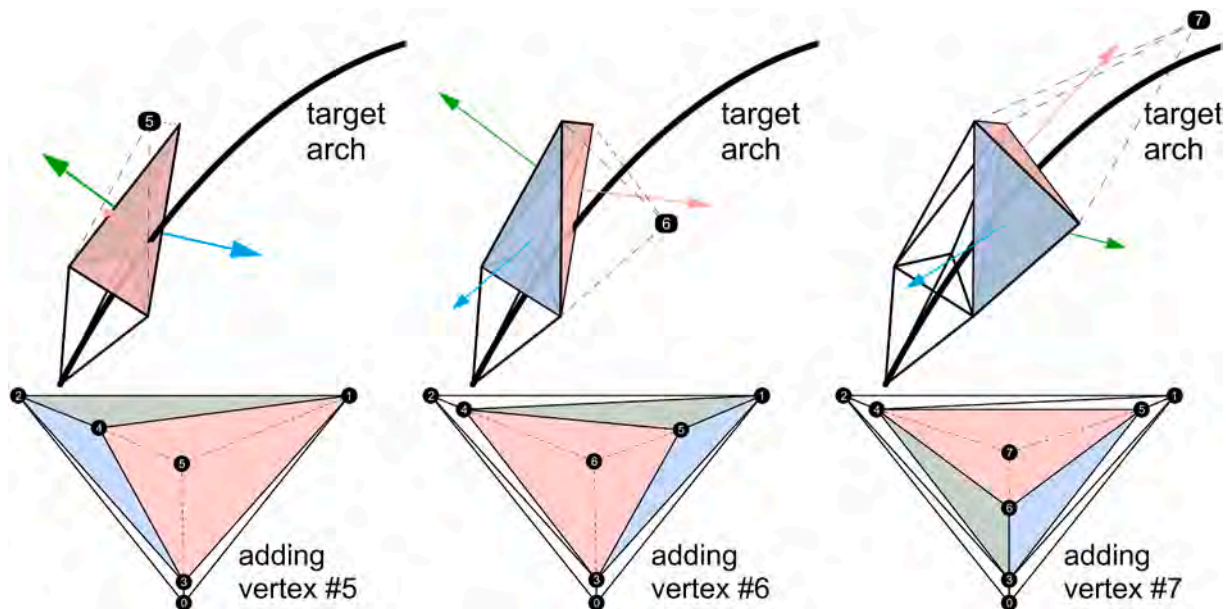


Fig. 8. The sequential design of a structure and its realization in space (top) corresponds to the sequential addition of vertices to the existing graph (bottom), forming a topology that can be characterized as a series of nested tetrahedral cells preserving the property of planarity in the overall graph at each aggregation phase.

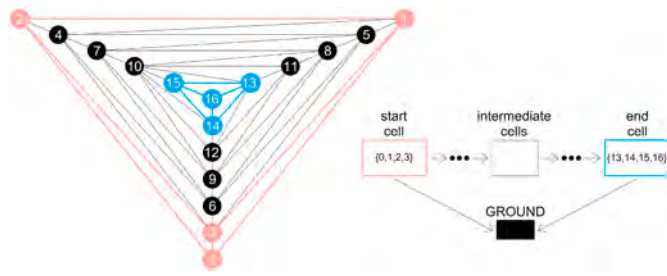


Fig. 9. The planar graph (left) isomorphic to the designed space frame structure (17 nodes and 45 elements), represented as a linear partial order of rigid tetrahedral cells (right).

for three sequential iterations in Fig. 8. The three potential regions in each iteration, and their corresponding faces on the structure, are shown in red, green, or blue on both the planar graph and its resulting embedding. In Fig. 8, the selected region and corresponding face to build from are shown in red; the vertices in the graph are shifted to make this region larger for clarity. The mapping of this new vertex is then calculated in Step 5 so that no new resulting member is co-linear with an existing member in the structure, and satisfying any remaining geometric constraints specified in Step 3.

4.5. Case study: design of a space frame arch structure

To test the implementation of the design method, a case study was performed with an input backbone curve of an arch, pin supported at its 2 extremities, with a final span of 3 m and a central height of 1.6 m. An arch is chosen as a structural system since its sequential fabrication requires working from one end to the other, which highlights the stability preserving nature of the fabrication method. If such a spanning structure was built using a more typical fabrication method (i.e., element-by-element construction starting from either end and meeting at the crown) it would be neither globally nor locally stable during construction without external scaffolding. Using the assembly logic and design method described in Section 4.3 and 4.4 respectively, an isomorphic planar graph is sequentially constructed. The graph ($|V| = 17$, $|E| = 45$) corresponding to the finished structure is shown in Fig. 9, with the corresponding vertex connectivity and embeddings in $\mathbb{R}^3$ that were computed at each stage summarized in Appendix A, Table A.1.

The only geometric constraint, in addition to those necessary for rigid realizations of tetrahedral cells outlined in Section 3.2, is that the vertex mapping function results in nodes satisfying the following: $0.3 \text{ m} \leq |n_a - n_b| \leq 1.2 \text{ m} \mid (a, b) \in G(E)$. Individual members must be longer than the width of the fingers spacing on robotic gripper to allow for an adequate gripping surface, but not exceed the length of the available construction material stock. Additional constraints based on optimization criteria (i.e., minimizing forces and displacements in the structure) can be used to further modify the mapping function to control of the design of the final structure. These were not implemented since the focus at this stage of the research is only linking topology and stability with the execution of a cooperative fabrication sequence.

The resulting arch structure starts and ends with the cells defined by the vertex subsets, $V_{start} = \{0, 1, 2, 3\}$ and $V_{end} = \{13, 14, 15, 16\}$, shown in the condensed partial order of rigid cells in Fig. 9 and in the rendering of the final structure in Fig. 10. This partial order specifies the directionality of the assembly, based on the nesting order of the planar K_4 graphs representing the tetrahedral cells (i.e., from outside $\rightarrow$ inside the graph). This direction is a function of how the graph is originally drawn. In some cases, it might be easier to draw new cells surrounding, rather than nesting into, the existing graph, thus the direction of assembly would be flipped (i.e., from inside $\rightarrow$ outside the graph).

The final space frame arch (Fig. 10) that results from the topology-driven approach to the construction of its isomorphic graph (Fig. 9), is

explicitly designed for fabrication efficiency. It is possible to assembly this structure without external scaffolding during any assembly step by utilizing robotic support.

5. Planning a cooperative robotic disassembly sequence

When designing a structure for robotic assembly, the emphasis is on allowing for geometric complexity in the design, which can be achieved by placing individual elements to build up a series of rigid cells. In contrast, a disassembly sequence is an operation on a completed structure, and thus a fixed final geometry. Therefore, the resulting process of taking the structure apart can occur through the hybrid removal of existing rigid cells and individual members, to minimize the total number of robotic operations necessary to complete the process. The fewer operations, the faster the full structure can be disassembled.

Planning the disassembly sequence is presented here based on subtractive operations on the graph representation of the structure. Instead of reducing the graph vertex by vertex, with inverse 3-member Henneberg steps, a more efficient approach is to locate, disconnect and then remove larger locally rigid regions of the graph. These regions correspond to the rigid cells formed in the structure during its design and assembly. The challenge is that their removal must be sequenced such that stability is maintained throughout the full disassembly process.

5.1. Cooperative robotic disassembly strategy

The approach to disassembly is similar to assembly in that the two robots are sequenced to provide temporary support to the structure while they also actively participate in its sequential disassembly. But although the overall goal of maintaining rigidity at intermediate stages is the same in both assembly and disassembly, the implementation here differs from assembly as performing the removal operation, does not need to occur element by element. The disassembly sequence takes advantage of the fact that the structure is specifically designed as a series of nested rigid tetrahedral cells, represented in a planar graph. A sequence can thus be determined where a rigid cell is located, supported, isolated from the rest of the structure, and then independently removed, while preserving the property of rigidity in the stability graph. Having the two robots dedicated to supporting the structure at all times offers more flexibility in this type of disassembly approach as it is now possible to support two distinct regions, and potentially disconnected components, of the structure at all times. The process starts by identifying a rigid cell in the structure that is targeted for removal, and assigning a robot to grab and support any member in this cell. Another distinct rigid cell, adjacent to the cell to be removed, is then located in the structure and the second robot is placed there as support. A distinct adjacent cell shares no members with another cell, and is separated by only one member from the cell it is adjacent to. These two cells, C_{remove} and $C_{support}$, are now supported by the two robots, and the edges in the graph between them become redundant from the point of view of minimal rigidity in the stability graph. These edges are effectively locked between two regions of the graph that are rigid when robotic support is provided to them. Thus, the physical members represented by these edges can be removed one by one without compromising either overall stability of the structure or the rigidity of the local region the members are adjacent to. Once the intermediate members are removed, the result is a graph partitioned into two independent rigid components: (1) a disconnected single rigid cell supported by a robot, (2) the remaining structure supported by the other robot and the external foundation support. The single cell can now be removed as a whole component from the work area by the robot supporting it, and the process of disconnection and removal begins again, now starting with the rigid cell that was supported by the other robot (i.e., $C_{support} \rightarrow C_{remove}$). This process is schematically shown in Fig. 11 for a simple 2D truss.

Using a cell-by-cell approach allows the full structure to be disassembled with fewer robotic repositioning steps than would be

required in a member-by-member approach. Determining which rigid cells to remove and support in each disassembly phase is performed through operations on the isomorphic graph representation – utilizing the property of planarity and the tetrahedral topology that was explicitly generated in the design process. A non-planar graph would imply a more complex connection hierarchy within the structure, representing substructures that are not independently rigid and can therefore not be isolated and removed in a stability-preserving way. The removal of the intermediate members is done manually, as an example of Human-Robot Interaction (HRI) in a collaborative fabrication process [39,54]. *Collaborative* refers to a process where the human works alongside the robot, while *cooperative* refers to a process where multiple robotic agents work together to accomplish a task one of them alone could not. The task of removing intermediate members is best performed manually, rather than robotically, because spatial accuracy and support capabilities are not required when members can be removed individually without compromising the stability of the structure.

5.2. Algorithm for locating rigid cells

Establishing a feasible disassembly sequence requires locating 1) rigid cells in the structure for the robots to support (Sections 5.2 and 5.3), and 2) intermediate elements to remove (Section 5.4). This procedure is performed through topological computations on the isomorphic planar graph generated from the design process described in Section 4.4. A graph-based algorithm finds rigid cells in the structure, and uses this information when planning a stability-preserving disassembly sequence. The rigid cell locating algorithm is initialized by choosing an edge in the graph to support, and then identifying the corresponding rigid tetrahedral cells to which it belongs. This procedure is described through pseudocode in Algorithm 1 and in greater detail in the remainder of this section.

Algorithm 1. (Rigid Tetrahedral Cell Locating Algorithm.)

```

1:  $C_{located} = []$ 
2: for Every  $e_{support} = \{v_1, v_2\} \in G'(E)$  do

3:   for Every  $v \in e_{support}$  do
4:     calculate closed neighborhood graph,  $N_G[v_i]$ 
5:   end for
6:   find common vertices,  $V' = N_G[v_1] \cap N_G[v_2]$ 

7:   for Every  $N_G$  do
8:     eliminate edges that are not adjacent to  $V'$ 
9:     remaining edges,  $E'_i$ , to form graph,  $N'_i = G(V', E'_i)$ 
10:  end for
11:  form adjacent cell subgraph,  $N' = N'_1 \cup N'_2$ 

12:  calculate complete edge set,  $E'_f$ 
13:   $E'_c = E'_f - N' =$  edges missing

14:  for Every  $e \in E'_c$  do
15:    form graph,  $C'_i$  by adding,  $e_i$ , to  $N'$ 
16:    form cell,  $C_i$ , by removing vertices not 3-connected
17:     $C_{located} += C_i$ 
18:  end for
19: end for

20: evaluate which cell(s) in  $C_{located}$  are adequate to support
    based on procedure described in Section 5.3

```

Next, the algorithm is further explained with a demonstration of its step-wise execution given an example starting edge, $e_{support} = (11, 12)$, operating on the graph generated in Section 4. This edge is indicated in the graph of the finished space frame arch structure designed in the case study (Fig. 12). The algorithm can be executed given a random edge, or an edge informed by the overall geometry and load-path. This decision is based on the type of structure being disassembled. For example, in a linear structure such as the arch, the disassembly should start from either end of the structure, rather than the middle, so that there are no more than two disconnected regions of the graph at any one time.

The process starts with choosing a set of supported edges in $G = (V, E)$, the planar isomorphic graph of the space frame structure. The supported vertices, $V_{support} = \{v_1 \dots v_n\}$, are adjacent to these edges. For a single supported edge, $n = 2$. A set, S , of closed neighborhood graphs (denoted as N_G) is then calculated from each supported vertex in G :

$$S = \{N_G[v] \mid v \in V_{support}\} \quad (2)$$

To simplify the notation, $N_G[v_i] = N_i$. Edges that are not components of the same rigid tetrahedral cells as the supported member must be removed from the neighborhood graphs. First, a subset of common vertices, V' , is created across the neighborhood graph vertices:

$$V' = N_1(V) \cap N_2(V) \quad (3)$$

Next, for each neighborhood graph, $N_i \in S$, a reduced neighborhood subgraph, $N'_i = (V', E'_i)$, is created by selecting only the edges that are adjacent to the common vertices in V' :

$$E'_i = \{(v_i, u) \in N_i \mid u \in V'\} \quad (4)$$

N' , defined as the *adjacent cell subgraph*, is then formed as the union of these reduced neighborhood graphs:

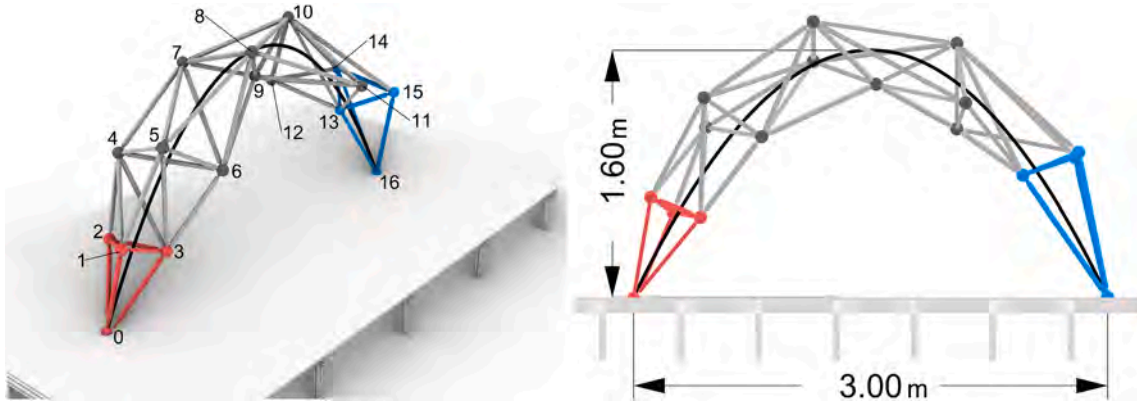


Fig. 10. Renderings of the designed space frame arch structure showing the node numbers and start/end tetrahedral cells (red/blue).

$$N' = N'_1 \cup N'_2 \quad (5)$$

N' only has vertices and edges that are part of a rigid cell that contains the supported element. Searching for a tetrahedral cell topology, it is evident that there are two potential cells that contain the supported member as seen in Fig. 13. But while all the vertices are present, there is one edge missing per cell.

The complete edge set, E'_f , for the rigid cells containing the supported member, is computed as all edges in the original graph, $G = (V, E)$, adjacent to the common vertices in V' :

$$E'_f = \{(u, v) \in G \mid u, v \in V'\} \quad (6)$$

The set of missing edges, E'_c , to complete the partial cells in N' , is computed as the difference between the full edge set, E'_f , and the edges of the adjacent cell subgraph:

$$E'_c = E'_f \setminus N'(E) = \{e_1 \dots e_n\} \quad (7)$$

The size of this edge set (i.e., $|E'_c|$) indicates the number of tetrahedral cells containing the supported element. This operation is based on the knowledge that the overall structure is specified to be made up of a series of tetrahedral rigid cells, thus the algorithm is locating as many copies of this specific topology that exist in the neighborhood of the supported member. The individual subgraphs of each cell are created by first adding each edge, $e_i \in E'_c$, separately to the adjacent cell subgraph:

$$C'_i = N'(V, E \cup e_i) \quad (8)$$

Then removing vertices, v , in the resulting graph, C_i , that are not 3-connected:

$$C_i = C'_i - \{v \mid v \in C'_i(V) \wedge \deg(v) \neq 3\} \quad (9)$$

In this example, with $e_{\text{support}} = (11, 12)$, the result is two cells, C_1 and C_2 , in the overall graph, G , that contain the supported edge, shown in Fig. 14. They have vertex sets $C_1(V) = \{9, 10, 11, 12\}$ and $C_2(V) = \{10, 11, 12, 13\}$, which share three common vertices (i.e., $|V_{C1} \cap V_{C2}| = 3$) because the cells share a common face.

It is possible that an edge is only part of one rigid cell (i.e., $|E'_c| = 1$). The reader can verify this result by following the steps outlined in this section given a different starting support edge (e.g., $e_{\text{support}} = (8, 11)$).

5.3. Selecting a rigid cell to support

In each disassembly phase, the current cell being isolated and removed is referred to as C_{remove} , and the other supported cell in the structure is referred to as C_{support} (Fig. 15). Once a phase is complete, $C_{\text{support}} \rightarrow C_{\text{remove}}$ in the next phase, and the process repeats. The algorithm described in Section 5.2 is executed multiple times per phase, varying the supported member initialization to generate an aggregated set of potential rigid cells, $C_{\text{located}} = \{C_1 \dots C_n\}$. Naively this requires $|G'(E)|$ runs for a complete characterization of the structure, where G' is a subgraph representing the structure in a partially disassembled state. In practice, this procedure can be reduced significantly as C_{support} must be located adjacent to C_{remove} in the structure. Once the set of rigid cells has been assembled, one cell from here must be chosen as the cell to support in the current phase. C_{support} is chosen based on the criteria that all intermediate edges connecting it to C_{remove} can be individually removed without breaking apart a non-supported cell in the structure or C_{remove} itself (i.e., compromising rigidity). Thus, the requirement for C_{support} is that it must be immediately adjacent to C_{remove} , expressed as the following two conditions:

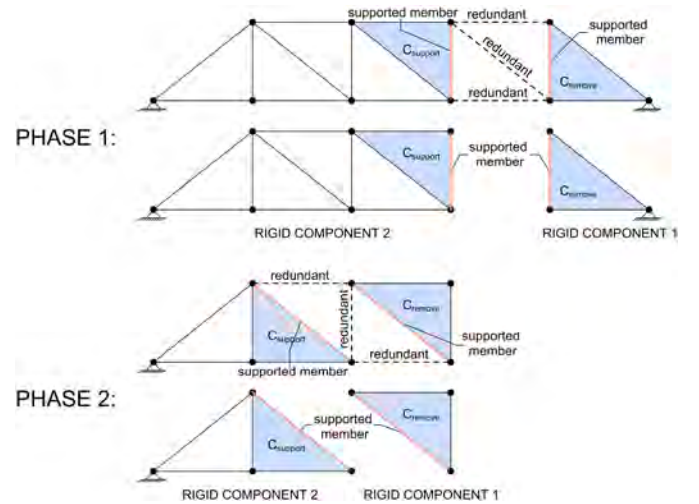


Fig. 11. Demonstration of a disassembly sequence on a 2D structure where rigid cells are sequentially located, supported, isolated and then removed from a structure.

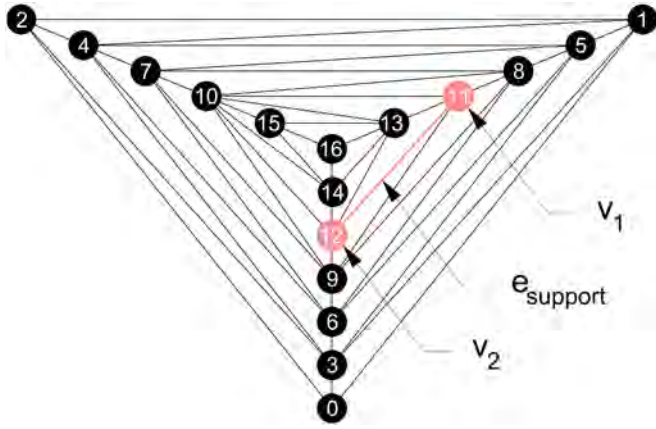


Fig. 12. Isomorphic planar graph of a space frame arch structure indicating a potential support edge, $e_{\text{support}} = (11, 12)$, used to initialize the rigid cell finding algorithm.

1. No vertices overlap between C_{support} and C_{remove} (i.e., not too close):

$$C_{\text{support}}(V) \cap C_{\text{remove}}(V) = \emptyset \quad (10)$$

2. No unsupported vertices between C_{support} and C_{remove} (i.e., not too far):

$$v \in \{C_{\text{support}}(V) \cup C_{\text{remove}}(V)\} \wedge v \in P \quad (11)$$

where, P , is a set of the vertices in all the possible walks of length two or less from all the vertices in both cells.

For example, Fig. 15 illustrates the two rigid cells, $\{C_1, C_2\}$, that are located when initializing the algorithm with $e_{\text{support}} = (11, 12)$, and the single cell, C_3 , that is located when initializing the algorithm with $e_{\text{support}} = (8, 11)$. C_2 fails criterion #1 (Eq. (10)) as one of the vertices overlaps with C_{remove} , thus it cannot be removed as a complete cell without destabilizing the rigid cell C_2 by removing one of its vertices. C_3 fails criterion #2 (Eq. (11)) as supporting this cell leaves a region of the remaining structure directly connected to C_{remove} unsupported. When removing individual members during the disassembly process this unsupported region of the graph is destabilized. Meanwhile, C_1 is feasible cell as a robot placed here supports the remaining structure while not leaving any unsupported intermediate vertices connected to C_{remove} . Thus, the intermediate members can be manually removed without compromising stability, as each member is immediately adjacent to a rigid supported cell.

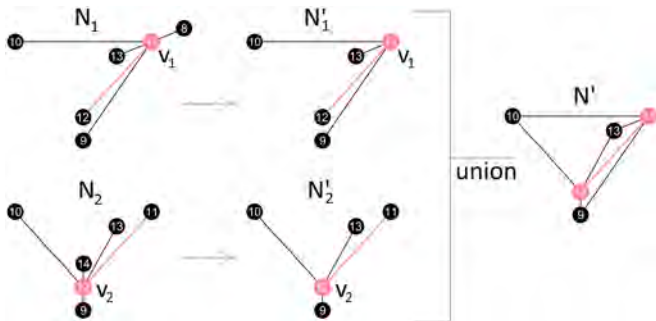


Fig. 13. The adjacent cell subgraph, N' , is formed from the union of the reduced support neighborhood subgraphs, N_1 and N_2 .

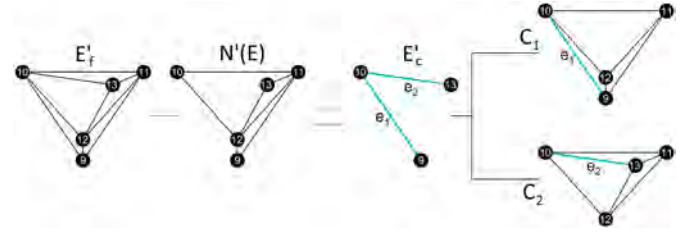


Fig. 14. Rigid tetrahedral cell subgraphs containing the supported member, $e_{\text{support}} = (11, 12)$. These cells are computed from the addition of a single edge, $e_i \in E_c$, to the cell adjacency subgraph, and then removing vertices that are not 3-connected.

5.4. Selecting individual members to remove

Once a viable rigid cell to support is chosen based on the criteria outlined in Section 5.3, the set of edges, E_{remove} , that connect C_{remove} and C_{support} , is computed as per Eq. (12). G' is the graph of the full structure in the current disassembly phase. Following the removal of the intermediate members in any order, the cell C_{remove} is disconnected from the remaining structure and is removed as an independent component.

$$E_{\text{remove}} = \{(u, v) \in G' \mid u \in C_{\text{remove}}, v \in C_{\text{support}}\} \quad (12)$$

5.5. Case study: planning the disassembly sequence for a spanning space frame arch

Computing a stability-preserving disassembly sequence using two robots is implemented on the space frame structure designed in Section

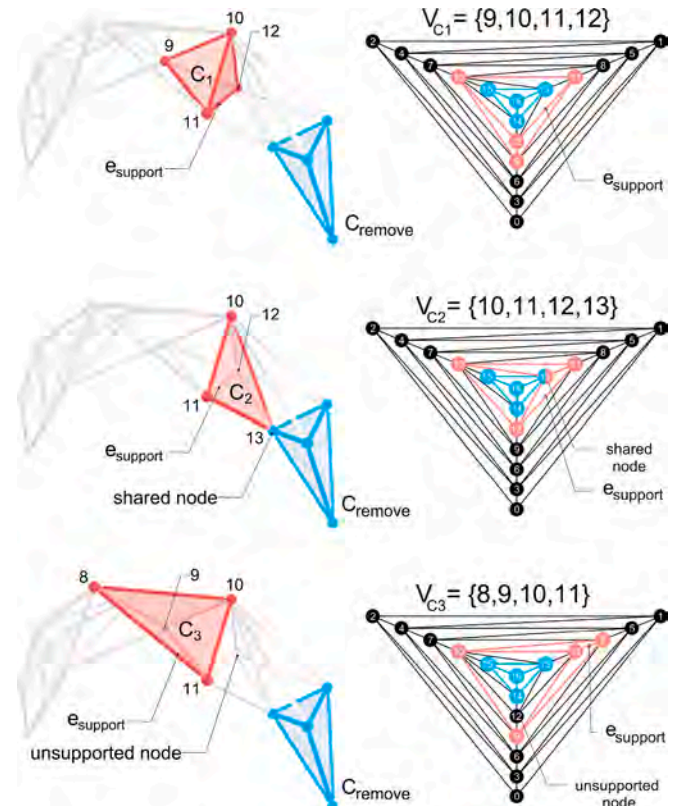


Fig. 15. Checking the viability of a rigid cell to support in relation to the location of the current cell being removed. C_1 is a valid cell to support, while C_2 and C_3 violate the adjacency criteria.

4.5 as a case study. The complete disassembly of the structure is found to only require four phases. Each phase starts with the repositioning of a robot to support a viable rigid cell, $C_{support}$, located as per Section 5.2 and 5.3, and ends with the removal of a set of intermediate members, E_{remove} as per Section 5.4, followed by the removal of C_{remove} . Table 1 summarizes the key information computed for each phase, where G' is the graph of the remaining structure at the start of each disassembly phase. G'_{end} shows the results of the algorithm operating on G' , with C_{remove} in blue, $C_{support}$ in red, and E_{remove} in green. The vertex and edge sets for each of these components are listed in the table for each phase. The process starts by choosing C_{remove} as the rigid cell at one end of the arch, and progresses over the whole structure, with $C_{support} \rightarrow C_{remove}$ in the following phase.

6. Assembly and disassembly results and discussion

6.1. Space frame assembly

The 17-noded space frame structure, designed for the target

$3.0m \times 1.6m$ arch geometry in Section 4.5, is physically assembled in the Embodied Computation Lab at Princeton University. This is done using a robotic cell comprised of two ABB IRB 4600-40/2.55 robots on 3.9 m tracks. The purpose of this exercise is to validate the design method presented in this paper and to test the rigidity-preserving cooperative robotic fabrication sequence that the structure is designed for. The material system used for the structure is 85 mm diameter 3D-printed PETG hollow nodes (with embedded slots for the member connection tabs) and 30 mm square wood members, which are fastened manually with 6 mm machine screws drilled through a 3D-printed tapered PETG tab inserted through a slot at each member's end. This tab acts as the connector between the member and the node, and is itself manually placed after the member is maneuvered to the correct location with the robotic arm, which is instrumented with a pneumatic gripper (Fig. 16).

Snapshots of the structure, and its corresponding stability graph, as examples of the assembly process are shown in Fig. 17 during the execution of a 3-member aggregation sequence for nodes 7 and 11. Fig. 17a, and 17b show the structure during a partially completed

Table 1
The four stability-preserving disassembly phases planned for the space frame arch structure designed in Section 4.5.

	G'	C_{remove}	$C_{support}$	E_{remove}	G'_{end}
Phase 1		$V = \{13, 14, 15, 16\}$ $E = \{(13, 14), (13, 15), (13, 16), (14, 15), (14, 16), (15, 16)\}$	$V = \{9, 10, 11, 12\}$ $E = \{(9, 10), (9, 11), (9, 12), (10, 11), (10, 12), (11, 12)\}$	$E = \{(10, 13), (10, 14), (10, 15), (11, 13), (12, 13), (12, 14)\}$	
Phase 2		$V = \{9, 10, 11, 12\}$ $E = \{(9, 10), (9, 11), (9, 12), (10, 11), (10, 12), (11, 12)\}$	$V = \{5, 6, 7, 8\}$ $E = \{(5, 6), (5, 7), (5, 8), (6, 7), (6, 8), (7, 8)\}$	$E = \{(6, 9), (7, 10), (7, 9), (8, 9), (8, 10), (8, 11)\}$	
Phase 3		$V = \{5, 6, 7, 8\}$ $E = \{(5, 6), (5, 7), (5, 8), (6, 7), (6, 8), (7, 8)\}$	$V = \{1, 2, 3, 4\}$ $E = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$	$E = \{(1, 5), (3, 6), (4, 5), (4, 6), (4, 7), (5, 6)\}$	
Phase 4		$V = \{1, 2, 3, 4\}$ $E = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$	–	$E = \{(0, 1), (0, 2), (0, 3)\}$	

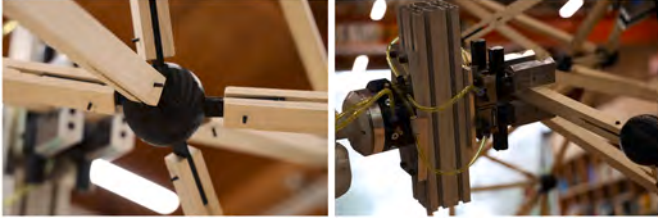
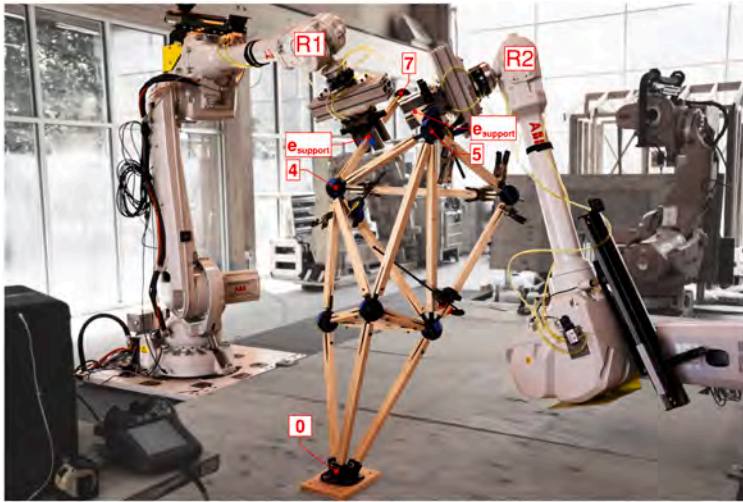


Fig. 16. A typical structural node with connecting tabs (left). A robotic arm with pneumatic gripper placing and supporting a member in the structure (right).

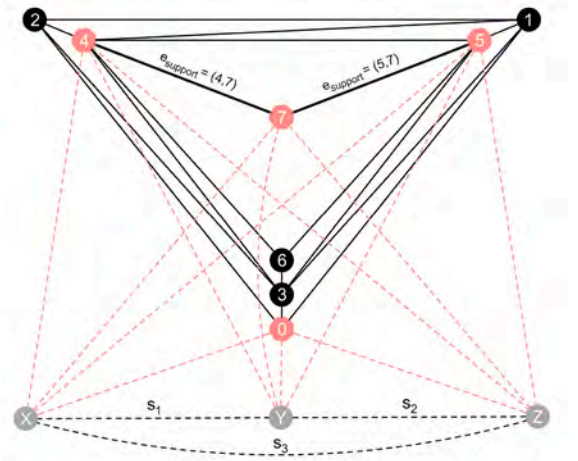
Henneberg sequence for node 7, where only two of the three members have been placed, and the structure is supported by both robots: $e_{\text{support}} = \{(4, 7), (5, 7)\}$. Fig. 17c and 17d show the stage where node 11 is fully connected and the structure itself becomes minimally rigid (i.e., at the end of a Henneberg 3-member sequence), with support from one robot only required for overall stability: $e_{\text{support}} = \{(10, 11)\}$. The full

fabrication sequence is shown in Appendix A, Table A.2, which summarizes the order of assembly, the edge in the graph each member corresponds to, and the robot (R1 or R2) that was used to place a member as part of a 3-member sequence. The finished structure is shown in Fig. 18.

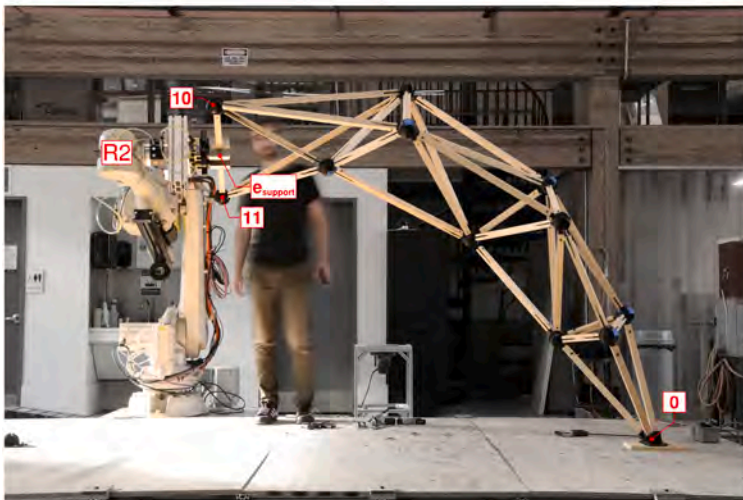
The robotic fabrication sequence for the full structure is planned using the COMPAS FAB framework [87] to calculate feasible collision-free paths for the robots to take while placing members. For a geometrically complex structure, individual members can be more easily maneuvered in space by the robots, which improves the likelihood of a feasible collision-free path to be found. But even so, as assembly progresses and the structure grows in size, it becomes more difficult to find feasible collision-free paths. To help with the path planning process in later stages of assembly, the calculation of trajectories is split into two discrete planning operations: (1) from the fixed pick-up location to an unobstructed intermediate plane in space; (2) from the intermediate plane to the final location of the member, also expressed as a plane located at the member's geometric center. The intermediate plane is located approximately 500 mm away from the structure and is parallel



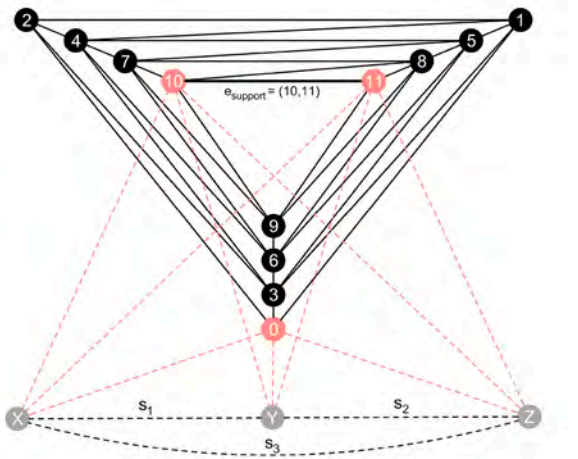
(a) both robots supporting the structure in a partially completed Henneberg Type 1 step for node 7.



(b) rigid stability graph of the structure showing the 4 supported vertices in red.



(c) a single robot supporting the structure after a completed Henneberg Type 1 step for node 11.



(d) rigid stability graph of the structure showing the 3 supported vertices in red.

Fig. 17. Snapshots of the cooperative robotic assembly process for the space frame arch structure designed as a case study. Robotic support is shown on the structure and its isomorphic graph for a partially completed (top) and completed (bottom) 3-member Henneberg sequence.

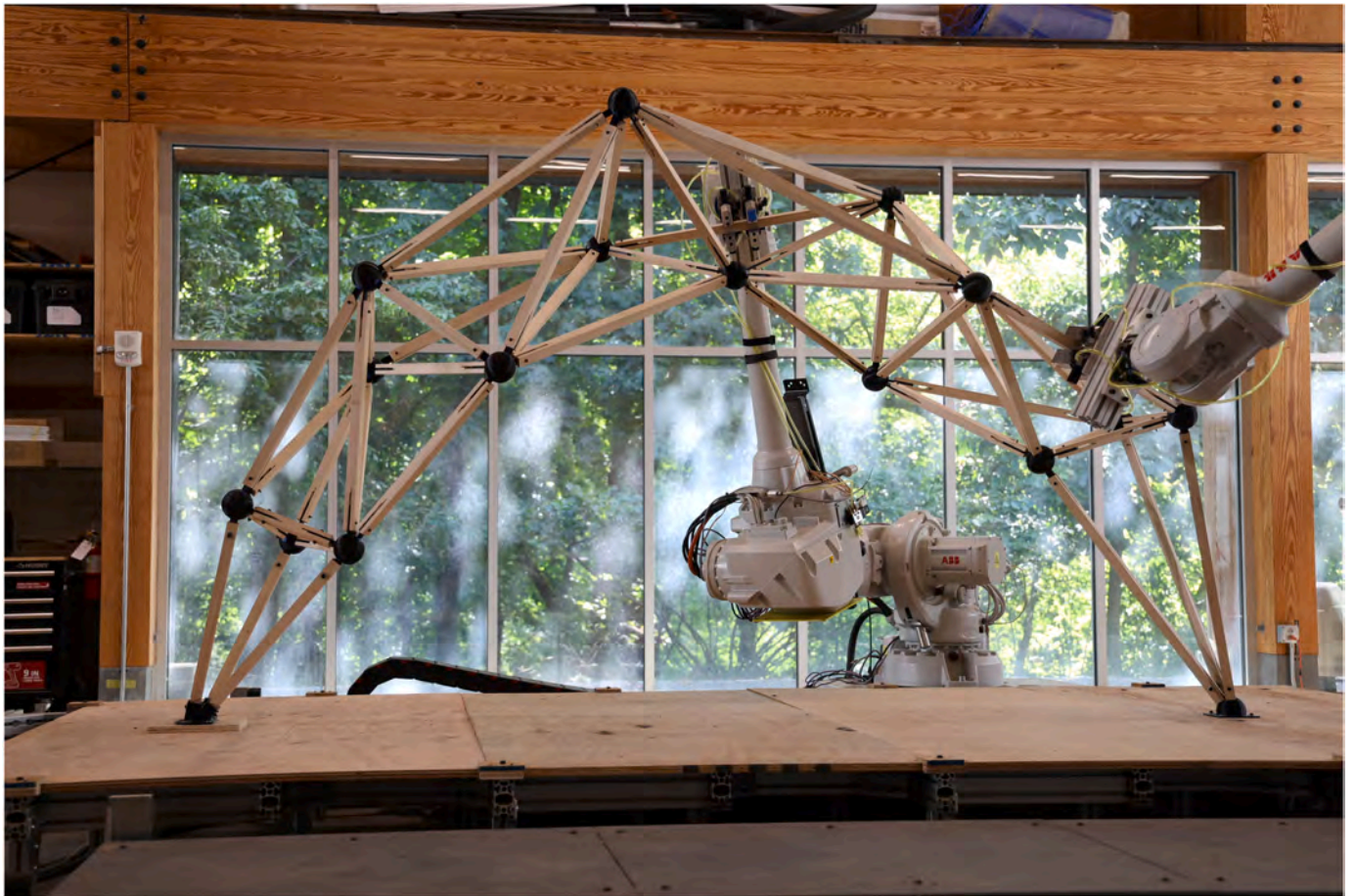


Fig. 18. The 17-noded space frame arch structure as the realization of a planar graph designed to have a rigidity-preserving (dis)assembly sequence.

Table 2

The two members supported by either robot resulting in the minimum average axial strain energy ($10^{-9} \text{ kN} \cdot \text{m}$) in each disassembly phase.

	R_1	R_2	$E_{e,avg}$
Phase 1	$C_{support}$ member = 31 $e = (11, 12)$	C_{remove} member = 41 $e = (14, 15)$	9.51
Phase 2	$C_{support}$ member = 15 $e = (5, 7)$	C_{remove} member = 31 $e = (11, 12)$	4.42
Phase 3	$C_{support}$ member = 6 $e = (1, 4)$	C_{remove} member = 16 $e = (6, 7)$	1.55
Phase 4	$C_{support}$ member = 6 $e = (1, 4)$	C_{remove} –	0.87

to the member's final location plane, which is based on the Cartesian position and orientation taken from an accurate 3D model of the structure. The likelihood of a feasible path being calculated is improved as the first operation would maneuver the robot in unobstructed space to a position in line with the member's final location, and the second operation would perform a relatively simple linear Cartesian move to the final location. The control of the robots, and execution of the calculated trajectories for the placement of each member, is then done using the COMPAS RRC framework [88]. Placement errors (i.e., positional differences between the digital model and as-built structure) ranged from 1–10 mm between the end of members and connection slots in the nodes. This was within the tolerance of the material system, therefore no correction using sensors or 3D imaging (as in [89]) was required during construction.

The rigidity-preserving assembly logic, and the resultant structure designed using the method developed in this paper, is specifically based around the use of two robotic arms. Adding a robotic support agent to the setup would greatly expand the design space for rigidity-preserving space frame structures, by allowing more complex cooperative robotic sequences to be performed. For example, it would be possible to construct rigid cells composed of five or six vertices, rather than the four as in the tetrahedral cell. These rigid cells can be realized through more complex rigidity-preserving fabrication sequences based on combinations of both Henneberg Type 1 and 2 steps. The ability to build cells with six vertices is particularly promising as there are four different rigid topological configurations that are possible, which would thus allow for additional flexibility during the design stage. Finally, a third robotic agent would allow for a more complex load-path, allowing for stable branching structures to be built beyond the linear partial order that is currently possible with two robots.

6.2. Space frame disassembly

The disassembly sequence computed in Section 5.5 is implemented on the completed structure at the end of the assembly sequence (Fig. 18). The results of the physical implementation of this rigidity-preserving sequence are documented for each of the four phases in Table 3. The

left column shows the disconnected structure, with one robot supporting the rigid tetrahedral cell to remove (C_{remove}) and the other robot supporting the remaining rigid portion of the arch structure. The right column shows close-up images of the rigid cell and the six intermediate members removed (E_{remove}) in the process of disconnecting the cell from the rest of the structure. The computation of the disassembly sequence does not provide information about which of the six elements comprising each selected cell a robot should support. While this choice does not impact rigidity (i.e., any of the six elements are valid), a finite element analysis was performed to inform which robot should support which cell in each phase, and the optimal combination of supported elements in each of the two cells (C_{remove} and $C_{support}$). The combination of supported elements that result in the minimum average strain energy during the full disassembly process in each phase are summarized in Table 2.

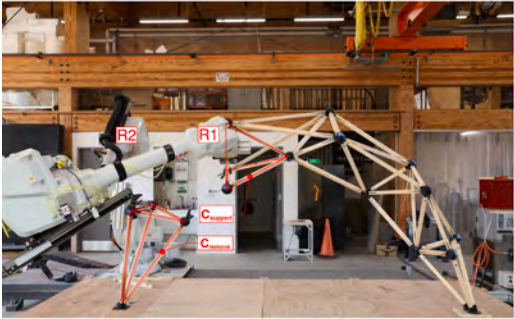
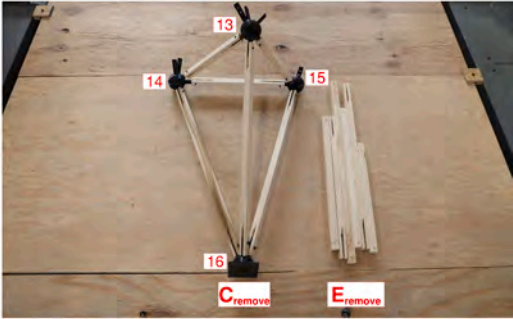
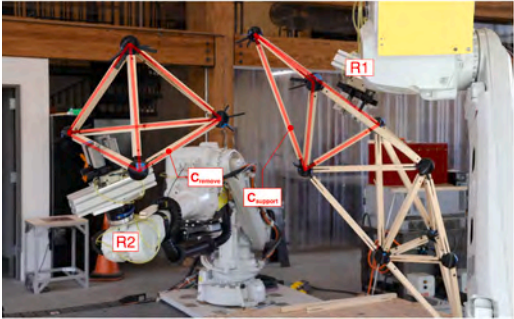
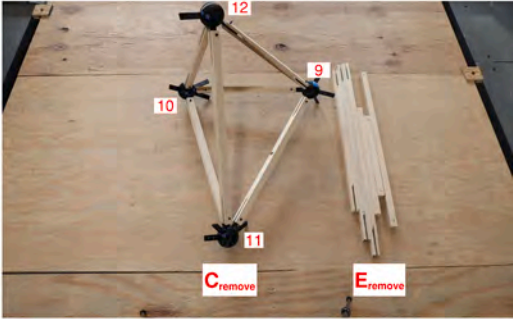
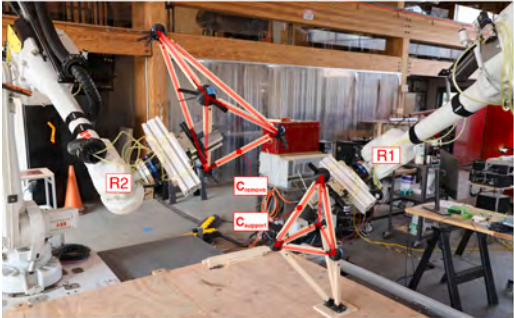
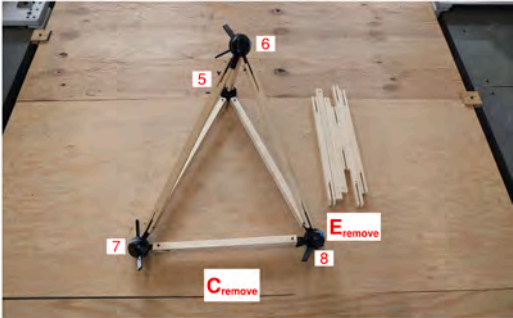
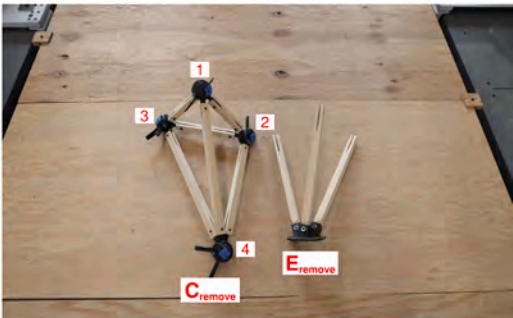
The methodology for this analysis and the full data for all possible support combinations in each phase are shown in Appendix B. Robotic reachability also impacts which members can be supported, but is not an issue in this specific case as most members are within reach of at least one robot, which is verified using the path-planning functionality in COMPAS FAB [87]. In general, supporting members that are located close to the central axis of the arch results in lower average axial strain energy in the structure, for all steps in a particular disassembly phase. This result is due to the structure experiencing less out-of-plane twisting when supported at these central locations.

Using a closed-form algorithm to locate and isolate rigid cells in a structure during disassembly, such as the one presented in Section 5.2, is a valid approach when given a structure with a uniform topology. For example, as in the case study presented in this paper, where the structure is known to be composed of tetrahedral cells. But a more generic method is necessary when expanding the feasible design space to include structures composed of both more complex and varied rigid cells, and branching load paths (i.e., not a linear partial order of rigid cells). With the future inclusion of a third robotic agent in the fabrication setup, assembling such structures will be possible. The calculation of a feasible rigidity-preserving disassembly sequence must thus be generalized for use in such structures with unknown and varied topologies by developing a more generic approach to the cell locating and isolating algorithm. This will be accomplished in future work through the implementation of a graph-search approach based on the rigidity-finding *pebble game* algorithm [77].

7. Conclusion

In this paper, a novel graph theoretic approach, based on rigidity theory, is presented as a way to assess the stability of spatially complex bar and joint structures (i.e., trusses and space frames) during any phase of their construction. A topological framework, referred to as a *stability graph*, is developed as an isomorphic representation of the physical structure (i.e., there exists a one-to-one mapping between the graph and structure). The structural elements, support conditions, and temporary robotic supports can all be represented in the *stability graph*, and their collective effect on the stability of the structure can thus be evaluated using only graph rigidity principles (e.g., Laman count and Henneberg construction steps). A schematic design method, based on a topology-driven approach to rigidity and structural stability, is then developed and used to design space frame structures that remain stable during all

Table 3
Documentation of the four phases of the rigidity-preserving disassembly sequence calculated and executed for the space frame arch structure.

	Disconnected Supported Structure	C_{remove} and E_{remove}
Phase 1		
Phase 2		
Phase 3		
Phase 4		

stages of their assembly and disassembly without requiring external scaffolding. The design method is *fabrication-informed*, because it is specifically developed to take advantage of the fabrication capabilities (with respect to sequencing) when two industrial robotic arms work together in a cooperative manner: placing elements (during assembly), removing cells (during disassembly), while also supporting the structure in its temporary state. The proposed method is demonstrated in this paper through the case study design, and robotic assembly and disassembly, of a rigidity-preserving space frame arch, composed of 17 nodes and 45 elements. The structure is sequentially designed to be assembled element-by-element with two robots, mirroring a Henneberg rigid-graph assembly step in the topological design space of the *stability graph*. The design process and assembly logic guarantee that the property of planarity is maintained in the graph, which physically corresponds to a structure that is composed of a series of rigid tetrahedral cells. The property of planarity, coupled with a tetrahedral cell topology, results in a structure that can then also be efficiently disassembled through the application of a graph-based algorithm to sequentially locate and isolate rigid cells in the graph. After the cells are located, they are robotically supported and removed from the physical structure as independent components, all while the stability of the remaining structure is preserved.

As robotic technology improves, its applications in the AEC industry will continue to grow, with multi-robot setups becoming more prevalent. Engineers and architects should consider how the structures they design can better take advantage of the potential afforded them through such robotic fabrication setups. The generic support/place sequencing approach, that is demonstrated with two robots in this paper, can help to

further expand the feasible design space for geometrically complex discrete element structures fabricated on-site or pre-fabricated in pieces off-site. Future work will integrate feedback in the form of 3D imaging of the structure during assembly or disassembly to allow the fabrication process to better adapt to changing conditions such as deformation of the structure due to self-weight. Overall, the goal is to further promote the application of robotic fabrication and automation in construction, while simultaneously improving the material use and construction efficiency of geometrically complex trusses and space frames.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The research presented in this paper was supported by the National Science Foundation (NSF) under Grant No. 2122271. The authors would like to further acknowledge the Natural Sciences and Engineering Research Council (NSERC) of Canada (funding reference # 532482), and Princeton University's ROBELARCH Global Collaborative Research Network for their financial support. In addition, we would like to thank the ACADIA conference (Realignments: toward critical computation) for hosting our workshop, Remote Robotic Assemblies, during which the space frame arch structure discussed in this paper was assembled.

Appendix A. Space frame arch design and assembly

This appendix presents additional information for the space frame arch structure that was designed in Section 4.5 and the cooperative robotic sequence that was used to assemble the structure as discussed in Section 6.1. Table A.1 summarizes the results from each iteration of the design process, showing the new edges connecting each new vertex to the existing graph and the calculated mapping of this vertex to $\mathbb{R}^3$. Table A.2 summarizes the cooperative robotic fabrication sequence that was used to build the resulting space frame arch. This table shows the member #, its corresponding edges in the graph representation, and the specific robot (either R1 or R2) that was used to place the member. Each new node is added through one of the following feasible 3-member placement sequences using both robots: R1-R2-R1, R1-R2-R2, R2-R1-R2, or R2-R1-R1. Note that the first 6 members comprise a complete tetrahedral that was added manually as a base to initialize the construction, and are therefore not associated with a robotic placement.

Table A.1

Vertex mapping to $\mathbb{R}^3$ and graph connectivity at each aggregation phase (G') following an iteration of the design process.

v	Node $n : v \rightarrow \mathbb{R}^3$	Added edges $(u, v) \in G'(E)$
0	(− 0.639, 1.730, 0.000)	-
1	(− 0.529, 1.646, 0.670)	{{(0, 1)}}
2	(− 0.385, 1.927, 0.576)	{{(0, 2), (1, 2)}}
3	(− 0.211, 1.558, 0.546)	{{(0, 3), (1, 3), (2, 3)}}
4	(− 0.177, 2.015, 1.117)	{{(1, 4), (2, 4), (3, 4)}}
5	(− 0.188, 1.568, 1.307)	{{(1, 5), (3, 5), (4, 5)}}
6	(0.181, 1.390, 1.058)	{{(3, 6), (4, 6), (5, 6)}}
7	(0.512, 2.065, 1.539)	{{(4, 7), (5, 7), (6, 7)}}
8	(0.512, 1.413, 1.793)	{{(5, 8), (6, 8), (7, 8)}}
9	(0.914, 1.754, 1.394)	{{(6, 9), (7, 9), (8, 9)}}
10	(1.431, 1.885, 1.655)	{{(7, 10), (8, 10), (9, 10)}}
11	(1.487, 1.237, 1.274)	{{(8, 11), (9, 11), (10, 11)}}
12	(1.431, 2.050, 1.107)	{{(9, 12), (10, 12), (11, 12)}}
13	(1.854, 1.710, 0.812)	{{(10, 13), (11, 13), (12, 13)}}
14	(2.211, 2.024, 0.957)	{{(10, 14), (12, 14), (13, 14)}}
15	(2.189, 1.444, 0.925)	{{(10, 15), (13, 15), (14, 15)}}
16	(2.397, 1.725, 0.000)	{{(13, 16), (14, 16), (15, 16)}}

Table A.2
Cooperative robotic assembly sequence for the space frame arch.

Member #	Edge (u, v)	Robot #
0	(0, 1)	-
1	(0, 2)	-
2	(0, 3)	-
3	(2, 3)	-
4	(1, 3)	-
5	(1, 2)	-
6	(1, 4)	R2
7	(2, 4)	R1
8	(3, 4)	R2
9	(1, 5)	R2
10	(4, 5)	R1
11	(3, 5)	R2
12	(3, 6)	R2
13	(4, 6)	R1
14	(5, 6)	R2
15	(5, 7)	R2
16	(6, 7)	R1
17	(4, 7)	R1
18	(6, 8)	R2
19	(7, 8)	R1
20	(5, 8)	R1
21	(7, 9)	R1
22	(6, 9)	R2
23	(8, 9)	R2
24	(9, 10)	R2
25	(7, 10)	R1
26	(8, 10)	R2
27	(9, 11)	R1
28	(8, 11)	R2
29	(10, 11)	R2
30	(10, 12)	R1
31	(11, 12)	R2
32	(9, 12)	R1
33	(12, 13)	R1
34	(10, 13)	R2
35	(11, 13)	R2
36	(10, 14)	R1
37	(13, 14)	R2
38	(12, 14)	R1
39	(10, 15)	R2
40	(14, 15)	R1
41	(13, 15)	R2
42	(13, 16)	R2
43	(14, 16)	R1
44	(15, 16)	R2

Appendix B. Axial strain energy in disassembly

This appendix presents all the strain energy data that was generated in the process of determining what members to robotically support in each of the four disassembly phases for the space frame arch. The sequence of rigid cells to remove and support was planned in [Section 5.5](#) with results of the physical implementation of the disassembly sequence shown in [Section 6.2](#).

Each table summarizes the average axial strain energy ($10^{-9} \text{ kN} \cdot \text{m}$) per member in the structure, for each disassembly step in a specific disassembly phase, for every possible combination of supported members. There are 7 distinct disassembly steps in a typical phase (i.e., 7 entries per table cell). These 7 steps corresponds to: first supporting the structure with two robots, and then manually removing the 6 intermediate members, E_{remove} , one-by-one. The structure is analyzed, and the average strain energy is recorded after the simulation of each of these 7 steps.

There are 36 unique combinations of members for the two robots (R1 and R2) to support in each phase because there are 6 potential members to support in either of the chosen rigid cells, C_{remove} and C_{support} . The horizontal and vertical axes of each table show the member #, and corresponding graph edge, representing a potential member to support in each cell. The combination of support members representing the lowest average axial strain over all 7 steps is highlighted in the table for each phase. These members were then chosen to be supported in the physical implementation of the disassembly sequence shown in [Section 6.2](#). The final phase only requires one robot supporting the structure and the removal of 3 members connecting the final cell to the foundation, hence there are only 4 steps and 6 unique support combinations possible.

Table B.1

Average axial strain energy ($10^{-9} \text{ kN} \cdot \text{m}$) in disassembly phase 1.

R1 \ R2	24 (9, 10)	27 (9, 11)	29 (10, 11)	30 (10, 12)	31 (11, 12)	32 (9, 12)
39 (13,15)	7.89 8.05 8.23 8.36 8.54 8.65 38.64	8.40 8.48 9.91 10.57 11.89 12.19 12.50	9.20 9.55 9.77 10.16 11.63 11.93 12.23	9.41 9.62 9.84 10.03 10.27 11.55 22.81	8.98 9.15 9.33 9.50 9.73 9.98 10.23	9.46 9.54 11.80 12.03 12.32 12.68 16.01
40 (13,14)	7.86 8.04 8.22 8.40 8.58 8.63 38.61	8.09 8.35 9.47 10.57 11.89 12.16 12.47	9.09 9.30 9.52 10.17 11.64 11.90 12.21	9.74 9.96 10.19 10.43 10.69 11.53 22.78	8.94 9.10 9.29 9.51 9.74 9.95 10.21	9.30 9.48 11.88 12.16 12.46 12.65 15.98
41 (14,15)	8.06 8.24 8.54 8.70 8.98 9.29 38.60	8.70 9.55 10.16 11.06 11.84 12.13 12.46	9.11 9.32 9.59 10.56 11.58 11.87 12.19	9.53 9.74 10.04 10.27 10.56 13.21 22.77	8.95 9.10 9.23 9.45 9.68 9.93 10.20	11.34 11.80 13.44 13.76 14.27 13.95 15.97
42 (13,16)	8.08 8.24 8.43 8.56 8.74 8.80 38.78	8.60 8.64 10.26 10.75 12.08 12.33 12.65	9.49 9.84 10.07 10.35 11.82 12.07 12.38	10.13 10.40 10.64 11.13 11.40 11.70 22.95	9.12 9.28 9.47 9.69 9.92 10.12 10.38	9.57 9.67 12.18 13.08 13.40 12.82 16.16
43 (15,16)	8.31 8.46 8.79 8.99 9.25 9.62 38.74	10.14 11.62 12.93 12.84 12.77 13.09 12.60	9.44 9.77 10.08 12.21 12.52 12.83 12.34	9.77 9.97 10.28 10.49 10.97 15.47 22.91	9.39 9.46 9.59 9.98 10.62 10.88 10.34	11.89 12.01 13.68 14.85 15.29 14.50 16.11
44 (14,16)	8.66 8.86 9.14 9.36 9.55 65.77 38.75	8.86 9.71 10.35 11.22 12.16 12.49 12.61	9.34 9.55 9.81 10.72 11.90 12.23 12.35	10.23 10.46 10.81 11.07 11.38 33.63 22.92	9.29 9.44 9.56 9.79 10.00 10.29 10.35	12.74 13.36 15.78 16.15 16.99 22.28 16.12

Table B.2Average axial strain energy ($10^{-9} \text{ kN} \cdot \text{m}$) in disassembly phase 2.

R2 \ R1	14	15	16	18	19	20
	(5, 6)	(5, 7)	(6, 7)	(6, 8)	(7, 8)	(5, 8)
24 (9,10)	4.68	5.46	6.63	5.70	7.67	3.85
	3.30	2.91	4.93	4.87	6.89	2.95
	3.48	2.86	5.04	5.02	7.11	3.05
	3.31	2.96	5.21	5.25	7.35	3.00
	7.99	3.99	5.33	5.43	7.60	3.11
	7.95	4.14	5.52	37.53	7.88	5.70
	8.25	4.22	5.73	38.92	10.07	13.09
27 (9,11)	9.86	7.22	8.94	8.25	10.32	6.57
	10.17	8.38	9.73	8.51	10.65	6.78
	219.87	184.75	186.93	256.24	189.00	219.19
	2.84	2.49	4.73	4.78	6.88	2.53
	7.50	3.51	4.85	4.94	7.12	2.62
	7.45	3.63	5.02	37.03	7.37	5.19
	7.72	3.70	5.20	38.40	9.55	12.56
29 (10,11)	5.68	3.37	5.20	4.98	6.59	3.48
	5.76	4.69	5.57	5.14	6.80	3.58
	18.24	6.88	6.25	5.31	7.02	3.70
	17.05	7.11	6.45	6.02	7.25	3.68
	23.99	10.93	11.93	34.71	20.78	22.16
	24.69	25.90	22.26	54.27	57.00	31.22
	8.02	3.99	5.50	38.70	9.84	12.86
30 (10,12)	4.75	4.32	5.81	5.43	7.13	3.98
	5.09	4.05	5.57	6.13	7.18	5.60
	5.94	4.15	5.62	6.33	7.41	5.78
	5.85	4.29	5.81	23.18	7.66	15.27
	9.21	4.69	5.88	24.19	8.10	15.50
	8.93	6.27	6.51	38.52	8.36	23.91
	8.69	4.66	6.17	39.37	10.51	13.53
31 (11,12)	4.39	3.29	5.10	5.01	6.46	3.72
	4.47	3.86	5.23	5.17	6.66	3.84
	5.45	4.10	5.25	5.31	6.86	3.96
	5.31	5.16	5.57	6.03	7.12	4.98
	8.41	5.28	5.60	6.51	7.50	5.77
	8.14	5.33	5.71	37.72	7.78	22.96
	7.97	3.94	5.45	38.65	9.79	12.81
32 (9,12)	8.70	8.02	9.80	12.68	11.07	8.24
	7.84	6.95	9.25	12.41	10.80	7.83
	8.59	7.35	9.53	13.34	11.60	8.57
	3.09	2.74	4.98	5.03	7.13	2.78
	7.76	3.76	5.10	5.20	7.37	2.88
	7.71	3.90	5.29	37.30	7.64	5.46
	8.00	3.97	5.48	38.68	9.82	12.84

Table B.3Average axial strain energy ($10^{-9} \text{ kN} \cdot \text{m}$) in disassembly phase 3.

R2 \ R1	3	4	5	6	7	8
	(2, 3)	(1, 3)	(1, 2)	(1, 4)	(2, 4)	(3, 4)
14 (5,6)	7.00	7.11	6.63	5.68	6.29	5.92
	1.73	1.92	1.84	1.62	2.27	1.88
	1.84	2.03	2.22	1.71	2.39	1.98
	2.24	3.39	2.48	1.80	2.52	2.09
	2.37	3.59	2.69	1.90	2.56	2.21
	2.50	3.81	2.85	2.02	2.64	2.48
	2.67	4.06	3.38	2.22	15.01	2.64
15 (5,7)	2.53	2.73	3.54	3.33	3.38	2.65
	2.61	2.79	3.65	3.50	3.55	2.79
	2.75	2.94	5.02	3.94	3.85	2.89
	3.19	4.34	5.26	4.16	4.06	3.05
	1.53	2.75	1.85	1.07	1.73	1.38
	1.62	2.92	1.96	1.13	1.75	1.59
	1.72	3.11	2.43	1.27	14.06	1.69
16 (6,7)	1.66	2.04	1.98	1.27	2.45	1.52
	1.66	2.00	2.04	1.33	2.57	1.59
	1.65	1.96	2.25	1.40	2.71	1.67
	1.79	2.93	2.27	1.48	3.00	1.79
	1.90	3.10	2.59	1.61	2.99	1.90
	1.97	3.28	3.30	1.90	23.31	1.95
	2.32	3.71	3.02	1.87	14.66	2.29
18 (6,8)	4.56	5.36	5.01	2.11	3.30	2.35
	4.56	6.13	4.97	1.90	3.13	2.14
	4.75	7.71	4.97	2.00	3.29	2.25
	16.96	17.96	22.42	20.30	34.22	17.15
	17.96	19.01	26.97	23.57	35.91	18.15
	18.79	20.09	33.16	27.80	142.06	18.76
	2.34	3.73	3.05	1.89	14.68	2.31
19 (7,8)	1.87	2.00	3.46	2.59	5.81	1.92
	1.93	2.03	3.80	2.72	6.10	2.01
	1.82	1.89	4.09	2.72	7.40	2.11
	2.13	3.25	4.23	2.87	8.03	2.17
	2.12	3.30	4.69	2.96	8.13	2.17
	2.13	3.44	5.72	3.48	60.95	2.11
	1.35	2.74	2.05	0.90	13.69	1.32
20 (5,8)	6.97	7.24	8.60	4.34	4.68	4.25
	7.64	7.56	9.92	3.70	4.08	3.64
	27.25	27.44	29.36	30.86	33.28	27.39
	29.05	30.21	31.61	32.58	35.13	28.91
	2.42	3.64	2.74	1.96	2.61	2.26
	2.56	3.86	2.91	2.08	2.70	2.53
	2.73	4.12	3.43	2.28	15.07	2.70

Table B.4Average axial strain energy ($10^{-9} \text{ kN} \cdot \text{m}$) in disassembly phase 4.

- \ R1	3	4	5	6	7	8
	(2, 3)	(1, 3)	(1, 2)	(1, 4)	(2, 4)	(3, 4)
-	1.31	3.73	2.59	0.66	21.98	1.36
	1.57	4.20	2.91	0.74	31.44	39.18
	1.79	0.88	3.32	0.93	35.93	0.96
	2.09	1.02	1.49	1.13	1.88	1.12

References

- [1] S. Sharif, R. Gentry, BIM for masonry: development of bim plugins for the masonry unit database, in: *Proceedings of the 33rd Education and Research in Computer Aided Architectural Design in Europe Conference*, 2015, pp. 567–576. <http://paper.s.cumincad.org/cgi-bin/works/paper/ecaade2015.261>.
- [2] C. Meyer, M.H. Sheer, Concrete shells deserve another look? *Concr. Int.* 27 (10) (2005) 43–50. <https://www.concrete.org/publications/internationalconcreteabstractsportal.aspx?m=details&ID=14717>.
- [3] R. Zimmann, H. O'Brien, J. Hargrave, M. Morrell, The Circular Economy in the Built Environment, Tech. rep., Arup, Arup, London, 2016. <https://www.arup.com/perspectives/publications/research/section/circular-economy-in-the-built-environment>.
- [4] S. Kaethner, J. Burrige, Embodied CO₂ of structural frames, *Struct. Eng.* 90 (5) (2012) 33–40. <https://www.istructe.org/sitefiles/handlers/DownloadFile.ashx?productid=1583>.
- [5] F. Barbosa, J. Woetzel, J. Mischke, M. Ribeiro, M. Sridhar, M. Parsons, N. Bertram, S. Brown, Reinventing Construction: A Route to Higher Productivity, Tech. rep., McKinsey Global Institute, 2017. <https://www.mckinsey.com/business-functions/operations/our-insights/reinventing-construction-through-a-productivity-revolution>.
- [6] W. Lu, H. Yuan, A framework for understanding waste management studies in construction, *Waste Manage.* N.Y. 31 (6) (2011) 1252–1260. <https://doi.org/10.1016/j.wasman.2011.01.018>.
- [7] J. Brütting, J. Desruelle, G. Senatore, C. Fivet, Design of truss structures through reuse, *Structures* 18 (2019) 128–137. <https://doi.org/10.1016/j.istruc.2018.11.006>.
- [8] J. Brütting, G. Senatore, M. Schevenels, C. Fivet, Optimum design of frame structures from a stock of reclaimed elements, *Front. Built Environ.* 6 (2020) 57. <https://doi.org/10.3389/fbuil.2020.00057>.
- [9] J. Brütting, G. Senatore, C. Fivet, Design and fabrication of a reusable kit of parts for diverse structures, *Automat. Constr.* 125 (2021) 103614. <https://doi.org/10.1016/j.autcon.2021.103614>.
- [10] G. Bravo-Palacios, A.D. Prete, P.M. Wensing, One robot for many tasks: Versatile co-design through stochastic programming, *IEEE Robot. Automat. Lett.* 5 (2) (2020) 1680–1687. <https://doi.org/10.1109/LRA.2020.2969948>.
- [11] P. Eversmann, F. Gramazio, M. Kohler, Robotic prefabrication of timber structures: towards automated large-scale spatial assembly, *Constr. Robot.* 1 (2017) 49–60. <https://doi.org/10.1007/s41693-017-0006-2>.
- [12] IFR, World Robotics Report, 2018. <https://ifr.org/ifr-press-releases/news/global-industrial-robot-sales-doubled-over-the-past-five-years>.
- [13] B. García de Soto, I. Agustí-Juan, J. Hunhevicz, S. Joss, K. Graser, G. Habert, B. T. Adey, Productivity of digital fabrication in construction: cost and time analysis of a robotically built wall, *Automat. Constr.* 92 (2018) 297–311. <https://doi.org/10.1016/j.autcon.2018.04.004>.
- [14] V.R.P.M.B. Kumar, S.J. Raj, Robotics in construction industry, *Indian J. Sci. Technol.* 9 (23) (2016) 1–12. <https://doi.org/10.17485/ijst/2016/v9i23/95974>.
- [15] S. Parascho, Cooperative robotic assembly: Computational Design and Robotic Fabrication of Spatial Metal Structures, PhD Thesis, ETH Zurich, 2019. <https://doi.org/10.3929/ethz-b-000364322>.
- [16] T. Bock, Construction robotics, *Automat. Robot.* 22 (3) (2007) 201–209. <https://doi.org/10.1007/s10514-006-9008-5>.
- [17] T. Bock, The future of construction automation: technological disruption and the upcoming ubiquity of robotics, *Automat. Constr.* 59 (2015) 113–121. <https://doi.org/10.1016/j.autcon.2015.07.022>.
- [18] J.S. Albus, Trip report: Japanese progress in robotics for construction, *Robotics* 2 (2) (1986) 103–112. <https://doi.org/10.1016/0167-84938690047-1>, publisher: North-Holland.
- [19] L. Cousineau, N. Miura, *Construction robots: the search for new building technology in Japan*, Am. Soc. Civil Eng. Reston VA (1998). ISBN: 978-0-7844-0317-4.
- [20] P.Y. Huang, M. Sakurai, Factor automation: the Japanese experience, *IEEE Trans. Eng. Manage.* 37 (2) (1990) 102–108. <https://doi.org/10.1109/17.53712>.
- [21] T. Ueno, J. Maeda, T. Yoshida, S. Suzuki, Construction Robots for Site automation, in: *Proceedings of the 3rd ISARC*, Marseille, France, 1986, pp. 627–639. <https://doi.org/10.22260/ISARC1986/0049>.
- [22] T. Bock, T. Linner, *Site automation: Automated/robotic on-site factories 3*, Cambridge University Press, Cambridge, 2016. ISBN: 978-1-107-07597-9.
- [23] M.J. Skibniewski, *Robotics in civil engineering*, Van Nostrand Reinhold, Southampton, 1989. ISBN: 978-0-442-31925-0.
- [24] J.M. Davila Delgado, L. Oyedele, A. Ajayi, L. Akanbi, O. Akinade, M. Bilal, H. Owolabi, Robotics and automated systems in construction: understanding industry-specific challenges for adoption, *J. Build. Eng.* 26 (2019) 1–11. <https://doi.org/10.1016/j.job.2019.100868>.
- [25] F. Gramazio, M. Kohler, in: *Digital Materiality in Architecture*, 1st Edition, Lars Muller, Baden, 2008. ISBN: 978-3-03778-122-7.
- [26] F. Gramazio, in: M. Kohler (Ed.), *Made by robots: Challenging Architecture at a Larger Scale*, 1st Edition, Academy Press, London, 2014. ISBN: 978-1-118-53548-6.
- [27] M. Kohler, F. Gramazio, J. Willmann, *The programmed column*, in: *The Robotic Touch: How Robots Change Architecture*, The University of Chicago Press, Chicago, 2014, pp. 208–215. ISBN: 978-3-906027-37-1.
- [28] T. Bonwetsch, D. Kobel, F. Gramazio, M. Kohler, The informed wall: applying additive digital fabrication techniques on architecture, in: *Proceedings of the 25th Annual Conference of the Association for Computer-Aided Design in Architecture*, 2006, pp. 489–495. ISBN: 978-0-9789463-0-2.
- [29] T. Bonwetsch, F. Gramazio, M. Kohler, Digitally fabricating non-standardised brick walls, in: *ManuBuild*, 2007, pp. 191–196. ISBN: 978-0-86017-710-4.
- [30] M. Kohler, F. Gramazio, J. Willmann, *Gantenbein vineyard façade*, in: *The Robotic Touch: How Robots Change Architecture*, The University of Chicago Press, Chicago, 2014, pp. 66–75. ISBN: 978-3-906027-37-1.
- [31] EMPA, DFAB HOUSE - Digital fabrication and living, 2021. <https://www.empa.ch/web/nest/digital-fabrication>.
- [32] J. Willmann, M. Knauss, T. Bonwetsch, A.A. Apolinarska, F. Gramazio, M. Kohler, Robotic timber construction – Expanding additive fabrication to new dimensions, *Automat. Constr.* 61 (2016) 16–23. <https://doi.org/10.1016/j.autcon.2015.09.011>.
- [33] N. Hack, T. Wangler, J. Mata-Falcón, K. Dörfler, N. Kumar, N. Walzer, K. Graser, L. Reiter, H. Richner, J. Buchli, W. Kaufmann, R.J. Flatt, F. Gramazio, M. Kohler, Mesh mould: an on site, robotically fabricated, functional formwork, in: *Second Concrete Innovation Conference*, Tromsø, Norway 19, 2017, p. 11. ISBN: 978-82-8208-054-5.
- [34] N. Hack, K. Dörfler, A.N. Walzer, T. Wangler, J. Mata-Falcón, N. Kumar, J. Buchli, W. Kaufmann, R.J. Flatt, F. Gramazio, M. Kohler, Structural stay-in-place formwork for robotic in situ fabrication of non-standard concrete structures: a real scale architectural demonstrator, *Automat. Constr.* 115 (2020) 1–13. <https://doi.org/10.1016/j.autcon.2020.103197>.
- [35] R. Bärtschi, M. Knauss, T. Bonwetsch, F. Gramazio, M. Kohler, Wiggled Brick bond, in: *Advances in Architectural Geometry 2010*, Springer, Wien/New York, 2010, pp. 137–147.
- [36] L. Piš Korec, D. Jenny, S. Parascho, H. Mayer, F. Gramazio, M. Kohler, The brick labyrinth, in: *Robotic Fabrication in Architecture, Art and Design 2018*, Springer International Publishing, Zurich, 2018, pp. 489–500. https://doi.org/10.1007/978-3-319-92294-2_37.
- [37] K. Dörfler, T. Sandy, M. Gifftaler, F. Gramazio, M. Kohler, J. Buchli, Mobile robotic brickwork: Automation of a discrete robotic fabrication process using an autonomous mobile robot, in: *Robotic Fabrication in Architecture, Art and Design 2016*, Springer International Publishing Sydney, Australia, 2016, pp. 204–217. https://doi.org/10.1007/978-3-319-26378-6_15.
- [38] M. Gifftaler, T. Sandy, K. Dörfler, I. Brooks, M. Buckingham, G. Rey, M. Kohler, F. Gramazio, J. Buchli, Mobile robotic fabrication at 1:1 scale: The in situ fabricator, *Constr. Robot.* 1 (1) (2017) 3–14. <https://doi.org/10.1007/s41693-017-0003-5>.
- [39] I.X. Han, F. Meggers, S. Parascho, Bridging the collectives: A review of collective human-robot construction, *Int. J. Architect. Comput.* 19 (4) (2021) 20. <https://doi.org/10.1177/14780771211025153>.
- [40] K.H. Petersen, N. Napp, R. Stuart-Smith, D. Rus, M. Kovac, A review of collective robotic construction, *Sci. Robot.* 4 (28) (2019). <https://doi.org/10.1126/scirobotics.aau8479>.
- [41] S. Bragança, E. Costa, I. Castellucci, P.M. Arezes, A Brief Overview of the Use of Collaborative Robots in Industry 4.0: Human Role and Safety, in: *Occupational and Environmental Safety and Health, Studies in Systems, Decision and Control*, Springer International Publishing, Cham, 2019, pp. 641–650. https://doi.org/10.1007/978-3-030-14730-3_68.
- [42] E.P.G. 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, *Automat. Constr.* 129 (9) (2021) 103803. <https://doi.org/10.1016/j.autcon.2021.103803>.
- [43] P. Carneau, R. Mesnil, N. Roussel, O. Baverel, Additive manufacturing of cantilever – From masonry to concrete 3D printing, *Automat. Constr.* 116 (2020) 1–16. <https://doi.org/10.1016/j.autcon.2020.103184>.
- [44] M. Motamedi, R. Oval, P. Carneau, O. Baverel, Supportless 3d printing of shells: adaptation of ancient vaulting techniques to digital fabrication, in: *Proceedings of the Design Modelling Symposium Berlin, Berlin*, 2019, pp. 714–726. https://doi.org/10.1007/978-3-030-29829-6_55.
- [45] P. Latteur, A. Fascetti, S. Goessens, The droxel: a universal parametric construction component, *Eng. Struct.* 250 (2022) 113363. <https://doi.org/10.1016/j.engstruct.2021.113363>.
- [46] S. Goessens, C. Mueller, P. Latteur, Feasibility study for drone-based masonry construction of real-scale structures, *Automat. Constr.* 94 (2018) 458–480. <https://doi.org/10.1016/j.autcon.2018.06.015>.
- [47] S. Parascho, A. Gandia, A. Mirjan, F. Gramazio, M. Kohler, in: A. Menges, B. Sheil, R. Glynn, M. Skavara (Eds.), *Cooperative fabrication of spatial metal structures*, Fabricate, UCL Press, London, 2017, pp. 24–29. <https://doi.org/10.3929/ethz-b-000219566>.
- [48] 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: *Advances in Architectural Geometry*, 2018, pp. 112–139. <https://research.chalmers.se/en/publication/504188>.
- [49] A. Thoma, A. Adel, M. Helmreich, T. Wehrle, F. Gramazio, M. Kohler, Robotic fabrication of bespoke timber frame modules, in: *Robotic Fabrication in Architecture, Art and Design 2018*, Springer International Publishing, Zurich, 2018, pp. 447–458. https://doi.org/10.1007/978-3-319-92294-2_34.
- [50] 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*, Mexico City, 2018, pp. 342–349. ISBN: 978-0-692-17729-7.
- [51] 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, *Constr. Robot.* 4 (3) (2020) 1–10. <https://doi.org/10.1007/s41693-020-00041-w>.
- [52] S. Parascho, I.X. Han, A. Beghini, M. Miki, S. Walker, E.P.G. Bruun, S. Adriaenssens, LightVault: A design and robotic Fabrication Method for Complex

- Masonry Structures, in: *Advances in Architectural Geometry 2020*, Presses des Ponts, Paris, France, 2021, pp. 350–375. https://thinkshell.fr/wp-content/uploads/2019/10/AAG2020_18_Parascho.pdf.
- [53] I.X. Han, E.P.G. Bruun, S. Marsh, S. Adriaenssens, S. Parascho, From concept to construction: A transferable design and robotic fabrication method for a building-scale vault, in: *Proceedings of the 40th Annual Conference of the Association for Computer Aided Design in Architecture*, Online, 2020, pp. 614–623. http://papers.cuminacad.org/cgi-bin/works/Show?acadia20_614.
- [54] E.P.G. Bruun, I. Ting, S. Adriaenssens, S. Parascho, Human-robot collaboration: a fabrication framework for the sequential design and construction of unplanned spatial structures, *Digital Creativity* 31 (4) (2020) 320–336, <https://doi.org/10.1080/14626268.2020.1845214>.
- [55] L. Euler, Solutio problematis ad geometriam situs pertinentis, *Sci. Imperial. Petropol.* 7 (1758) 9–28. <https://scholarlycommons.pacific.edu/euler-works/53/>.
- [56] E.C. Kirby, R.B. Mallion, P. Pollak, P.J. Skrzynski, What Kirchhoff actually did concerning spanning trees in electrical networks and its relationship to modern graph-theoretical work, *Croat. Chem. Acta* 89 (4) (2021), <https://doi.org/10.5562/cca2995>.
- [57] G. Kirchhoff, Ueber die auflösung der gleichungen, auf welche man bei der untersuchung der linearen vertheilung galvanischer ströme geführt wird, *Ann. Der Phys. Chem.* 148 (12) (1847) 497–508, <https://doi.org/10.1002/andp.18471481202>.
- [58] G. Kron, in: *Elastic structures from the point of view of topological network theory*, in: *RAAG Memoirs of the Unifying Study of Basic Problems in Engineering and Physical Sciences by Means of Geometry*, 1st Edition 3, Gakujutsu Bunken Fukyukai, 1962, pp. 329–337. <https://lccn.loc.gov/60004906>.
- [59] S.J. Fennes, F.H. Branin, Network-topological formulation of structural analysis, *J. Struct. Division* 89 (4) (1963) 483–514, <https://doi.org/10.1061/JSDAEG.0000954>.
- [60] S.J. Fennes, Structural analysis by networks, matrices, and computers, *J. Struct. Division* 92 (1) (1966) 199–222, <https://doi.org/10.1061/JSDAEG.0001379>.
- [61] N. Lind, Analysis of structures by system theory, *Trans. Am. Soc. Civil Eng.* 128 (2) (1963) 786–806, <https://doi.org/10.1061/TACEAT.00088>.
- [62] P. Block, J. Ochsendorf, Thrust network analysis: a new methodology for three-dimensional equilibrium, *J. Int. Assoc. Shell Spatial Struct.* 48 (3) (2007) 167–173. ISSN: 1996-9015.
- [63] P.O. Ohlbrock, P. D'Acunto, A computer-aided approach to equilibrium design based on graphic statics and combinatorial variations, *Comput-Aided Design* 121 (2020) 102802, <https://doi.org/10.1016/j.cad.2019.102802>.
- [64] R. Pastrana, P.O. Ohlbrock, T. Oberbichler, P. D'Acunto, S. Parascho, Constrained form-finding of tension-compression structures using automatic differentiation, 2017 arXiv, arXiv:2111.02607.
- [65] O. Shai, K. Preiss, Isomorphic representations and well-formedness of engineering systems, *Eng. Comput.* 15 (1999) 303–314, <https://doi.org/10.1007/s003660050025>.
- [66] O. Shai, K. Preiss, Representation of Embedded Engineering Knowledge for Design, in: *5th International Conference on Artificial Intelligence in Design*, Springer Netherlands, Dordrecht, 1998, pp. 149–167, https://doi.org/10.1007/978-94-011-5121-4_8.
- [67] O. Shai, K. Preiss, Graph theory representations of engineering systems and their embedded knowledge, *Artific. Intell. Eng.* 13 (3) (1999) 273–285, <https://doi.org/10.1016/S0954-18109900002-3>.
- [68] A. Kaveh, *Structural mechanics: Graph and matrix methods* Macmillan International Higher Education, Macmillan International Higher Education, London, 1992. ISBN: 978-1-349-87760-7.
- [69] A. Kaveh, in: B. Topping, C. Mota Soares (Eds.), *The role of algebraic graph theory in structural mechanics Progress in Computational Structures Technology*, 1st Edition, Saxe-Coburg Publications, Scotland, 2005, pp. 77–110. ISBN: 978-1-874672-21-0.
- [70] J.C. Maxwell, On the calculation of the equilibrium and stiffness of frames, *The London Edinburgh, and, Dublin Philosophic. Magazine J. Sci.* 27 (182) (1864) 294–299, <https://doi.org/10.1080/14786446408643668>.
- [71] T.-S. Tay, W. Whiteley, Generating isostatic frameworks, *Struct. Topol.* 11 (2009). <https://upcommons.upc.edu/handle/2099/1047>.
- [72] C. Calladine, Buckminster Fuller's "Tensegrity" structures and Clerk Maxwell's rules for the construction of stiff frames, *Int. J. Solid. Struct.* 14 (2) (1978) 161–172, <https://doi.org/10.1016/0020-76837890052-5>.
- [73] R. Connelly, P.W. Fowler, S.D. Guest, B. Schulze, W.J. Whiteley, When is a symmetric pin-jointed framework isostatic? *Int. J. Solid. Struct.* 46 (3) (2009) 762–773, <https://doi.org/10.1016/j.ijsolstr.2008.09.023>.
- [74] G. Laman, On graphs and rigidity of plane skeletal structures, *J. Eng. Mathemat.* 4 (4) (1970) 331–340, <https://doi.org/10.1007/BF01534980>.
- [75] M. Grötschel, L. Lovász, A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica* 1 (2) (1981) 169–197, <https://doi.org/10.1007/BF02579273>.
- [76] S.J. Gortler, A.D. Healy, D.P. Thurston, Characterizing generic global rigidity, *Am. J. Mathemat.* 132 (4) (2010) 897–939, <https://doi.org/10.1353/ajm.0.0132>.
- [77] D.J. Jacobs, B. Hendrickson, An algorithm for two-dimensional rigidity percolation: the pebble game, *J. Comput. Phys.* 137 (2) (1997) 346–365, <https://doi.org/10.1006/jcph.1997.5809>.
- [78] S. Bereg, Certifying and constructing minimally rigid graphs in the plane, in: *Proceedings of the twenty-first annual symposium on Computational geometry*, Association for Computing Machinery New York, NY, USA, 2005, pp. 73–80, <https://doi.org/10.1145/1064092.1064106>.
- [79] O. Daescu, A. Kurdia, Towards an optimal algorithm for recognizing Laman graphs, *J. Graph Algorithm Appl.* 13 (2009) 219–232, <https://doi.org/10.7155/jgaa.00185>.
- [80] L. Henneberg, Die graphische Statik der starren Körper, in: *Encyklopädie der mathematischen Wissenschaften mit Einschluss ihrer Anwendungen*, Vol. IV Vieweg+Teubner Verlag, Wiesbaden, 1908, pp. 345–434. ISBN: 978-3-663-16021-2.
- [81] G. Grassegger, Realizations and Constructions of Minimally Rigid Graphs, in: *Bridges 2018 Conference Proceedings* Stockholm, Sweden, 2018, pp. 539–542. <https://archive.bridgesmathart.org/2018/>.
- [82] G. Grassegger, C. Koutschan, E. Tsigaridas, Lower Bounds on the number of realizations of rigid graphs, *Exp. Mathemat.* 29 (2) (2020) 125–136, <https://doi.org/10.1080/10586458.2018.1437851>.
- [83] J. Cheng, M. Sitharam, I. Streinu, Nucleation-free 3D rigidity, in: *Proceedings of the 21st Annual Canadian Conference on Computational Geometry*, 2009, pp. 71–74, in: http://cccg.ca/proceedings/2009/cccg09_19.pdf.
- [84] H. Pollack-Geiringer, Über die gliederung ebener fachwerke, *Zeitschrift für Angewandte Mathematik und Mechanik* 7 (1) (1927) 58–72, <https://doi.org/10.1002/zamm.19270070107>.
- [85] J. Cruickshank, On spaces of infinitesimal motions and three dimensional Henneberg extensions, *Discrete Comput. Geomet.* 51 (3) (2014) 702–721, <https://doi.org/10.1007/s00454-014-9580-y>.
- [86] R.J. Trudeau, in: *Introduction to Graph Theory*, 2nd Edition, Dover Publications, New York, 1994. ISBN: 978-0-486-67870-2.
- [87] T.V. Mele, A. Liew, T.M. Echenagucia, M. Rippmann, COMPAS: A framework for computational research in architecture and structures, 2017. <https://compas-dev.github.io/>.
- [88] P. Fleischmann, G. Casas, M. Lyrenmann, COMPAS RRC: Online control for ABB robots over a simple-to-use Python interface Jul, 2020, <https://doi.org/10.5281/zenodo.4639418>.
- [89] L. Ding, W. Jiang, Y. Zhou, C. Zhou, S. Liu, BIM-based task-level planning for robotic brick assembly through image-based 3D modeling, *Adv. Eng. Informat.* 43 (2020) 100993, <https://doi.org/10.1016/j.aei.2019.100993>.