Algorithmic planning for robotic assembly of building structures

by

Yijiang Huang

B.S., University of Science and Technology of China (2016)
S.M., Massachusetts Institute of Technology (2018)

Submitted to the Department of Architecture
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Architecture: Building Technology

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2022

Author . . . .
Department of Architecture
August 5, 2022

Certified by . . . .
Caitlin T. Mueller
Associate Professor of Architecture and Civil and Environmental Engineering
Thesis Supervisor

Accepted by . . . .
Leslie K. Norford
Professor of Building Technology
Chairman, Department Committee on Graduate Theses

Thesis Supervisor

Caitlin Mueller, Ph.D.
Associate Professor of Architecture and Civil and Environmental Engineering
Massachusetts Institute of Technology

and readers

Skylar Tibbits, SMArchS
Associate Professor of Design Research
Massachusetts Institute of Technology

Tomás Lozano-Pérez, Ph.D.
Professor of Computer Science and Engineering
Massachusetts Institute of Technology

Algorithmic planning for robotic assembly of building structures

by

Yijiang Huang

Submitted to the Department of Architecture
on August 5, 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Architecture: Building Technology

Abstract

This thesis develops the algorithmic foundations for applying automated planning techniques to program robots to assemble discrete spatial structures. Benefitting from the robot's capacity for moving, positioning, and holding elements precisely, robotic assembly aims to neutralize the cost and time impact of increasing demand for non-standard, customized designs using programmable robotics and automated process. Programming robots to assemble structures requires us to reason about the construction sequence and the robotic motions. The critical planning challenge is satisfying both stiffness constraints that limit the deformation of the structure and geometric constraints that ensure the robot does not collide with the structure. Current planning approaches either require a significant amount of human intervention or do not scale to the numeric scale and geometric complexity demanded by construction. As we shift from mass production in manufacturing to mass customization in construction, we need versatile planning tools that can adapt to different structural typologies, off-load tedious human programming work, and involve human expertise when relevant.

This thesis addresses this need by proposing a unified algorithmic framework to formulate and solve assembly planning problems. Our investigations are grounded on three broad classes of assembly planning problems: (1) spatial extrusion, (2) pick-and-place assembly, and (3) robotic assembly with multiple tool changes. For each class of assembly problems, we propose scalable, efficient planning algorithms and test them with simulated and real-world case studies. This thesis demonstrates how algorithmic planning can provide us with a much smoother transition between an assembly design and its final execution on the robot. Based on these sound foundations of the "forward-evaluation" of robotic constructability in various contexts, we finally attempt to "close the loop" - deriving a metric to measure constructability and use it to guide the performance-driven exploration of a discrete design catalog.

Thesis Supervisor: Caitlin T. Mueller
Title: Associate Professor of Architecture and Civil and Environmental Engineering

Acknowledgments

I am extremely grateful for the mentors, colleagues, friends, and family members who have made this thesis possible. First, I would like to thank my advisor, Professor Caitlin Mueller, who had a profound impact on my development as a researcher and made my experience as a graduate student so joyful and rewarding. This research grows out of her patience and her open-mindedness which allowed me to pursue my intellectual curiosity freely without constraints. I am extremely grateful that she agreed to mentor me despite my distinct academic background in the building technology program. Beyond guiding me to become an optimistic, curious, creative and rigorous engineer and scientist, she has taught me that "designer" is a blanket term that even I can identify myself with. Our weekly meeting throughout the past six years has been the most important source of research energy for me - I always felt more confident and motivated to do more after each meeting. Her words of encouragement and support have been always perfectly timed. I aspire to become a similar type of supportive mentor for my students in my future academic career.

I also thank my thesis readers, Professor Skylar Tibbits and Professor Tomás Lozano-Pérez, for their insightful comments and collective visions on the future of automated construction and the relevance and impacts of automated planning tools. In addition to their intellectual contributions, my committee has been incredibly supportive and understanding of their mentee that always has the tendency to procrastinate. I am grateful for their empathy and kindness.

The works presented in this thesis all came out of collaborative efforts, both internally at MIT and externally across the globe. Caelan Garrett has been by my side since the early days of this research and has become a friend and an important mentor for me. He taught me about planning more than anyone else both in theory and in implementation. I am thankful for having been able to work with him, which started from a random email I sent him in 2016 asking for a chat on research. Victor Leung from the Gramazio and Kohler Research (GKR) at ETH

Zurich has been a patient, productive, and detail-focused collaborator from whom I learned a lot about the software and hardware challenges of on-the-ground automated construction. His dedication and open-mindedness have made our collaborations across two continents a smooth and rewarding experience during the COVID global pandemic in the past two years. I am also grateful for having the opportunity to work with Ian Ting and Professor Stefana Parascho from EPFL (previously Princeton) on the bar assembly project, Gonzalo Casas and Romana Rust from GKR on developing COMPAS-FAB in 2019, Inés Ariza from GKR on planning paths for in-place robotic welding, and Lavender Tessmer from MIT on additive robotic casting.

I also thank the Building Technology faculty - Professors John Ochsendorf, John Fernández, Leon Glicksman, Caitlin Mueller, Leslie Norford, and Christoph Reinhart - for fostering such a caring and supportive work environment in the BT lab. The BT lab and the Architecture Department here create the perfect learning and research environment that closely links design and engineering, from which I have benefited a lot. I also thank the students of the BT lab, and in particular, the past and current members of the Digital Structure Group, including Nathan Brown, Hanne Cloesen, Pierre Cuvilliers, Renaud Danhaive, Demi Fang, Eduardo Gascon, Kiley Feickert, Ashley Hartwell, Mohamed Ismail, Keith Lee, Paul Mayencourt, Bryan Ong, Courtney Stephen, Mark Tam, Jaya Manideep Rebbagondla, Jingwen Wang, Eamon Whalen, Leila Sory, and Ramon Weber. It was truly a privilege and joy to interact with them, and they helped shape this research through many insightful discussions over the years. I am also thankful for the brilliant graduate and undergrad students I had the chance to interact with through my teaching experience, in particular for the students that I met during my teaching assistance experience for 4.450 and instructor experience for 4.S48, Design for Robotic Assembly, a class that I co-developed with Caitlin Mueller and Justin Lavallee in 2018, as well as Thomas Cook, Kodiak Brush, and Khanh Nguyen, whose UROPs I was lucky to co-supervise. In addition, I especially thank Professor Justin Solomon and members of his Geometric Data Processing Group at CSAIL for taking me under their wings and allowing me to join their various meetings so that I can have a sprinkle of geometry (and get Wassersteinized) during my Ph.D. experience here. This is very important for me since geometry and related math always have a special place in my heart.

I am thankful for various MIT staff who had helped me on numerous occasions. Fabrication is never an easy task and I am thankful for the patience and help from Justin Lavallee, Jen O'Brien, and Chris Dewart of the architecture fabrication shop. I also thank past and current BT admin Kathleen Ross, Erin Buckley, Stacy Clemons, and Jennifer Roesch for their patience, love, and efforts in keeping the BT lab functioning and endowing it with a touch of warm feeling of a family.

I gratefully acknowledge several generous funding sources during my Ph.D. studies: the MIT Presidential Fellowship Program (2016-2018 and 2018-2022), the MIT Department of Architecture, and the National Science Foundation. The Chapter 6 of this thesis was partly supported by the National Science Foundation under Grant No. 1854833. My research stay at GKR of ETH Zurich in 2019 was made possible by generous support from MIT's MISTI Switzerland program, with the heartful coordination of Justin Leahey at MISTI and Hannes Mayer and Tanja de Almeida Pereira from GKR.

Before my time at MIT, faculties at USTC shaped my interests and growth as a scholar, and I am particularly grateful to professors Ligang Liu and Juyong Zhang for making my first research experience possible. I am grateful to Mr. Lei Yu at the AWS workshop who got me excited about architecture and taught me about various architectural principles and hands-on techniques for developing digital fabrication tools.

The last six years at MIT have been like living in a dream for me. I am extremely grateful for the time I have spent learning, exploring, growing, and just living in the MIT community. I met many friends through taking a large variety of classes at MIT, working at the fabrication shops, and playing more than an appropriate amount of soccer on the Roberts field. This is also where I met my wife Stella, together with whom we adopted two cats Morris and Meimei, resulting in our current family of four. I am also thankful to the teammates at the BKP soccer club for allowing me to escape from my research and play my favorite sport every week (sometimes every day).

I am beyond grateful to my parents, who have been supporting me emotionally, intellectually, and financially throughout my academic career. Mama, you gave me the best childhood that a kid could imagine. Papa, you instilled in me the importance of knowledge and academic research. Finally, none of the works presented in this thesis would have been possible without the unconditioned love and support from my wife Stella. Together with our cats Morris and Meimei, I wish to push the boundary of knowledge and enjoy the joy of life with you forever. Merci.

Relevant Publications

This dissertation incorporates materials from several papers published over the last several years, in particular the following: [GHLPM20, HGT+21, HVG+21].

These papers were developed and written jointly with many coauthors, who no doubt contributed text appearing in this dissertation. This work would not be possible without their patient collaboration.

Contents

Chapter 1

Introduction

This thesis develops the algorithmic foundations for applying automated planning techniques to program robots and other automated or semi-automated systems to systematically assemble discrete spatial structures.

Throughout the history of humankind, construction has been an endeavor that relies on master builders' craftsmanship and intuition to produce one-off solutions [Fit89]. However, due to the increasing demands for housing from urban expansion, construction needs to be more flexible, economical, and environmentally friendly. Due to the decline of craftsmanship in modern society, such an experience-based and relationship-based construction model does not respond well enough to such demands. One direction for improvement is to increase automation in construction, and more specifically, applying robotic systems to construction work and the fabrication of architectural elements. Benefitting from the robot's capacity for moving, positioning, and holding elements precisely, robotics-based assembly promises to substantially improve construction in the built environment. It offers advantages such as speed, quality, material efficiency, worker safety, and eventually, cost reduction. In addition, robotic construction opens up new fabrication possibilities beyond automating manual processes, leading to an expansion of design possibilities and formal expressions.

However, to this day, the commercial use of industrial robots is still mostly limited to repetitive tasks in a controlled manufacturing environment that can be programmed manually in advance. For the main customers of industrial robots, e.g. the automobile industry, employing a few technicians to spend hours or days manually programming the robot to assemble, weld or paint a single product works out economically since the same program would be executed on the assembly line for thousands or millions of times. In such mass production contexts, the robots are adapted to fulfill the needs of a high-volume market, delivering high reliability and productivity. Implementing a flexible manufacturing process that can respond to design changes swiftly comes at a high cost and is thus rarely needed and touched upon in the manufacturing industry [Bro07].

However, the construction industry comes with inherent complexity that makes a direct transfer of mass production techniques in manufacturing impossible. In construction, high-performing structures are designed specifically for each job site and condition. The load resistance and expressiveness of the structure come from the diversity of components' geometry and configurations. Such diversity means that hiring technicians to solve the programming problems for each part is not economically feasible since each program might only be executed a few times, instead of thousands of times as in the manufacturing setting. Thus, as we transition from mass reproduction in manufacturing to mass customization in construction, the economic and time costs coming from the technical difficulty of generating a detailed sequence of robotic instructions start to dominate. This process of programming the robots to achieve a high-level goal like assembling a product is termed *planning*. Automating the planning process so that it can respond to the forementioned diversities swiftly and intelligently remains the fundamental challenge to be solved before robotic techniques are widely adopted and their impacts are broadly felt on real construction sites.

Our objective is to build planning tools that closely connect design and robot assembly. Such tools should be equipped to give designers immediate feedback on whether a design is *feasible* for being assembled by robots while they are still developing the design in a CAD environment. Furthermore, such tools should be able to present useful information back to the designers on how to *improve* the design if it is not feasible initially. Thus, these planning tools can go beyond being simply a "reality-check" tool, to become a "design-guidance" tool through which machine-related fabrication logics and constraints can become design drivers in the development of an architectural project.

1.1 Motivation

This research is motivated by the challenges and opportunities which has emerged in recent years, as researchers and practitioners continue to push the boundary of fabrication possibilities of robotically assembled geometries.

1. **Complex assembly is hard to plan for, and the current approaches fall short.** Planning for robotic assembly requires considering both *geometric constraints* to ensure robots respect collision and kinematic constraints, as well as *structural constraints* so that at each step, the partial construction is structurally safe. Existing solutions in architecture rely on a lot of manual trial-and-error efforts and often lead to a lot of manual backtracking that renders these approaches ad-hoc and inflexible to design changes.

2. **Current planning algorithms cannot fully address the unique challenges in robotic assembly.** Despite the exciting algorithmic development in the robotic planning community over the years to automatically plan for both abstract robot actions and detailed motions, its adoption in automated construction remains low. This is partly due to these advanced algorithms requiring domain-specific knowledge to *encode* the planning problems at hand to an abstract format, which can be unfamiliar to many users in architecture. In addition, construction usually involves manipulating hundreds or even thousands of objects in complex spatial arrangements, which is rarely tackled in standard planning algorithm benchmarks. Thus, assembly planning algorithms need to be designed to be efficient and accessible specifically for construction problems.

3. **Planning can reveal new ways to evaluate designs.** The history of considering performance early in the design process usually is prefaced by the development of quantification, calculation, and computer simulation of the performance metric. Much progress has been made in the last decade to integrate complex engineering simulations into design environments by packaging complex analyses into user-friendly tools and plug-ins (e.g. [Pre13, Sol22a, RPS+13]). Combined with state-of-the-art optimization and learning machinery, diverse and high-performing design candidates have been uncovered and used to guide the designers [Bro19, Dan20]. Construction, while historically a completely manual process and hard to quantify, can now be precisely planned and executed in the context of robotic automation, or more generally, algorithmically planned construction. Thus, can we use construction to drive design decisions, turn infeasible into feasible, or minimize certain objectives, e.g. the maximum displacement during construction?

1.2 Research questions

Building from these motivations, a single, broad research question motivates all the work in this dissertation:

> *How can algorithmic planning expand the relevance and impacts of robotic assembly?*

In response to this question, as well as the challenges and opportunities outlined in the previous section, this dissertation tackles two axes of inquiry:

1. Can we design effective planning algorithms to streamline the design-to-robotic assembly workflow?

2. Can assembly planners support constructability-driven design space exploration?

These questions are addressed by proposing new planning and design evaluation strategies by leveraging recent advances in computational planning techniques and by marrying them with knowledge from construction engineering. Our investigations are grounded on three broad classes of assembly planning problems, namely (1) spatial extrusion, (2) pick-and-place assembly, and (3) robotic assembly with multiple tool changes. For each class of assembly problems, we start off by formulating the problem into an abstract, mathematical form so that it can be analyzed using an algorithmic lens. Then, we make algorithmic contributions to propose planning strategies to address the specific challenges and lead to scalable, efficient planning algorithms. Simulated and real-world case studies illustrate these approaches and demonstrate how we can obtain a much smoother transition between an assembly design and its final execution on the robot, resembling our experience with other existing digital fabrication machines like 3D printers and CNC machines. Based on these sound foundations of the "forward evaluation" of robotic constructability in various contexts, we finally attempt to "close the loop" - deriving a metric to measure constructability and use it to guide the performance-driven exploration of a discrete design catalog.

1.3 Research scope

In this thesis, we approach assembly planning in its general form without restricting the input shapes to any specific category or typology. The definition of a discrete spatial structure broadly includes all 3D structures that consist of individual elements connected via structural joints and behaves as a system when a load is applied.

In terms of robot setup, we make the following assumptions:

1. **Offline, long-horizon planning:** we consider detailed, long-horizon planning to fully specify the robots' behavior during the entire construction process. In all the real-world verifications presented in this thesis, the planning is performed offline before execution, and the plan is executed in an **open-loop**

fashion. The robots are assumed to work in a **deterministic** and structured world, which means that the environment will behave exactly as specified in the computed plan.

2. **Robot dynamics ignored:** we only compute robotic paths instead of trajectories, meaning that we do not explicitly address the system's dynamics. The control of the robot is assumed to be executed with position-based control, implemented by the industrial robot controllers in use.

Assumption 1 is made to ensure that the planning considers the entire construction process rather than a short control horizon. This assumption adheres to today's prevalent use cases of construction robotics for the prefabrication of building components. These prefabricated components will be transported to the construction site and assembled into the already-built structure of the building [KCKH12]. In these cases, the robots work in a structured factory setup, with little uncertainty in the environment. Thus, the automated, long-horizon planning methods presented in this thesis can be directly deployed to such settings. However, while such an assumption is practical to help leverage existing robotic technology, there are many building processes that require components to be built in place and thus require the robots to operate in a poorly structured and uncertain environment. Research in this area, termed in-situ construction robotics, has tackled challenges in short-horizon, robust planning and control [Gif18], sensorial tracking and estimation [San18], and these robotic workflow's integration into architectural design [Dö18]. We argue that even under such contexts, the long-horizon construction planning approach presented in this thesis is still essential and can be integrated. The computed long-horizon plan can serve as a blueprint-like baseline upon which the adaptive, feedback-driven sensing and control strategies can make local modifications.

Assumption 2 is made to focus our investigation on planning problem's geometric and structural aspects without having the dynamics complicate the picture. It is also because all the robotic systems we have access to are either fixed-based or mounted on a Cartesian-type gantry system, as in most prefabrication construction settings. However, for most mobile-based construction robot platforms, e.g. systems with a robotic arm mounted on a non-holonomic mobile base [GSD+17], or a legged robot [ZPC21], the system dynamics become another major constraint that cannot be ignored. Such constraint is not explicitly addressed in this thesis's algorithmic treatment and physical demonstrations.

1.4 Thesis overview

This thesis is divided into seven chapters. This first chapter serves as the introduction and gives the main problem statement, while also articulating the overall research question that motivates and echoes across the remaining chapters.

In Chapter 2, a critical literature review is provided to contextualize the new research presented in this dissertation. This review summarizes past research attempts in construction robotics and assembly planning, scattered across the various industrial and academic fields. The chapter highlights gaps in the existing literature and further motivates the focus of this thesis.

In Chapter 3 and Chapter 4, we describe scalable planning algorithms in the context of sequence and motion planning problems, where one robot follows a repetitive action pattern with a construction sequence undetermined. In Chapter 3, we first present a rigorous mathematical formalization of robotic spatial extrusion planning problems and provide several efficient and probabilistically complete planning algorithms. Then, in Chapter 4, we generalize the proposed algorithms to solve the planning problems in pick-and-place assembly domains, despite that very different materials and construction systems are involved. With the contributions of these two chapters, we devise a unified yet efficient strategy to solve sequence and motion planning problems.

In Chapter 5, we discuss formulation and planning strategies for more general task and motion planning problems arising in robotic construction, where the robot needs to compose actions that follow a non-repetitive pattern to achieve a high-level assembly goal. In this chapter, a spatial timber assembly process is used as an illustrating example. This construction process requires a robot to swap tools in order to switch between placing the beams and fixturing the joints. We establish a flexible modeling protocol for architectural and construction process designers to convert their high-level design intents to a computable plan skeleton, which can be solved by a newly proposed solving technique by chaining multiple motion planning calls effectively.

In Chapter 6, we investigate the questions of how automated planning techniques can be used to formulate and compute a new measure of constructability and how this new measure can be used to inform design. We propose new optimal search algorithms to uncover optimal construction sequences for scaffolding-free assembly of bar structures, which can be used to formulate an optimal constructability score of a given design. We present a case study to use this score to evaluate and compare a design catalog.

In Chapter 7, we conclude the thesis by describing directions for future work and reflecting on the thesis as a whole.

Chapter 2

Literature review

This chapter presents a critical literature review that spans across several, seemingly disjointed fields: performance-driven design in architecture (Section 2.1), robotics in construction (Section 2.2), design and analysis for manufacturing (Section 2.3), computational assembly analysis and design research in computer graphics and computer-human interaction (Section 2.4), and automated planning (Section 2.5). This chapter identifies and discusses specific developments in each of these areas, and illustrates the need for further research that the work of this thesis addresses.

2.1 Performance-driven design in architecture

An imperative for sustainability has been driving building designers and engineers to more closely consider various building performance in response to the fact that buildings account for a large portion of primary energy consumption ($\sim$40% in the United States), including significant amounts of embodied energy[WL17]. Computational design methods are well-suited to support these much-needed performance-driven design explorations. This section summarizes key work in this area.

2.1.1 Design evaluation

The ability to evaluate design options using a quantitative measure is central to performance-driven design. This usually involves running simulations, such as structural or thermal, based on the geometric attributes of a design and summarizing the analysis results to one or a few metrics characterizing how good a design may be. Before the last decade or so, the tools used to run complex simulations were far removed from design software platforms for geometric design. These standalone, specialized analysis tools were not meant to enable a flexible exploration of design variations. Instead, they were used as "reality-check" tools, used downstream of varying the design. In addition, these tools usually come with a steep learning curve and high costs, adding yet another layer of friction between architects and engineers.

While this paradigm still prevails, many simulation tools that are lightweight and easy-to-use have been integrated into existing 3D modeling software like Rhino3D [Rob22]. These tools are designed to be used for non-expert users and are closely linked to other geometric manipulations in the same digital environment. However, despite these advances and the increased accessibility of simulation in early-stage design, many drawbacks associated with engineering simulation remain. The most consequential one is the speed of the simulations. The objective of integrating performance feedback early on in the design process is to help designers converge on better solutions. Slow feedback results in fewer explorable designs; it also frustrates the design process by preventing real-time user experiences. Accelerating simulation speed can generally be achieved by either improving the existing algorithms or taking advantage of parallel or GPU-accelerated computation (e.g. [JR19]). However, these options are not easily achievable by non-experts, including design users.

In contrast to these strategies, a data-driven method called surrogate modeling uses performance data obtained by running high-fidelity analyses on a number of different design options to learn a regression function that provides an approximate model of the performance function of interest. The key advantage of this method lies in the speed of prediction of the regression models: they tend to be orders of magnitude faster than the original simulations used to train them. Even though the field of surrogate modeling is established as early as 1951 and has evolved into a mature field of research [FSK08], the introduction of surrogate modeling to problems related to the built environment is fairly recent. [Mue14] uses ensemble regression methods to predict the performance of structural concepts and uses these models to accelerate performance feedback for optimization and design space exploration. There are also attempts at training surrogate models that may be repeatedly used across different structural design spaces [TBM16], including the recent attempt in building transferrable graph neural networks on truss domains [WM21]. Taking advantage of the recent advances in deep learning, [Dan20] proposes a workflow to extend the surrogate models to the prediction of spatially distributed simulation fields (Figure 2-1).

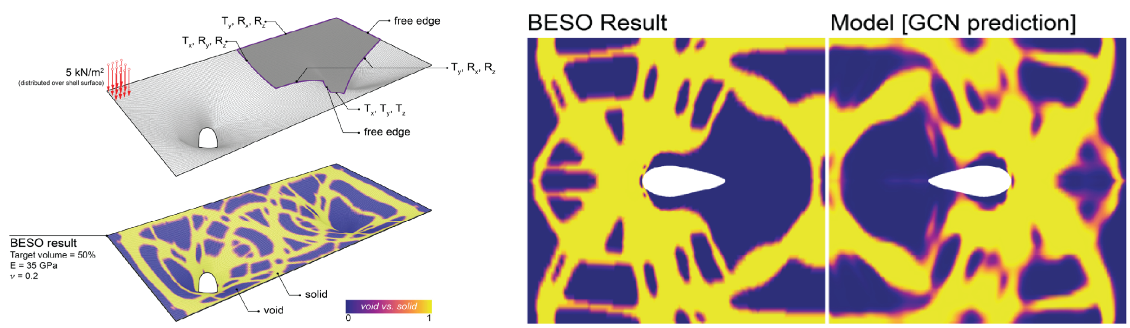


Figure 2-1: Field surrogate modeling (using a graph convolution network) for predicting topology optimization result of a shell [Dan20]. Images by corresponding papers.

Beyond the traditional physics simulation models that are based on computationally solving partial differential equations, new performance scores and their simulation tools are being developed to support new types of performances and design paradigms. For example, availability evaluation aims to evaluate how a structure can be built using reclaimed or recycled materials, driven by the circular economy principles. A review of recent computational strategies for circular design can be found in [HADWM21].

2.1.2 Design space formulation

Computational design methods aim to generate diverse design outcomes and directly connect those options to performance simulation. This allows the generation of numerous possibilities from a single system. The set of all possible designs that can be generated from a given system, is called the design space. The generative system, depending on its constructs, can be summarized into three categories: parametric model, form-finding model, and shape grammar.

Parametric model Parametric design formalizes the concept of design space as a map from design variables, i.e. numerical parameters, to a particular design sample, which can be evaluated according to one or multiple criteria of interest (see Figure 2-2 for an example). Popularity of built-in visual programming platform like Grasshopper3D on Rhinoceros 3D [Rob22] makes such practices widely accessible [Ted14].

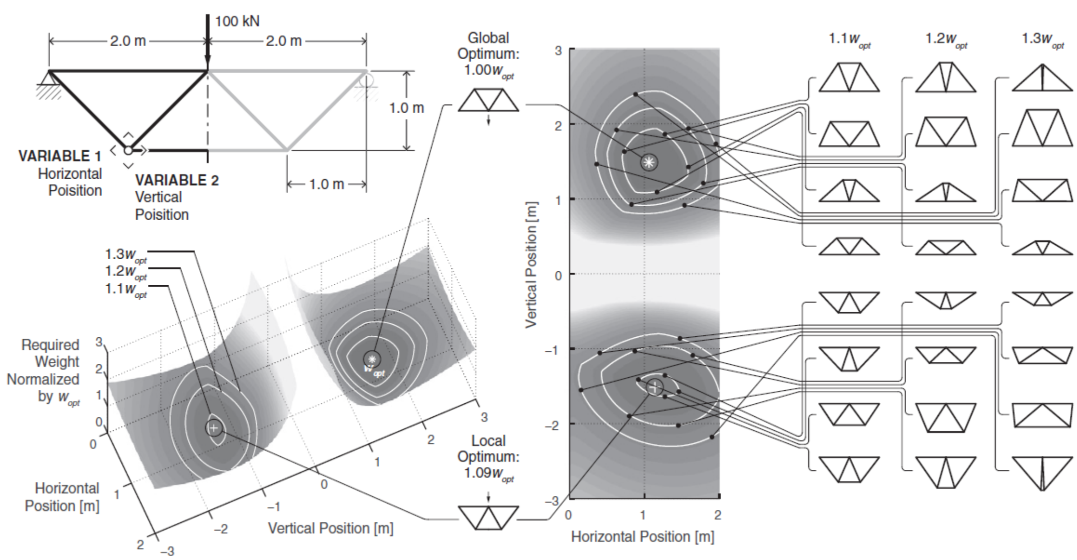


Figure 2-2: The design space of a weight minimization problem of a seven-bar truss parameterized by two variables. Images by [MO15].

However, in contrast to more traditional, analog ways of design modeling, such as sketching and physical model making which allows almost infinite freedom, codifying design into a parametric framework does dramatically restrain the design possibilities. First, it is well known that morphing geometry is much easier than morphing topology. Second, the process of formulating a design space itself is an art. Creating a diverse and interesting parametric design space demands proficiency in computational literacy and tools. Recent efforts propose to mitigate this problem by using machine learning techniques to create low-dimensional, synthetic design spaces from datasets [Bro19, Dan20, ODM21].

Form-finding Form-finding is a forward process in which parameters are explicitly controlled to find an "optimal" geometry of a structure that is in static equilibrium with a design loading [ABVW14]. Examples of form-found shapes include networks of hanging chains, stone arches, the fins and tensioned fabric of an umbrella, etc. This typical method is categorized separately from the generic parametric model from the last section because a specific type of structural performance is built into the form-finding process, and the design parameters are not directly connected to the final geometric shape. Typical design variables include topologies of a base grid, boundary conditions, and other physical parameters. With these design variables, designers have little direct control over the resulting shape, but can only "play" with the parameters until the desired shape is reached. Computational form-finding has a long history in engineering literature, aiming to reproduce the physical experiments of finding the equilibrium positions, e.g. finding the form of a funicular shell by a hanging network [Sch74, KO05, VB12]. Computational form-finding has been successfully applied to a variety of systems, like tensile structures [Bar99], compression-only funicular structures [BH14], bending-active structures [AB01], etc. (examples in Figure 2-3). Projective dynamics is a general method that bridges the finite-element method and position-based dynamics to create a form-finding tool that is adaptable to a wide range of applications and simple to implement yet robust [BML+14]. Its integration into the popular CAD system Rhino3D [Rob22] opens up the ideas and methods of form-finding to a wider audience and more applications than ever before [Pik22]. However, the tool does not necessarily guide the designer towards more efficient structures for the chosen construction system, and it does not provide intuition on how to find structures that are closer to the design intent.

One method, in particular, stands out for providing users with more control over the final shape. Thrust network analysis (TNA) combines visualization from graphics statics and the force density method to create a form-finding tool that provides designers with an editable force diagram [BO07]. By redirecting the flow of forces on the diagram, designers can create features like creases and ridges. TNA has been implemented as a popular design tool RhinoVault [RLB12], serving as a basis for the design of numerous successful masonry vaults in practice and in the design studio (Figure 2-3-(2)). Recent research looks into the wider design possibilities offered by topological patterns of the base grid [ORM+19]. Systems with similar characteristics that negotiate design intent and equilibrium through certain protocols include combinatorial equilibrium modeling, where graph theory is used to control the qualitative behavior of funicular structures [OOS16], and 3D graphic statics [Lee18].

Recent developments in form-finding move towards giving designers more control by coupling form-finding with an outer optimization layer that minimizes the distance between the form-found shape and the user-provided shape. Readers are referred to [Cuv20] for a comprehensive review on form-finding, including its most recent advances.

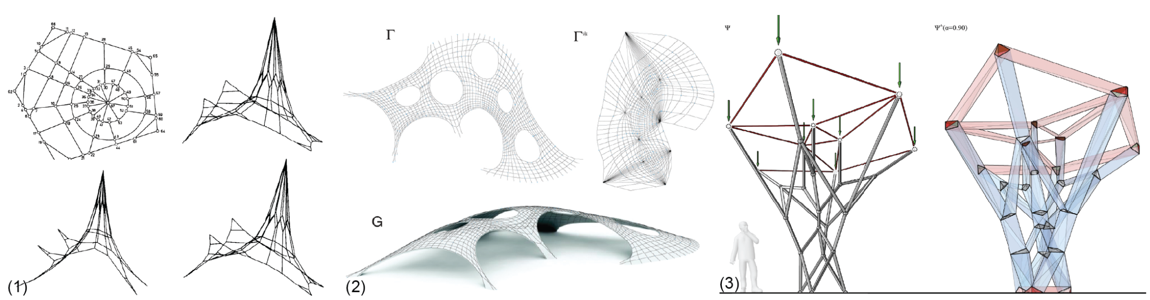


Figure 2-3: Examples of form-finding techniques: (1) tensile structures found by force density method with different force density values on the same base grid [Sch74]; (2) Form and force diagram and the resulting funicular network found by Thrust Network Analysis, implemented by RhinoVault [RLB12]; (3) Polyhedral reciprocal form and force diagram using 3D graphic statics [Lee18]. Images by corresponding papers.

Shape grammars A rule-based, grammatical system works by defining an initial state upon which a set of defined rules are iteratively applied and expanded. These procedurally generated design spaces are often much richer than the parametric design spaces with a fixed number of design variables. By relying on a fixed set of rules that can be flexibly applied, they are well suited for the generation of diverse and truly surprising solutions. Its application in design of patterns [Sti77], houses [SM78, KE81], and structures [SC99, MO13, LMF16, Clo18] have proven it a powerful tool for computational design ideation and brainstorming (examples in Figure 2-4).

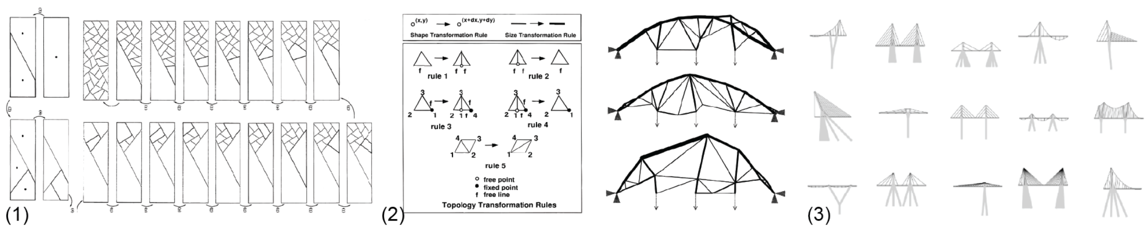


Figure 2-4: Examples of shape grammars: (1) procedural generation of Chinese ice ray pattern [Sti77]; (2) performance-driven grammatical generation of trusses - rules and resulting designs [SC99]; (3) Pedestrian bridges randomly generated using a structural grammar [MO13]. Images by corresponding papers.

However, because grammatical design spaces jump out of the mathematical framework of a mapping between parameter space and shape space, they are notoriously hard to control. Precisely because it has a variable number of parameters, its application has been limited in practice. However, in recent years, plugins like WASP [Ros22] have re-invigorated interest in grammatical systems.

2.1.3 Design space exploration

Parametric design is appealing and useful to designers because it allows them to explore design variations by applying numerical changes to the design variables. This is typically achieved by playing with sliders, the most popular user interface for numerical manipulation in most parametric environments. When fast performance feedback is available, designers can quickly explore regions of the design space through random and educated guessing. While this is straightforward for small design spaces, interacting with many sliders quickly proves difficult and daunting for large ones, especially when non-trivial performance metrics are involved. To enable faster, more systematic, and more effective searches of large design spaces, many strategies have been proposed to automatically generate, record, and compare design options. Existing design exploration methods can be largely categorized into sampling and optimization methods. Comprehensive reviews on existing efforts in performance-driven design space exploration techniques can be found in [Mue14, Bro19, Dan20]. Augmented by these performance-driven exploration techniques, designers are empowered to explore diverse, high-performing subspaces of the originally overwhelmingly vast design space.

2.1.4 Reflections

Recent advancements in computational performance-driven design have equipped designers with powerful tools to design efficient structural forms. However, since structural efficiency is achieved by the precise alignment of load demand and material capacity, efficient forms usually come with geometric complexity that resists the standardization that is pervasive in the construction industry today. We are at the stage where our abilities to generate high-performing designs go far beyond what the current construction paradigm can fabricate and build.

In the current construction practice, the only way designers can get information on constructability beyond using their own intuition is to wait until the design is finalized and get an evaluation from the contractor. In some cases, even when designers invent novel construction solutions to better materialize their designs, the contractor might refuse to cooperate with the proposed technologies that are unfamiliar to their expertise and retreat to the standardized solutions that they know and trust. The designers' sense of powerlessness with respect to control over the construction process is best exemplified by the design and construction of the facade of the Qatar National Convention Center. The building is designed by architect Arata Isozaki and its iconic, tree-like facade was designed by structural engineer

Matsuro Sasaki (Figure 2-5-1). Sasaki used topology optimization, an advanced computational method to find optimal material distributions to sustain loads, to generate the geometry. The structure was conceived to be built with a layer of sandwiched, stressed skin, consuming a fairly small amount of materials compared to standardized solutions. Furthermore, in his book, Sasaki detailed his thinking on how the construction should be executed, including the construction sequence(s) (Figure 2-5-2).

Figure 2-5: The design of the Qatar National Center's facade. (1) picture of the finished structure, photo credit to Nelson Garrido; (2-3) the detailed construction sequence and the sandwiched skin conceived originally by Matsuro Sasaki, image from [MS08].

However, the contractor, Victor Buyck Steel Construction, asserted that Sasaki's design and construction proposal was infeasible to be fabricated and transported. Instead, the facade was built from a series of straight, octagonal steel elements, with a non-structural cladding affixed to the enormous beams [JB17] (see Figure 2-6). The result was unideal in terms of sustainability - the final construction only delivers the appearance of an optimized structure, but in fact, it was built using conventional, prismatic elements that fail to truly capture the original intent of the designers. The final built structure not only undoubtedly performs differently than Sasaki's initial optimization intended, but also consumes much more material. Although the contractor is not the only one to be blamed for such an outcome, their resistance to exploring new construction solutions together with the designers provides a perspective from which we can see the reason for the current stagnated development in the construction industry.

The current closeness of the construction trade makes evaluations of constructability opaque and depends heavily on the specific contractor's intuition, experience,

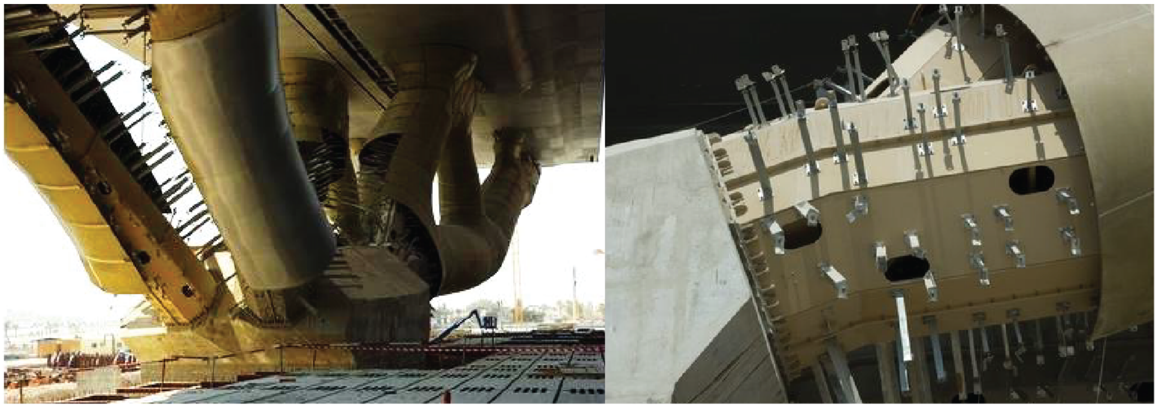


Figure 2-6: The prismatic steel core and the non-structural cladding used to mimic the original optimized structural skin. Images by Victor Buyck Steel Construction.

and imagination. Such relationship-based ways of evaluating whether a design is buildable deprive the designers of the opportunities to make decisions early on in the design process to address construction-related concerns and related performances, since they do not have a way to simulate, plan, and quantify the construction process. We need a more systematized way to plan and execute construction in order to reduce cost and risk, especially when fabricating complex building components demanded by efficient structural forms.

Automation is believed by many to be a promising step towards this future of construction. It is conceived to be able to improve accuracy and reduce the risk and uncertainty that might come out of human labor when fabricating complex geometries. Furthermore, automation has the potential to change the economy of bespoke fabrication, neutralizing the impacts on both time and cost. As an important branch of development in construction automation, robotic construction concerns the use of programmable robots to automate construction tasks. In the next section, we provide a historical review of the development of construction robotics in the last 40 years.

2.2 Robotics in construction

The historical development of construction robotics can be grouped into two stages: the first from 1980 to 2000 focuses on the use of machines for *standardization* to increase productivity, and the second from around 2010 to the present day focuses on the use of robots for *customization*. A timeline that summarizes key research from 1980 to today is presented in Figure 2-7. In the first stage of development, driven by a managerial mindset and developed by mostly construction process engineers, the design considerations come subservient to the specific building typologies imposed by the employed automated construction technology. In contrast, the second stage of development is driven by designers and architects. They adopt a new approach that applies robotic systems not only as means for automation of construction work but for design exploration. This section summarizes key research that emerged from these two stages and argues that automated planning is an important, underdeveloped area of research that is critical for expanding the relevance and impacts of construction robotics.

2.2.1 Productivity-driven construction robotics (1980-2000)

The early attempts of introducing robots to construction were part of a general agenda to rationalize and industrialize the building industry, driven by increased demand for housing in the post-war boom years between 1950 and 1970. With the "robot boom" in the general manufacturing industry in the 1970s, people see applying robots to construction as the logical next step to boost productivity in this sector [BL14]. The first traceable research efforts in construction robotics started in 1978 in Japan as a collaboration among universities, robot manufacturers, and general contractors [Has00]. Since then, applying robotics and automation in construction has been continuously attracting interest, evidenced by the founding of the International Association for Automation and Robotics in Construction (IAARC) in 1990. The association's annual symposia proceedings, which originated in 1984, give a

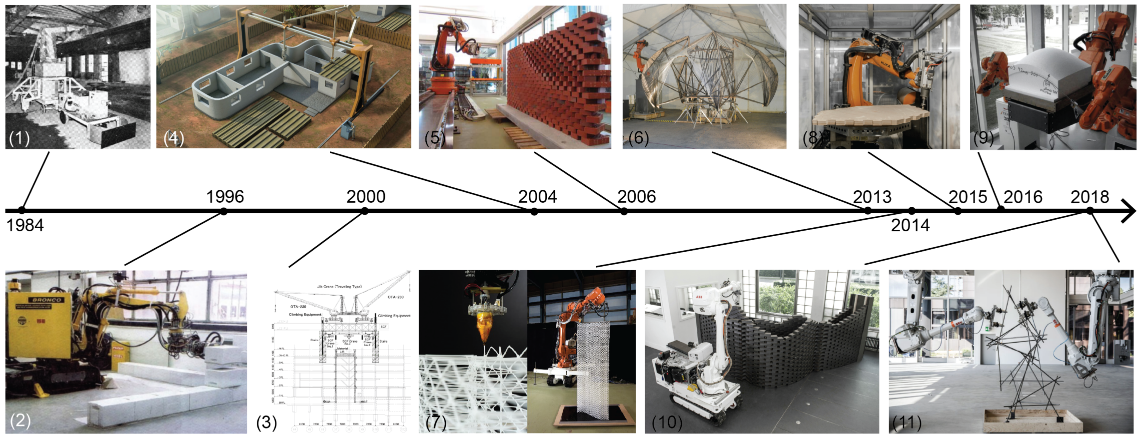


Figure 2-7: A timeline that summarizes key research of construction robotics in the last 40 years. (1) SSR-1 fireproof spraying robot [YUNY84]; (2) robotic masonry construction system BRONCO [PDKG96]; (3) Automated Building Construction System (ABCS) [IH06]; (4) 3D printing construction system Contour Crafting [Kho04]; (5) robotic brick stacking [Tob15]; (6) robotic fiber winding [WSS+13]; (7) robotic spatial extrusion for creating a mesh mould [HL14]; (8) robotic milling [KSM+15]; (9) robotic hot wire cutting [SFN+16]; (10) mobile robotic brick stacking [Dö18]; (11) cooperative robotic assembly of steel bar structures [PGM+17]. Images by timeline compilation made by the author. Individual images are credited to corresponding papers.

comprehensive overview of the research devoted to this area[1].

In contrast to programmable machines for manufacturing, e.g. CNC machines, industrial robots were primarily conceived as handling and assembly tools from their onset. At the time, Japan's restrictive foreign workers' policy, mingled with a shortage of skilled labor and an aging workforce, generated an active need for increasing productivity in construction. Thus, early research in this area is geared towards substituting the human workers on site with robots. Robots are conceived to perform specialized tasks, such as distributing materials, fitting equipment to ceilings, setting interior walls, welding steel members, painting, etc.

In 1984, one of the largest Japanese construction giants, Shimizu Construction Company, released the Shimizu Site Robot-1 (SSR-1) that performs fireproof spraying [YUNY84] (Figure 2-8-1). While demonstrating the feasibility of applying robots for on-site construction work, SSR-1 was too large to be transported and the con-

[1] https://www.iaarc.org/publications/

trol of the robot was so complex that it required specialized operators to attend to it all the time. Although SSR-1 was developed on a commercially available robot arm, the development focused on customizing and optimizing the robotic device to perform specialized tasks. Along this line of task specialization, multiple construction companies developed mobile robotic devices for surface finishing of concrete slabs, which came to widespread use in practice [TSA+86, KFI88, KY89] ((Figure 2-8-2,3)). Many of these specialized machines lacked the flexibility to be applied to construction tasks other than their predefined processes. The level of autonomy is also relatively low in these robotic devices, with the majority of them requiring teleoperation from humans [TWS03].

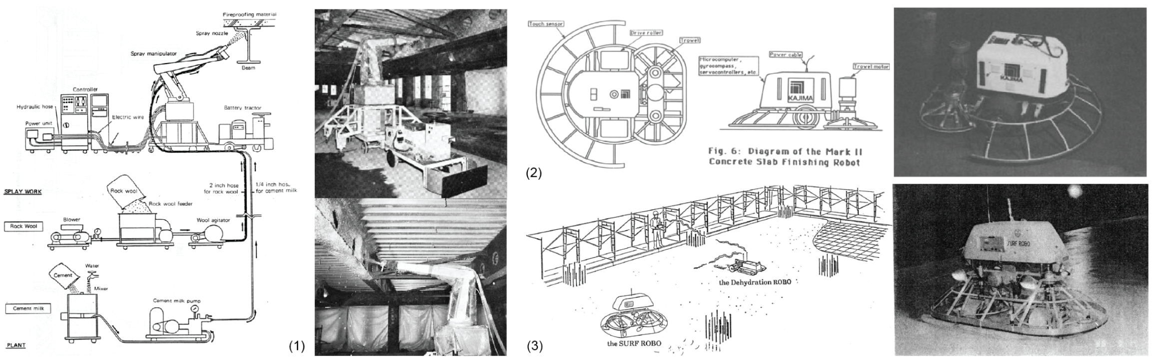


Figure 2-8: (1) SSR-1 fireproofing spray robot [YUNY84, Yos06]; (2) Mark-II concrete surface finishing robot [TSA+86]; (3) Surf Robo concrete surface finishing robot [KFI88]. Images by corresponding papers.

Alongside single-purpose construction robots, integrated construction systems were developed to create a factory-like, structured environment on the construction site. In the 1980s, multiple Japanese construction firms developed these systems for robotic construction of high-rise buildings, including Obayashi Corporation's Automated Building Construction System (ABCS) [MOOS00, IH06] (Figure 2-9), the Big-Canopy system [WFIS00], and the Shimizu Corporation's Manufacturing System by Advanced Robotic Technology (SMART) [YM98a]. These vertically moving construction factories provided a fully enclosed and systematized working envelope within which robots could perform diverse construction tasks, such as automated welding of steel frames, placing of prefabricated concrete floor panels, and placing of interior and exterior wall panels. The construction platform could be raised to complete the next floor. The core themes of these integrated systems are standardization and prefabrication. For example, the wall panels processed by the SMART system feature special joints that are specifically designed for the robotic system [Mae94, YM98b]. In addition, the working platform limits the building to certain vertical configurations and thus limits the design freedom and possibilities for the design to respond to surrounding conditions. Remarkably, these early integrated systems envisioned not only the robotic machinery on the building site but also a complete information system to streamline the workflow from planning to construction. This concept of an integrated information management system was referred to as Computer-Integrated Construction (CIC). CIC was conceived as a centralized, digital approach to assist contractors, including the logistical planning of resources (e.g. material supply, machine and operator assignment), the setting up of the machines on site, and the development of operator interfaces (e.g. planning and simulating robot actions and on-site robotic control) [YM98b, BA08, GBG00].

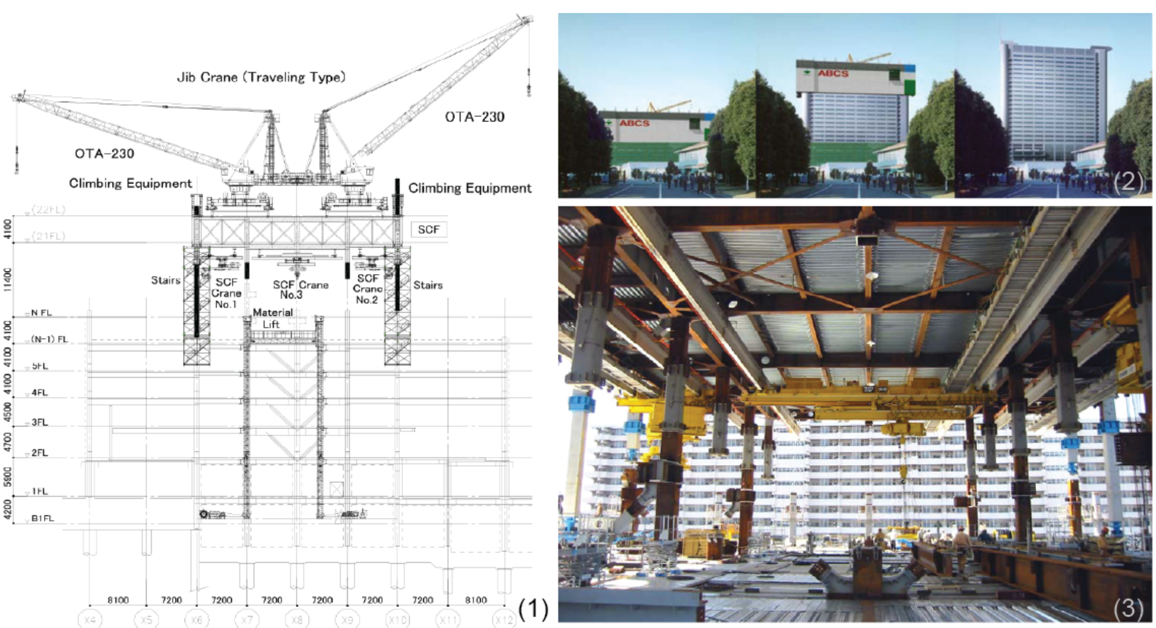


Figure 2-9: (1) Cross-section of the ABCS; (2) Conceived building procedure of ABCS; (3) Inside View of the Super Construction Factory (SCF). Images by [IH06].

As an alternative to stationary robotic systems, the EU Robotic Assembly System for Computer Integrated Construction (ROCCO) [GBG00] (Figure 2-10-(1-2)) and the Bricklaying Robot for Use on the Construction Site (BRONCO) [PDKG96] (Figure 2-10-(3)) were among the earliest projects that developed semi-autonomous robotic systems for the construction site. These large-scale robotic systems were conceived for the automation of block assembly according to a given building plan. Equipped with hydraulically actuated arms and an external laser tracking system for aiding the self-alignment of the assembled blocks, the prototypes of both systems demonstrated their abilities on a real-world scale. However, by the end of their research cycle in 1990, neither of them was commercially successful. Technical barriers included the complex handling of dimension tolerances and repositioning of the giant robot system on site. Furthermore, small or medium-sized construction firms, which make up 80% of the industry, were hesitant from investing the estimated amount of 250,000 EUR for a specialized machine designed to perform only one specific task - the layering of bricks [Ste02] (in German, summarized in English in [Tob15]-Chapter 3.2).

Figure 2-10: (1-2): the EU Robotic Assembly System for Computer Integrated Construction (ROCCO) [GBG00]; (3) the Bricklaying Robot for Use on the Construction Site (BRONCO) [PDKG96]. Images by corresponding papers.

Soon after its peak in the 1990s, R&D on robotics in construction declined. The explosion of the Japanese economic bubble played a big part in this, as the country was the largest driving force in developing robotic solutions for the building industry. In addition, the failure of success stories in robotic integration in the building industry left the big stakeholders and investors disillusioned, realizing that they were over-expecting. Eventually, both of these aspects lead to strongly reduced investment in research activities [BA08]. Meanwhile, researchers started to look at other domains to apply automation techniques, not limiting them to robotic solutions. Recent research published in ISARC and the journal of automation in construction features applications of all hot-topic technologies to the construction industry, including AR/VR, cloud-computing-supported management, blockchain-based bidding systems, and machine learning-based risk estimation and management systems, to list a few.

2.2.2 Design-driven construction robotics (2010-now)

The last decade has witnessed a renewed interest in the field of robotics in construction. The first wave of development was predominantly driven by construction engineers and contractors, where the architectural and structural designers had little say in the process. In contrast, in the current wave, the disciplines of architecture and design take on a fundamentally different perspective. The research focus is not on imitating and automating the manual construction process but instead to transform and rethink construction's relationship with design via digital tools. Together with the recent advancements in digital design tools, robots offer a media onto which new constraints and novel design logic are projected and derived.

In contrast to the 1980s to 1990s' R&D attempts in developing the machines themselves to perform specialized tasks, "the second go at robotic construction" utilized the versatility of standard industrial robots to achieve customized construction tasks that follow specific design intentions [Bec10]. These newly developed tasks can be broadly put into four categories: assembly-based, transformative, subtractive, and additive.

Figure 2-11 summarizes the 12 years of research in architectural robotics, with data drawn from major, related conference proceedings and journals: Association for Computer Aided Design in Architecture (ACADIA, annual), Computer-Aided Architectural Design Research in Asia (CAADRIA, annual), Education and research in Computer Aided Architectural Design in Europe (eCAADe, annual), Robotic Fabrication in Architecture, Art, and Design (RobArch, bi-yearly from 2012), Fabricate (tri-yearly from 2011), and the journal of *Construction Robotics* (founded in 2017). Due to the space limitations, a few key pieces of research from each category are summarized below, but this summary by no means covers the vast amount of literature published in the last decade.

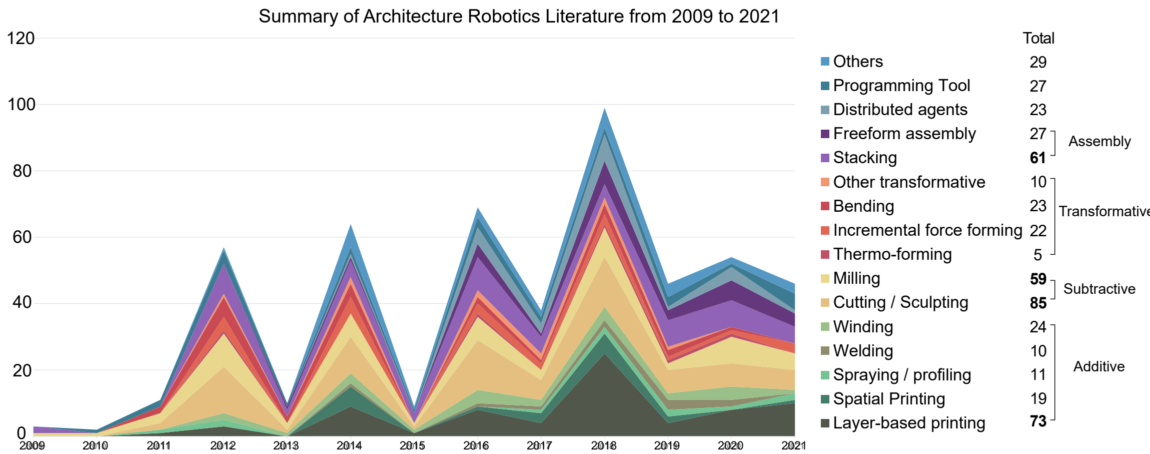


Figure 2-11: Summary of architectural robotics literature from 2009 to 2021. Contributions from *ACADIA*, *CAADRIA*, *eCAADe*, *RobArch*, *Fabricate* and the journal of *Construction Robotics* with keywords that contain "robot" are categorized and counted. The observed peaks in total publications that happened every other year starting from 2012 are due to the hosting of RobArch conferences. The data from ACADIA 2021 is missing because the proceeding is not published online at the time of this thesis's writing. The RobArch 2020 conference was canceled due to the global COVID-19 pandemic and thus not counted. Statistics graph made by the author.

Additive process Our classification of additive manufacturing (AM) processes is broad, encompassing all processes that transform and deposit materials from a spool or feed and solidify the material in designated spatial configurations. This is a more inclusive definition of AM compared to the more widely used, traditional definition of AM that almost only refers to 3D printing [GRS15]. Existing research in additive processes can be further classified into five sub-categories: layer-based printing, spatial printing, spraying, welding, and winding. Figure 2-12 shows representative projects from the four sub-categories below.

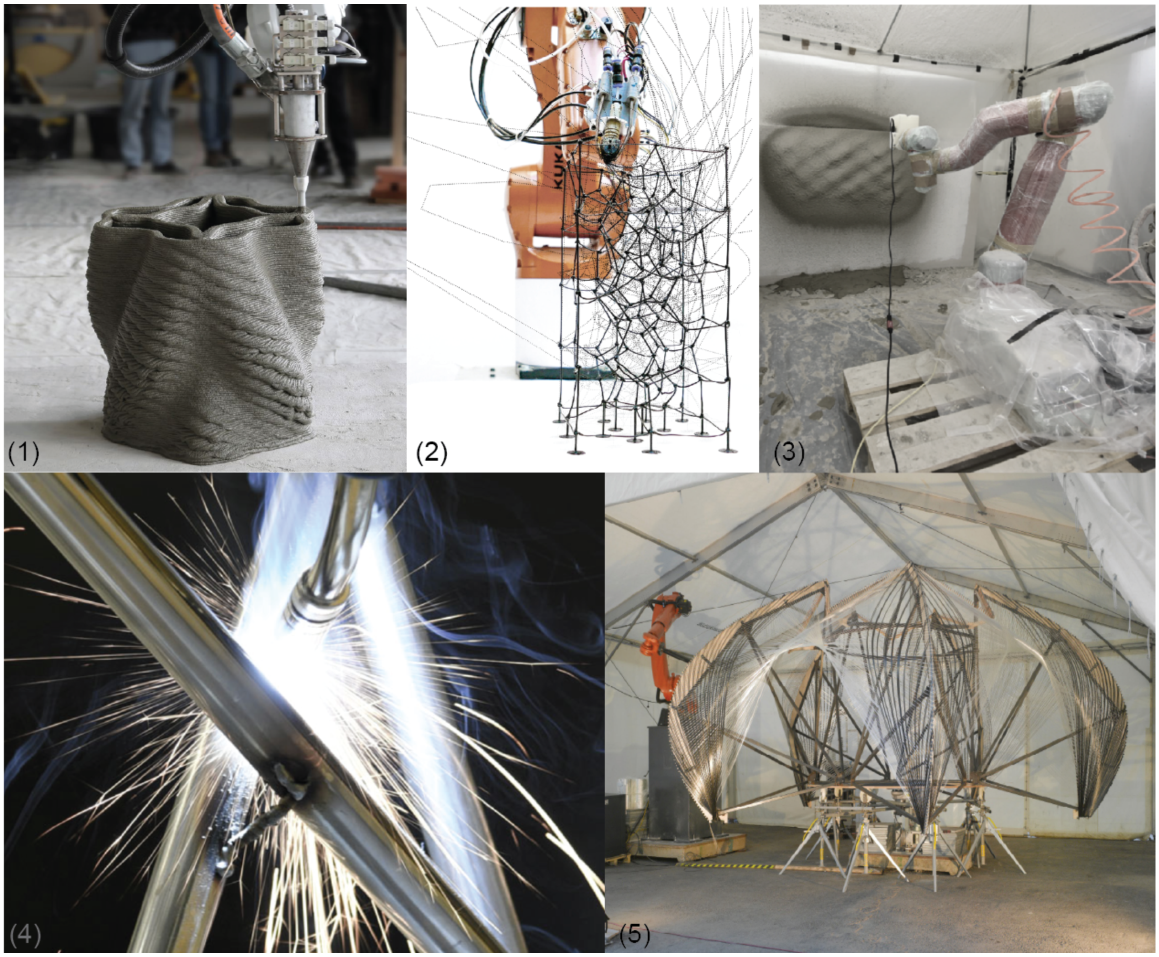


Figure 2-12: Examples of robotic additive processes. From left to right: (1) layer-based concrete printing [ABY+20]; (2) Spatial printing [HCTM18]; (3) Welding [AMG+18]; (4) Spraying [EJLFGK20]; (5) Winding [WSS+13]. Images by corresponding papers.

- **Layer-based printing** techniques create volumetric form by depositing materials through a nozzle in a layer-by-layer fashion. By mounting an extrusion head on an industrial robot system, the robotic printing system overcomes the size limit set by gantry machines. Existing projects explore the design-build potential offered by a direct translation from design to fabrication, facilitated by the relative ease of slicing a volume into layers, which can be directly translated to robotic programs. Existing research has explored a wide range of printing materials, including but not limited to thermoplastic elastomer [KO13, MVTT17, TCFM16], carbon-reinforced concrete [KEKD19], ceramic-clay [FKM14, RMS17, MB16], natural composite [DHV+18], foam minerals [BSZD22], glass [Web20, GD22] and concrete [BVMB18, AYB+19, GDB+19, ZZY21]. With proper reinforcement, multi-shell printed volumes are conceived to be used directly as walls from the early days of the technology's introduction to construction [Kho04]. Single-shell printed parts can be used either directly as final architectural products, or as molds for concrete casting [JBD+17, BLFS+20, BDZD21], or as parts of a multi-material print [KEKD19]. Recent research has shifted toward applications of mobile robots or cable robots to further expand the workspace [DDL+22, IDH+17], novel extrusion techniques in curved layers [MBD20a], and new design possibilities by applying more advanced geometric techniques [BVMB20]. Concrete, as a widely used building material, has been given special academic and industrial research attention. With the establishment of the Digital Concrete conference, recent research focuses on concrete mixture design, fresh state behavior, and mechanical behavior in the context of 3D printing concrete [BBCK22]. Layer-based printing is the second most researched category, summed up to a total of 73 publications from 2009 to 2021, second to 85 in the cutting/sculpting category.

- **Spatial printing** involves extruding material (e.g. thermoplastic or concrete) along linear or curved paths in space, typically to form a mesh or grid structure, using robotic motions. In recent years, robotic spatial extrusion has been presented as an alternative to layer-based AM for discrete truss structures, with advantages both in terms of mechanical properties [TMG+18] and speed of construction [OLK+13, HL14, Mat22, Bra22]. However, in most of the previous work, the flexibility of the robots has mostly been used to facilitate complexity in shape (as opposed to topology): morphed grids with repetitive zigzag topologies have been shown to be useful both for formal variation [HWT+15, YMYZ16, SLU17, CZY19, PM20a] or structural efficiency [TMG+18]. In many examples of this line of work, the robot follows a manually assigned zigzag end effector path with limited variation in the end effector direction. To enable spatial extrusion of a much broader class of truss topologies, recent research from the author and colleagues started to investigate the extrusion sequence and motion planning problem, introducing various planning techniques to find both the printing sequence and robotic motions [HZH+16, YHL+16, HCTM18, HGM18]. Thermoplastic extruded truss structures are conceived as moulds for bespoke concrete components [HL14] or foam walls [Bra22], pavilion installations [CZY19, Bra22, SLU17], small-scale protocols for studying complex sequence and motion planning involved in robotic assembly [HGT+21], and pedagogical tools for design [Gra15]. In addition to thermoplastic extrusion, we also put the slip-form concrete casting technique into this spatial extrusion category. Already a widely adopted casting approach in infrastructure construction, recent research in robotic slip-form casting focuses on utilizing the combinations of robotic movement and an actuated dynamic formwork mounted as an end effector to create complex concrete forms that cannot be achieved efficiently otherwise [LF16, YLX19, Sza20]. Printing in tank techniques are developed to overcome certain material's incapability to solidify on their own to stand in the air, with attempts in urethane rubber [HSG+17] and concrete [HDB+20].

- **Welding** refers to additive processes that involve steel as the production material, where the manufacturing process can be either layer-based or spatial. This sub-category is separated out due to its special material property and specialized machine setup requirements. Due to the rather expensive and involved hardware requirements, this direction is not as widely explored as the other categories of additive processes (Figure 2-11), limited to contributions from institutions that have collaborations with industrial steel production companies. Recent research looks into mobile robotic platform design for practical implementation on construction sites [DS18], Wire Additive Arc Manufacturing (WAAM) for creating spatial connection details [Ari22], and direct printing of volumetric metal structures in a layer-wise fashion [WDK14, FLE+20, MX322].

- **Spraying (profiling)** is one of the oldest envisioned application of robotic construction that can be traced back to SSR-1 (Section 2.2.1). Modern investigations focus on using robotic movement for creating visually appealing sprayed artifacts [BMS13], mud spraying from drones [CBVS20], and more systematic study of the complex-to-simulate plaster material behavior in this process [EJLFGK20]. In addition to plaster, shotcrete spraying, also termed shotcrete 3D printing, is gaining attention as a new way to fabricate lightweight spatial structures made by reinforced concrete [HLK19].

- **Winding** involves using robots to control the placement of fiber materials from a spool. Pioneered by developments from the Institute for Computational Design and Construction of the University of Stuttgart, core-less filament winding techniques have been explored and demonstrated in architectural-scale prototypes [WSS+13, PDS+14, PKD+15, DZB+19] In [VBD+15], an inflated pneumatic was iteratively reinforced through the robotic placement of carbon fiber, allowing the membrane to slowly transition into a stable compression shell. [RTB+19] proposes to use carbon-fiber wound webs as ceiling structures. Various mobile robotic systems are used for spatial winding to get around the reachability limitation of arm-based systems [Mir16, PE17, YM19, VFP+20]. Recently, a robotic fiber winding process was established, where a robot arm interfaces with a Cartesian machine to expand the traditional planar or surface-based winding pattern to a spatial configuration to form a spatial truss structure after curing [DEKW+20], with associated design strategy explored [EKWM20]. Wound fiber has been conceived to serve as structural stiffening to constrain the form of an inflated shell [PBSM16], formwork for thermoformed surface [Wan20], and formwork for shotcrete applications [SRLL20].

Subtractive processes use robots to control the removal of material, conceived as a direct extension from Computer Numerical Control (CNC) machines to extend their allowable dimension and fabrication flexibility. Subtractive processes can be largely put into two classes depending on the type of tools used for material removal: cutting/sculpting and milling. The robotic subtractive technologies are fairly mature these days, with extensive study from academia and successful applications in real building production. Representative works in this area are shown in Figure 2-13.

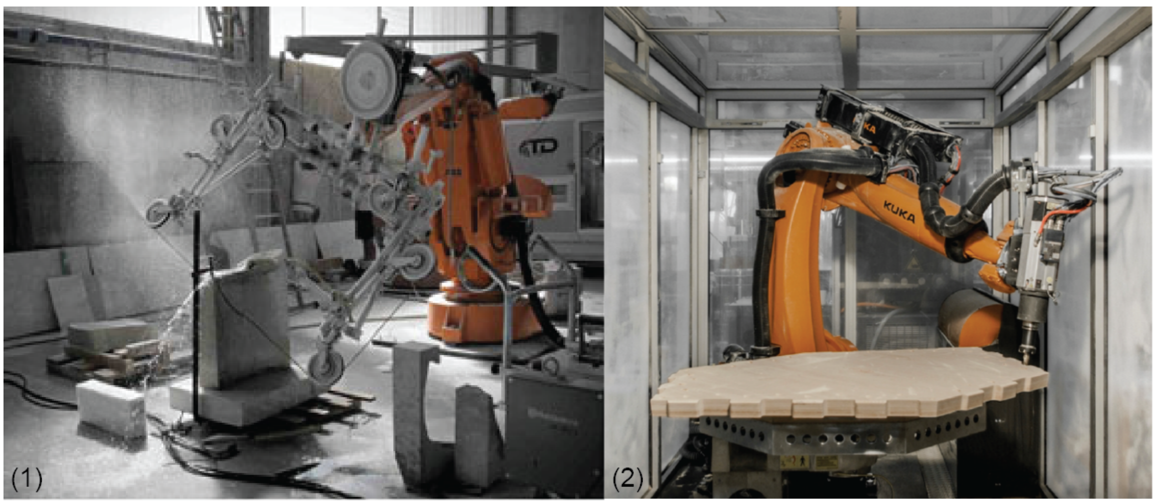


Figure 2-13: Examples of robotic subtractive processes. (1) robotic abrasive wire cutting [SFSM18]; (2) milling [KSM+15]. Images by corresponding papers.

- **Cutting/sculpting** processes involve procedures that separate a piece of material into parts. Existing developments involve equipping robots with various types of cutting tools, e.g. hot wire cutting, abrasive wire cutting, and knife cutting, to cut a variety of materials, e.g. foam blocks, clay, and stone. This is the most researched category, summed up to 85 publications in total from 2009 to 2021 (Figure 2-11). Single and multi-robot hot wire cutting has been studied in depth, including analysis of wire deformation during cutting and related design integration [SFN+16, Rus17]. Hotwire cut foam blocks are used as molds for casting non-standard concrete shapes or used as structural elements as part of pavilion installations [YMD14, SM14, WMF16, SMDAV16]. Due to the easiness of deployment in the studio, robotic hot wire cutting has been used extensively as a teaching tool for the education of robotic fabrication technology and related design thinking [BK13, BLPL14]. Robotic abrasive wire cutting (RAWC) is developed as an alternative to robotic hot wire cutting to avoid the hard-to-control sagging effect of the hot wire [SFSM18]. RAWC has been successfully deployed in real architectural projects, pioneered by the Danish construction robotics company Odico [SF17]. Robotic bandsaw cutting has attracted significant attention in its application in wood processing, used to produce ruled wood panels [WC16, ZLB+20] and trim raw wood log [SVM+17, LA20, AHM+20]. Additional applications of RAWC include freeform ceramic brick production [RWS+21, AB17]. Robotic cutting has been applied to process harder materials like stone [ASD+17] and steel (e.g. plasma cutting [DKR+20]). Robotic clay sculpting is conceived to produce cheap casting molds [SP13a] and stylized artistic objects [DHM14, MDS+21]. Similar sculpting techniques have also been applied to achieve complex concrete surfaces [BCWZ18] and geometric patterns on foam columns [CELM14]. Robotic chiseling is conceived to automate the production of stylized surface finishing, to compensate for the rapid decline of craftsmanship in this longstanding historical technique. Existing works develop adaptive control strategies (preprogrammed or machine-learned) to achieve the desired finishing effect, including attempts on sandstone [KTT20], stone [SBB+16, SBSS20] and wood[BH19]. Recently, driven by the demand for renovation of existing buildings, research starts to investigate using robots to disassemble old wall plasters [LHVBC17, DCEF+17].

- **Milling** processes remove material by equipping robots with a spinning drill bit or wheel. Applications of robotic milling technology showed up in almost all wood-related robotic fabrication projects. A summary of major projects that involve robotic milling in recent years can be found in [MSK16, Wei16]. As an almost mature technology, robotic milling processes have been successfully applied to wood constructions of impressive architectural scale [WAGM20, LLB+20]. Robotic milling enables the fabrication of custom, complex wood panels and joints [KSM+15, Pag17, TKH19, JP20, RLW21]. For some professional carpenters, the fabrication time and cost efficiency of robotic milling promote a renaissance of crafts that are hard to be reproduced manually today [Sch20]. While milled foam formwork for casting (e.g. [KZN+20]) is seen by many as too time-consuming for practical applications, custom formwork that only requires a few milling actions have been devised [dAVS18]. Milled detailed geometry can be used to enhance acoustic performance [RCJW16]. Recent research looks into more intuitive control interfaces for carpenters [PSR21], AR-based design-fabrication integration [HOL21], and an automatic workflow that connects raw-wood scanning, automatic joint generation and robotic milling path planning [Ves21].

Transformative processes involve using robots to manipulate materials into different shapes by various physical processes without adding or removing materials. Existing works include thermo-forming, sheet force-forming, bending, and other more specialized processes. Figure 2-14 illustrates representative works in this area.

- **Thermo-forming** processes involve using robots to apply heat to melt the plastic surface, so it can change shape locally or globally. Existing efforts include using robotically wound formwork to aid the thermoforming process [Wan20] and using a robot to heat the surface locally and use air pressure to inflate and mold the surface [SJ19]. However, it is hard to control the precision of these thermo-forming processes, and thus existing fabrication results are all fairly small, limited to model and pavilion demonstrations.

- **Sheet force-forming** uses one or multiple robots to apply pointed force to locally deform sheet materials, creating complex trace patterns. The most prevalent and precise method is the double-point incremental forming that

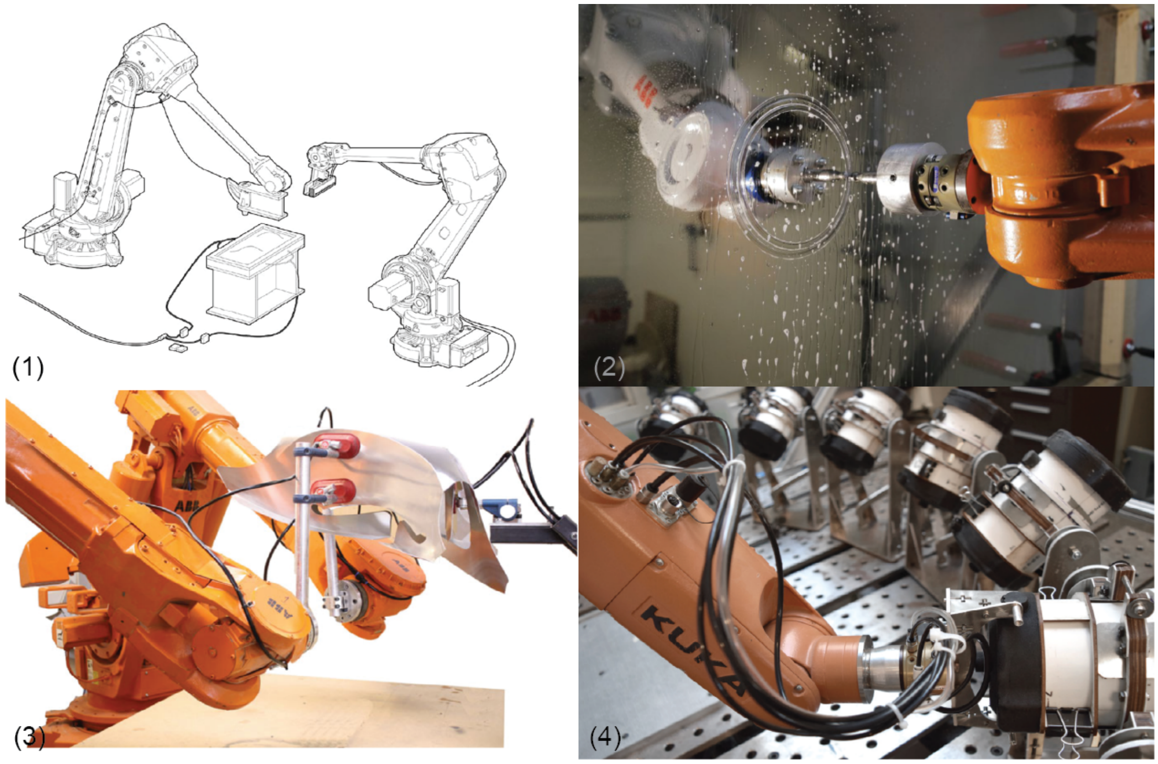


Figure 2-14: Examples of robotic transformative processes. (1) thermo-forming [SJ19]; (2) sheet force-forming [NZN+17]; (3) bending/folding [SE16]; (4) other transformative processes: e.g. reconfigurable concrete casting [THM19]. Images by corresponding papers.

involves a pair of robots, one for supporting and then the other pushing [BKK13]. This technique has been shown to have the ability to create parametric surfaces with great complexity [KN14]. Single-point incremental forming uses a single robot to push on a sheet of material that is fixed on its boundary [TPBVM21, Wei17]. Compared to double-point incremental forming, this technique is easier to set up but is harder to precisely control the results. [CDPR20] uses this technique to create space-filling curves. Detailed finite-element analysis has been performed to obtain precision of the forming production [NSN+16]. [NCSS21] investigate macro and micro tool pathing strategies to counteract the buckling effect during forming for creative repurposing of reclaimed metal sheets. Materials that are less ductile than metal, like plastic panels, can be pre-processed to make the force-forming process easier, e.g. laser-cut [Wei14] or heated [ZWT21]. [RAP14] uses robotically pressed clay to make reusable formworks. Tool profiles other than the most widely used ball-like tool have been experimented with, e.g. bead [FHL14] and wheel [RN18]. A variation of robotic force-forming involves using CNC machines to pre-cut patterns and the pressed by controlled robotic motions to form parametrically designed, nuanced surface patterns [KG13, BBC12b]. This technique has been applied in practice to create customized facade panels on an amazingly big scale[2].

- **Bending** processes include having robotic arms directly bend bars [SJ16] or interact with CNC wire benders [PM11, PMM+13, FGS17]. Plastic bars are usually preheated before bending can be applied [LWS+20, GHRV18]. [COSS13] use two bending robots to create a differentiated reinforce rebar network. [BLPL14] discusses bending and folding of aluminum sheets or acrylic rods for high-rise building model making. Robotic curved-crease-folding techniques are explored by pre-cutting metal sheets and then having one or multiple robots fold them [LVK11, SE16, WGZ+19].

- **Other transformative processes** include various new fabrication processes that cannot be grouped into the three sub-categories above. [DD13] shapes iron-based material by robotic manipulation of magnetic fields. [KTGW14] deforms polymer during its curing process. Various techniques of robotic manipulation of concrete casting formwork have been researched, including robotically manipulated fabric [CKS16, YLL19], robotically pushed pin-fields [APTP16, VS18, FT14] and re-positionable cylinder casting devices precisely manipulated by a robot arm [THM19].

Assembly processes use robots to pick and place building elements into specified positions. When elements are not identical or custom connection joints need to be fabricated, these assembly processes are often coupled with cutting/milling to produce the required geometries before assembly. Existing work can be categorized

[2]For example, see works from Zahner: https://www.azahner.com/

into stacking and spatial assembly, depending on whether a layer-based building sequence based on height can be applied to the assembly processes.

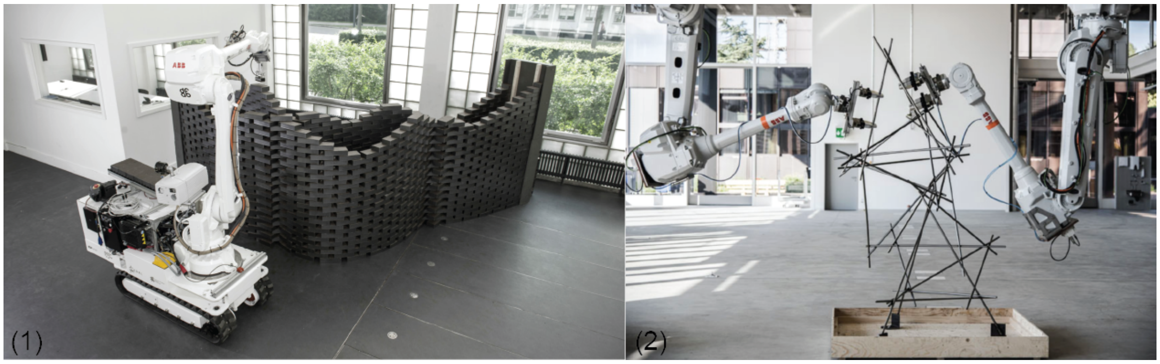


Figure 2-15: Examples of robotic assembly processes. Left to right: (1) brick-stacking by a mobile robotic platform [Dö18]; (2) multi-robot assembly of steel structures [PGM+17];. Images by corresponding papers.

- **Stacking** includes processes that the robot places elements in a layer-by-layer fashion, where elements can be connected either by pure contacts and friction or other types of connection details (see Section 2.4.1 for different types of joint details). To many, the first robotic brick wall stacking work, published in 2006 by Bonswetch et al., marked the start of the modern investigation of architectural robotics [BKGK06, BK07]. This work later grew into a complete set of work that investigates the relationship between parametric design and robotic brickwork [Tob15] and eventually led to the founding of a practice specialized in automated solutions for bespoke brickwork[3]. Because the reachability space of robots is usually much bigger than the objects being assembled, path planning problems in stacking processes can usually be solved by design-specific heuristics (see Section 2.2.2 below). Thus, the technical barrier to using robotic stacking to realize complex design prototypes is relatively small, making it the 4th most popular category of research, in a total of 61 papers from 2009 to 2021 (Figure 2-11). Robotic stacking has been used as a protocol to study how to open up the fabrication process to allow design intervention [DRR13, Jef16, SBT16, JA18] and a pedagogi-

[3]ROB technologies: https://rob-technologies.com/

cal tool for teaching parametric design and basic robotic fabrication principles [BK13]. The robotic stacking technique can enable detailed control of a massive number of identical elements to be organized in a global way to give macro artistic expressions [KS13, AKGK16, PJP+19, XLG19]. Custom-made grippers and joining machinery for automation have been investigated [VVB+14, NKKS21]. Recent investigations look into incorporating various vision components to enable a close-loop control strategy to enhance the robustness of the process [DH19, TKC20, CYJ21]. Mobile robot platforms, associated sensor techniques, and integrated design strategy for brick stacking have been studied [Dö18]. In-situ construction of dry-stone by an autonomous hydraulic excavator is researched, combining on-the-fly identification and structural planning of newly found stone elements [JWM+20]. [DSM13] studies robotic pouring of spiky elements, which form into aggregations with self-organized stability. [AL21] proposes a fabrication process that combines granular jamming with strategically placed, continuous reinforcement for precise and reusable fabrication of jammed structures.

- **Freeform assembly** concerns the robotic assembly of more freeform, spatial structures that do not necessarily follow the layer-by-layer construction sequences as in stacking. Robotic spatial assembly usually uses standard elements that are cut to custom length and utilizes the overall configuration of these elements to achieve the intended complexity, instead of relying on the diversity of each element. In these cases, the structure being assembled creates a populated workspace for the robot to maneuver within, so the planning problem becomes much harder [DHS+19]. Spatial assembly gives new light to constructing structurally efficient, geometrically complex structures that are hard to build manually [SAE+16]. Integrated robotic solutions that automate the cutting, drilling, holding, and assembling of timber elements have been demonstrated [EGK17, TAH+19, XHP20]. Beyond the dominant timber and steel bar systems, there are attempts in timber plates [RWH+17] and other more specialized element design [BA20, Cla20]. Connections between elements become especially important in spatial assembly because of the heterogeneous spatial configurations of the elements. Design, analysis and fabrication methods have been researched in-depth for steel [Ari22] and wood [WKB+16, KAGK17]. [LAT+21] devised remotely-controlled, high-force robotic clamps to cooperate with a robotic arm to overcome challenges of robotic timber joint assembly, such as providing large assembly forces and correcting misalignment. [PGM+17] investigates a multi-robot assembly system that uses one robot as a structural support agent and the other as the assembly agent. The mutual constraining relationship between the design of a spatial truss system, its structural performance, and the existence of a feasible assembly sequence and robot trajectories is researched in [Par19], leading to sequence-driven design logic. Similar cooperative assembly-and-support strategies are applied in the construction of full-scale vault structures without any scaffolding [WK18, PHW+20]. Applications of motion planning tools to mitigate the problem of finding collision-free robot trajectories are researched in [GPR+18]. [Gan20] investigates the impact of assembly sequences on construction tolerances. The "design-as-you-build" process investigates how the geometric collision constraint of the robot influence a sequential design process [BTAP20, AMS+20]. An iterative design strategy based on robotic stochastic assembly and a learning-based vision system has been developed to generate spanning structures [WK19]. [BAP22] uses graph rigidity theory to identify stable assembly and disassembly sequences for dual robot construction without scaffolding. Other works investigate equipping the robot with tactile sensors for more intuitive robot programming [SB19] and enabling human-robot interaction during construction [WBS+19].

Other processes include robotic fabrication processes that cannot be put into the four main categories above, e.g. robotic needle felting [MNP19], weaving [BBVM16], stitching [SKM16], wrapping fiber-reinforced polymer for formwork [ZOL21]. Beyond fabrication, architectural robots are used to conduct daylight study [KPB+14], augment 3D projective video making [Öz14, Pou19], and interactive installation [Pia14]. See Figure 2-16 for a sample of works.

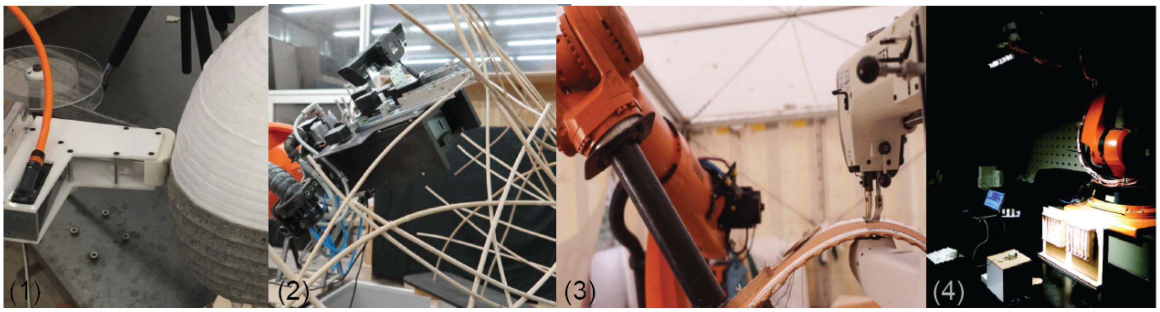


Figure 2-16: Examples of distributed robotic processes. Left to right: (1) felting [MNP19]; (2) weaving [BBVM16]; (3) stitching [SKM16]; (4) robotic platform for daylight studies [KPB+14]. Images by corresponding papers.

Distributed agents Compared to dominant robotic arm systems that are designed and built to mimic human, distributed systems offer a new paradigm through organized construction behavior of many much smaller, agile and low-payload agents. Existing investigations include termite-inspired robots making brickwork collaboratively [WPN14], aerial robots for tensile structure construction [Mir16] and brickwork [WAC+12], UAV teams that transport specifically designed panels to construct pavilion [WYA+19], a heterogeneous robot team for filament winding of spatial tensile structure [YM19], and robotic swarms that fabricate by each pulling fiber and resin and winding around its own body to build spatial tubular structures [KCB+19]. Recently, learning techniques are proposed to enable mobile robots to manipulate natural materials with variable properties [KŁH+22]. Light-weight robot teams are conceived to interact with indoor fabric to create an adaptive textile system [WYAM20]. [LWW+19] proposes distributed system that uses the timber material itself as part of its locomotion system. [JARCG19] proposes distributed robot systems that can transport and place cuboctahedral unit-cells. Cable robots are conceived as a higher-payload and efficient alternative for arm-based systems for construction tasks with aerial clearance [SPM+16, DIC+19, CCC+18]. More speculative work envisioned that a robot swarm itself forms a reconfigurable architecture [AJNP16]. [Mel17] proposes a planning strategy for agents to infer the global structural state from local force measurements.

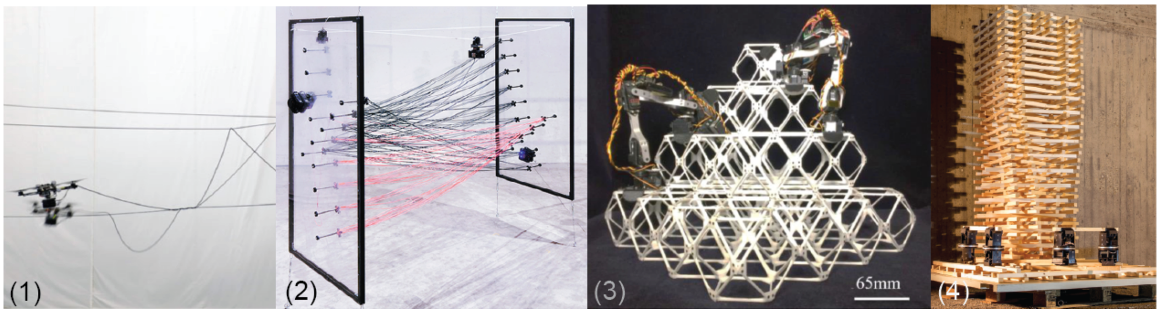


Figure 2-17: Examples of distributed robotic processes. (1) aerial robot winding [Mir16]; (2) winding [YM19]; (3) discrete cellular structure [JARCG19]; (3) timber assembly [LWW+19]. Images by corresponding papers.

Programming tools In all of these robotic fabrication processes above, researchers encounter a similar programming problem: how to generate a sequence of commands to control the robot to achieve certain tasks while ensuring that it do not collide with itself and objects in the workspace. Modern industrial robots come with vendor-specific programming languages to write machine code to control their behavior. These programming languages (e.g. the KRL language for KUKA robots and the RAPID language for ABB robots) often feature built-in commands like Linear movement (LINE or MoveL) that instructs the robot to follow a path that keeps its end effector in a straight line in the workspace or Joint movement (PTP or MoveJ) that moves the robot in the shortest path in the joint space from its current configuration to a target joint configuration. Although these programming languages are regarded as "high-level" by the robot manufacturers since users do not need to worry about the low-level torque control, the offered programming paradigm does not guarantee to generate collision-free robot commands. In contrast to Cartesian machines like 3D printers, where there is a one-to-one correspondence between the machine code and machine behavior, the interpretation of robot programs is not that straightforward, because of the complex kinematic chain involved in these high degrees of freedom manipulators. The program interpretation depends a lot on the inverse kinematics and path interpolation routines implemented inside the industrial robot's controller, which are not exposed or well-explained to the users. This means that the users must rely on vendor-provided simulation tools to visualize and check the entire, continuous path behind the sequence of robot commands specified. Under this paradigm, a typical programming process for a generic robotic task usually takes the following steps:

1. Decide on a sequence of high-level robotic actions to be performed.
2. Specify path points (also called target positions) for the robot's end effector to trace for each robotic action.
3. Weave in IO commands for controlling the end effector (e.g. open or close the gripper, turn on or off printing extruder) in between the key path points.

There are a lot of programming tools developed in architectural robotics to simplify the design-to-robot-program process following the steps above, by embedding tool path simulation directly into the parametric CAD environment like Grasshopper3D [Rob22]. These tools can be robot-vendor specific [Paw13, BBC11, WSF22, Com22, EG14] or support arbitrary robot models [Sch12, CyL19, Sol22b]. Many of these programming tools have abilities to directly upload the generated program to the controller, further expediating the execution cycles. Other research aims to augment these programming tools with the abilities to control and interface with peripheral devices, e.g. custom-made end effectors controlled by Arduino board [BBC12a] or vision devices [SAS17, LZZ+21b] Alongside these generic programming tools, there are also programming tools designed for specific fabrication processes, for example, milling [Sim16], brick stacking [BBH13], high-rise building model making [LGK13], and robotic spatial extrusion [SRJG17].

However, because these programming tools are following the "program workspace path points first, simulate second" promoted by the robot vendors, much extra effort is needed from the users to dry-run their programs and make sure the execution does not contain collisions, out-of-range joint values, and joint singularities. The users often need to use ad-hoc heuristics to devise planning tricks to address these problems, like generating a guiding curve or inserting reset configurations to help the robot unwind itself or hover away from collisions (e.g. see [SAE+16, EGK17]). While these ad-hoc planning strategies are technically feasible to be applied to designs and fabrication processes with a repetitive pattern, for example, spatial extrusion of lattices with layers of zigzag patterns [HL14, HWT+15, Bra22], planning the robot motion is much more nuanced for designs with arbitrary topologies. These ad-hoc solutions do not have any guarantee of the feasibility of the generated programs and are hard to generalize beyond the targeted design typologies. This slow and cumbersome planning workflow deviates from the initial purpose of having such a design-fabrication workflow: to forge a smooth and direct transition from digital design to real-world machine materialization; instead, the current process often requires a complete re-program for the robot whenever the target geometry has a small change.

Among all the fabrication categories summarized in this section, planning for spatial assembly is the most difficult. This is because the in-progress assembly creates an increasingly populated space for the robot to maneuver within, and ad-hoc planning strategies need to be devised separately for each of the hundreds of elements. This challenge motivates some attempts in the community to use motion planning algorithms to automatically find collision-free robot trajectories [PGM+17, Gan20]. However, assembly planning involves solving the construction sequence and robot motions simultaneously, which is a hybrid, discrete-continuous search space that is very hard to explore computationally. Existing solutions usually separate the sequence planning and motion planning, creating a rigid planning hierarchy that cannot backtrack if failures are encountered in the later planning stages [HZH+16, HCTM18, HGM18].

Alongside the programming tools that are designed for generating programs before the execution, other paradigms of human-robot interaction are proposed, e.g. haptic programming [SB19] and proxy for teleoperation [Pay11]. New control tools for teleoperation of existing construction machines are proposed, e.g. demolition machines [LBC21]. New network protocols are proposed for communications between robot controllers and external clients [Sch15] and for general network infrastructure on construction sites [DDB21].

2.2.3 Reflections

Robotics-based construction promises to substantially improve construction in the built environment, offering benefits such as speed, quality, material efficiency, worker safety, and eventually cost reduction. In addition, robotic construction opens up new fabrication possibilities beyond automating manual processes, leading to an expansion of design possibilities and formal expressions. Recent years have finally started to see these decade-old promises come to fruition, primarily in the form of impressive prototypes, proof-of-concept, and demonstrator projects at the "pavilion" scale, showing the potential of robotic fabrication and assembly in architecture. On the other hand, new businesses have started to emerge that propose robotic solutions for construction, for example, the robotically 3D printed freeform building facades by Branch Technology[4], the automated construction planar roof truss by House of Design[5], and robotic assembly of rebar cages by Toggle[6]. Addressing the planning challenges is an important step toward bridging the vast geometric possibilities demonstrated in academic projects and their adoption by the industry. As detailed in Section 2.2.2, the currently prevalent way of programming the robot for the fabrication of bespoke geometries involves hours of human labor that is not scalable and cost-effective. Although this ad-hoc, heuristic-driven planning paradigm is already a step forward from the manual, planning-by-teaching paradigm that is widely adopted in the automated manufacturing industry, an automated planning paradigm is a key for expanding the relevance and impacts of these technologies on the construction site.

[4] https://branchtechnology.com/
[5] https://thehouseofdesign.com/
[6] https://www.toggle.is/

By categorizing work in architectural robotics in the past 12 years, we see a clear trend of research focusing on four fabrication categories, with the cutting and sculpting category topping with 85 papers, followed by layer-based printing (73 papers), stacking (61 papers), and milling (59 papers). For these top four categories, effective heuristics exist to solve the robot programming problems involved. We believe the under-development of freeform assembly (only 27 papers) that is critical for the production of high-performing long-spanning systems, is due to the technical challenges of the assembly planning problems.

If we look at the development history of other, more mature digital fabrication technologies like Computer Numerical Control (CNC), we observe a similar development trajectory that starts from manual planning, then ad-hoc planning, and finally automated planning. Figure 2-18 illustrates the research plan outlined by the early CNC pioneers at MIT, where the manual labor required in programming the machines is gradually reduced by the invention of more advanced, automated planning algorithms [Lla15]. Almost 55 years after this proposal, originally developed for wartime manufacturing needs and heavily sponsored by the military, automated CNC machines have become the foundation of the modern manufacturing industry and later become easily accessible in small-scale fabrication shops around the world. Augmented by the development of 3D printers in the last decade, designers are equipped with powerful fabrication technologies to make customized parts with an ever-shortening time between design and fabrication. The software that converts the input design geometry into detailed, executable machine code, e.g. the slicing software for 3D printing, has played an important role in such design-to-fabrication smoothness. This thesis develops automated planning techniques for robots to re-create such success in automated construction and aims to open up new design-build potentials that have never been shown possible before.

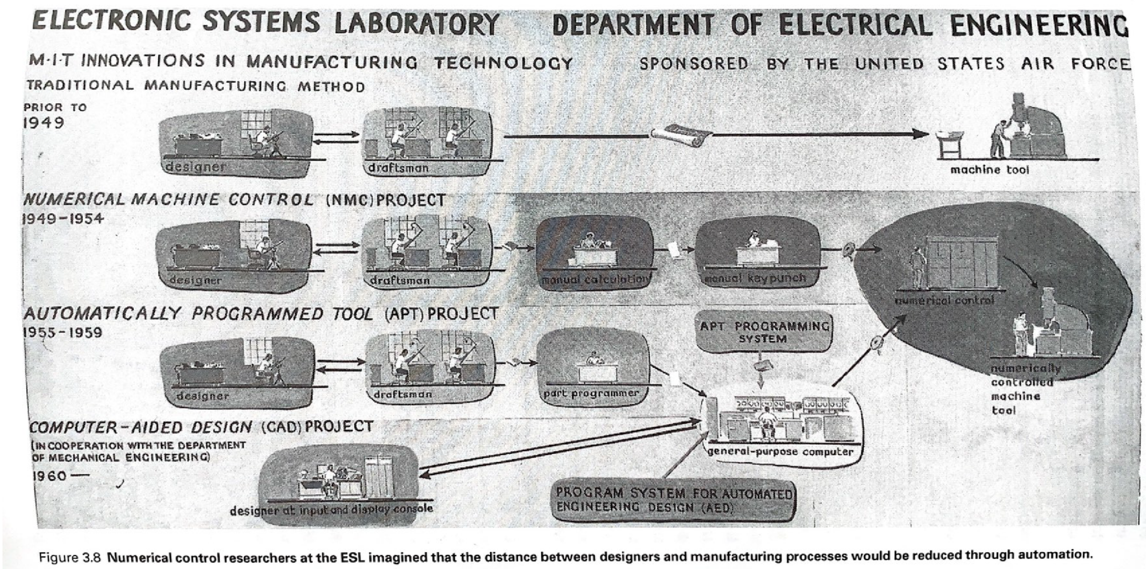


Figure 2-18: Research plan envisioned by the early research pioneers at the Electronic Systems Laboratory from the department of electrical engineering at MIT in the 1950s. The plan projected three research stages that go through manual planning, ad-hoc, case-by-case planning, and automated planning, gradually shortening the distance between designers and manufacturing processes through automated planning techniques. This image is scanned from [Lla15], with the original print from MIT Archives, Douglas T. Ross papers, MC 414, box 220.

2.3 Design and analysis for manufacturing

The exploration of the relationship between designing and making started from the first day of humans building tools and artifacts. In contrast to the computational age in the last 40 years when much knowledge is embedded and codified into algorithms and software, most of the historical knowledge about designing and making is passed down across generations of craftsmen and -women by training apprentices via visual examples and oral directions, with little written record[Sen08, Fit89]. When new material and fabrication technologies emerge, it took individual ingenuity to expand the boundary of existing design stereotypes. The reciprocal relationship between new fabrication techniques and design languages is best exemplified by the design evolution of chairs (Figure 2-19). For example, the lightweight Thonet bentwood armchair comes out of the German furniture maker Michael Thonet's mastery of wood steaming and bending techniques

(Figure 2-19-2). The modernist era featured sleek, cantilevered designs made possible by the advancement in steel manufacturing (see, for example, the tubular Brno chair by Mies Van Der Rohe that features a sinuous tubular steel frame that makes its guests appear to defy gravity Figure 2-19-2). Later, advances in plastic and fiberglass technology developed during the Second World War made Charles and Ray Eames's famous DAR design possible.

Figure 2-19: Design evolution of chairs. Left to right: (1) American Queen Anne side chair, John Elliott, 1750. Image from 1stdibs.com; (2) Thonet bentwood armchair, Michael Thonet, 1830. Image from 1stdibs.com; (3) Tubular Brno chair, Ludwig Mies Van Der Rohe, 1929. Image from Knoll. (4) DAR, Charles and Ray Eames, 1948. Image from vitra.com. (5) Masters chair, Philippe Starck and Eugent Quitllet, 2009. Image from flinders.nl.

In construction, the intertwined trajectory of design and manufacturing capabilities can be observed by looking at the evolution of structural joints (Figure 2-20). Starting from the intricate integral wood joints made by Japanese craftsmen in the pre-industrialization era (Figure 2-20-1), connection details evolved from massively producible, standardized die-casted ball joints (Figure 2-20-2) to more complex, bespoke joints made by 3D printing (Figure 2-20-(5,6)).

In mechanical engineering, design for manufacturing and assembly (DFMA) aims to establish design procedures and rules-of-thumb to guide designers and engineers to include assemblability and cost considerations in their design process [BDK10]. Although these principles are useful especially for inexperienced designers, their adoption in practice leaves too much flexibility and the evaluation of final design outcomes is subjective, depending heavily on the engineer's experience. Recent advancements in this field go toward further codifying some of these aspects

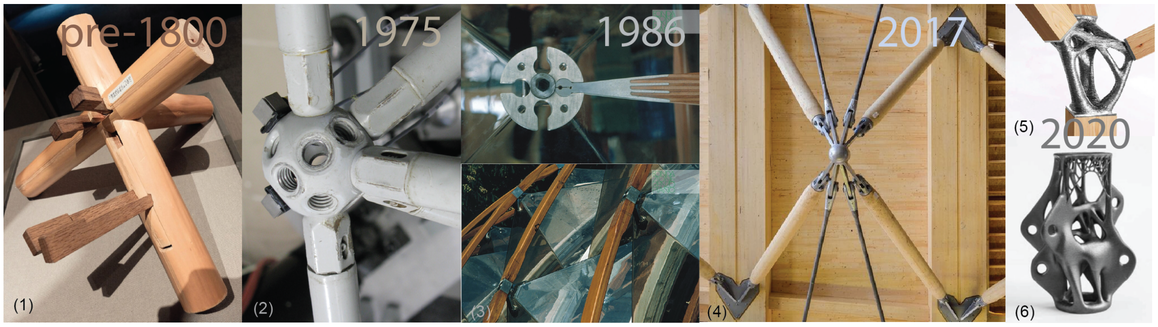


Figure 2-20: Design evolution of structural connections : (1) traditional Japanese wood joinery displayed in the Takenaka Carpentry Tools Museum. Image from [Bro16]; (2) Mero node system [Men75]. Image from [SJ11]; (3) Node details from the IBM Padiglione Itinerante for IBM's traveling exhibition. Image from [Fou17]. (4) Connection details of zipper trusses at the John W. Olver Design building. Image from [CAS]. (5-6) Robotically fabricated Takenaka connector [MX320] and 3D printed connector [GHR15].

into computational algorithms. Such codification not only offers an objective and quantifiable way to evaluate a design's performance concerning manufacturing but also opens up the possibilities to connect such evaluations to an optimization routine to find better-performing designs. In this section, we provide a short overview for recent work in design for additive manufacturing (Section 2.3.1) and assembly planning of mechanical parts (Section 2.3.2).

2.3.1 Design for additive manufacturing

Additive manufacturing (AM) enables the fabrication of complex geometries that are not possible with traditional machining techniques. This has raised the need for new design methods that can leverage the new, increasingly complex manufacturing possibilities [TMV+16]. Much research effort has come into formulating and quantifying the constraints that come with AM (see Section 2.4.2.1 for details). Topology optimization (TO) is often suggested as a design-for-AM method since it has the potential to generate new, high-performing design solutions. It is a free-form design approach that does not require a pre-conceived notion of material layout, and the resulting solution is obtained by running a numerical optimization process minimizing target objectives like stiffness or volume, subject to certain constraints. Existing research in combining TO and AM includes eliminating overhangs [GG16], allowing design with multiple materials [WT16], designing porous infill [WAWS17], addressing nozzle size restrictions [Car20], etc. Readers are referred to [LGC+18] for a comprehensive review of work in this area.

2.3.2 Assembly planning for mechanical parts

There is a large body of work in assembly planning of mechanical parts, dating back to the 1980s and 1990s with the influential work by Homem de Mello [DMS90], Wilson [Wil92], and others [DFW87, Wol89]. The primary concerns in these works were to find an assembly sequence from a given CAD file that satisfies low-level constraints such as mutual blocking relationship during assembly, mating constraints, and assembly tolerances. The methods involved focus on the product itself, without considering the constraints brought by operators that perform the assembly (human or robot). This might be a valid assumption for this domain since the environment and assembly robot can be specifically engineered for the task.

The key contributions in the area of classic assembly planning include: (1) the mathematical formulation of assembly planning problems under certain assumptions (2) the proposal of several compact and efficient representations of intermediate assembly states and (3) theoretical characterization of the computational complexity of the involved problems. Interested readers are referred to [Jim13, GM15, BB16] for a more detailed and extensive review. However, most of the literature in this domain considers only geometric constraints: an assembly plan is defined as valid if there is no collision when assembling each part. Other constraints that are especially important for robotic assembly in construction, like collision constraints that ensure the existence of robotic assembly motions and structural constraints that limit deformation or stress, have been ignored in these works. This thesis shows that new formulations and planning techniques need to be proposed to address the new challenges brought by these additional constraints.

2.4 Computational assembly design and analysis in computer graphics and human-computer interface

The computer graphics research community has extended research in computational design and analysis methods for discrete assembled structures and their fabrication. In this section, we provide an overview of work in this area, following the general taxonomy of literature presented in [WSP21], which provides a recent, more detailed review on design and fabrication methods for complex assemblies in graphics. The literature in this area has a lot of potential impacts on architecture and construction. I believe that these potentials have not been fully realized due to a lack of community overlap. This section is an attempt of bridging this gap by providing a critical review of key research.

This section is organized into two parts: Section 2.4.1 for computational analysis for assemblies which makes the foundation for quantifiably evaluating certain performances related to an assembly, and Section 2.4.2 for computational design strategies for assembly. We end the section by providing some reflections on the relationship between this research and architectural design and construction.

2.4.1 Computational analysis of assemblies

Given an assembly with part geometries, computational analysis methods evaluate various aspects of the assembly, including joint analysis, assembly planning, structural mechanics of the whole assembly and efficiency for packing the parts.

Joint analysis A joint provides the connection among elements so that they can behave as a system when external loads are applied. Joints can be classified as permanent and non-permanent joints. Typical permanent joints include adhesive materials (like glue and mortar), or mechanical connectors (like screws or rivets). Widely used in load-bearing structures, permanent joints provide guarantees of structural stiffness. In contrast, non-permanent joints enable easy disassembly and reassembly and facilitate easy storage and maintenance. Non-permanent joints can be categorized into external joints and integral joints, depending on whether the joint geometry is integrated into each individual part. Figure 2-21 provides examples for each of these joint types. To satisfy the needs of joining specific parts, customized, 3D printed external joints have been designed [MMZ18]. More research attention has been turned to the integral joints, which have a long tradition in wood joinery [Fai13].

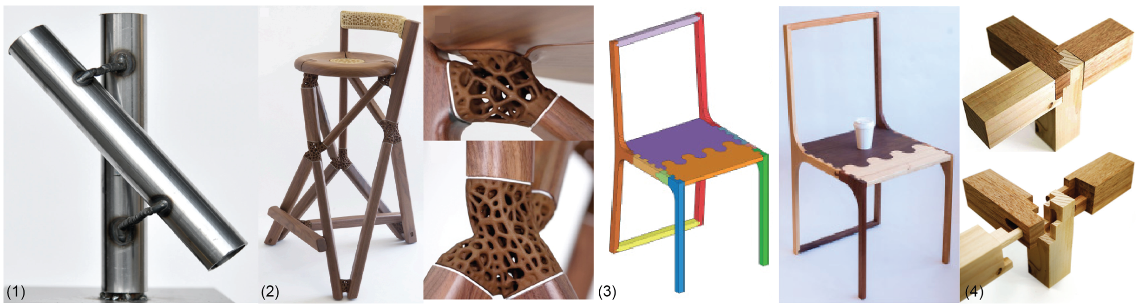


Figure 2-21: Examples of joint types. Left to right: (1) permanent welded joint [AMG+18]; (2) non-permanent, external joint [MMZ18]; (3-4) non-permanent, integral joint [YKGA17, LYUI20]. Images by corresponding papers.

The joint analysis attempts to computationally identify if the joints facilitate connectivity among parts via a part graph [FSY+15], relative mobility among adjacent parts by solving infinitesimal motions through a linear system [WM92], or analyzing joint strength by finite element analysis [YCXW17].

Assembly planning Complementing the research efforts from the mechanical assembly planning literature (Section 2.3.2), graphics research in assembly planning aims to go beyond finding a feasible assembly sequence for a better-performing or user-preferred assembly plan. Figure 2-22 provides some examples for research in this direction. Various practical algorithms are developed to find sub-optimal solutions, e.g. using various heuristic-driven greedy algorithms [APH+03, DPW+14, MSY+15, WPGM16, HZH+16]. Agrawala et al. presented a computational framework to automatically generate assembly instructions, given assembly object geometry, orientations, and optional grouping and ordering constraints on the object's parts [APH+03]. Moving towards assembly on an architectural scale, Deuss et al. studied the physical construction of self-supporting structures with masonry blocks by leveraging external cables to help identify self-supported partial constructions [DPW+14]. Computational assembly strategy of reciprocal frame structure [MSY+15] and disassembly strategy of masonry structure [BBW15] has also been investigated.

New research questions related to assembly planning arise when people try to augment traditional layer-based 3D printers with more degrees of freedom. WirePrint [MIG+14] proposes an efficient way to print wireframe meshes, where edges in the mesh are directly extruded in 3D space. Compared to traditional 3D printed objects, WirePrint generates the wireframe of a model by slicing it horizontally and filling each slice with zigzagging filaments. This approach constrains the types of meshes that can be printed. To improve flexibility, Peng et al. introduced a 5-DOF printer, which modifies a standard delta 3D printer by adding two rotation axes to the print bed [PWMG16]. Wu et al. presented a printing sequence planning algorithm for this 5-DOF printer [WPGM16]. They separated the consideration of collision constraint and connectivity constraint and use a heuristic constraint removal technique to identify a subgraph in the full constraint graph to balance the existence of feasible extruder orientations and printing sequences. Recently, Dai et al. extended the degrees of freedom by having a stationary extrusion head and a rotatable printing bed held by an industrial robot and presented shape decomposition algorithms to handle volume-to-surfaces and surfaces-to-curves decomposition, while taking into account the support and robot constraints [DWW+18].

Huang et al. worked on an extrusion sequence planning problem for robotic spatial printing [HZH+16]. They mounted a 3D printing extruder on an industrial robot and extrude plastic directly in 3D space. They considered the printing sequence planning problem as a constrained search problem and developed a constrained graph cut algorithm to divide the input wireframe into subgroups. This divide-and-conquer strategy makes the search more tractable, but the robot's kinematics and collision constraints are not considered in the search.

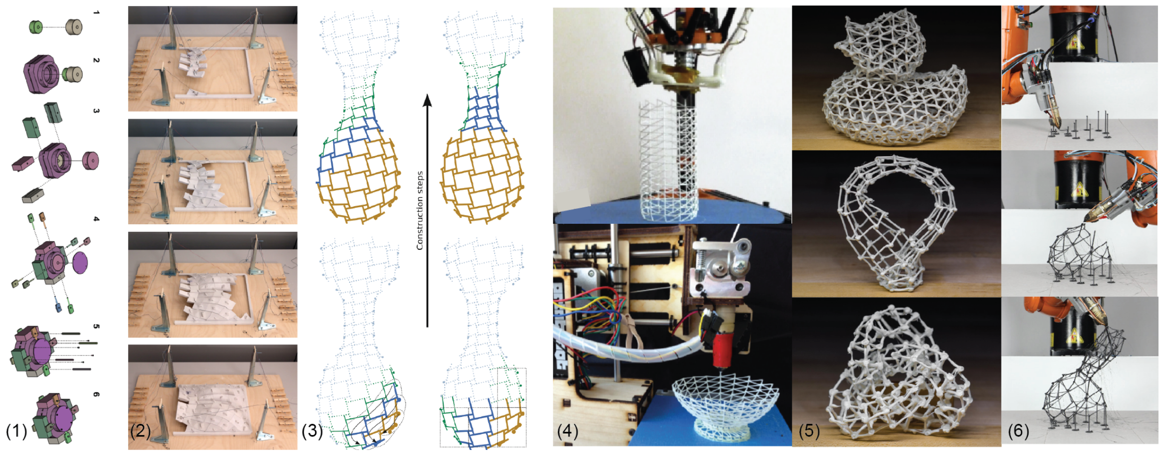


Figure 2-22: Examples of assembly sequencing work in computer graphics: (1) assembly instruction generation [APH+03]; (2) cable-assisted self-supporting masonry assembly [DPW+14]; (3) WirePrint [MIG+14]; (4) meshes printed by a 5-dof wireframe printer [WPGM16]; (5) robotic fabrication of spatial frames [HZH+16]. Images by corresponding papers.

Structural mechanics Finite-element-based elastic simulation of continuum bodies has become a mature computational technique and has been widely applied in the industry from visual effects to stress prediction of mechanical parts [SB12]. These FEA-based techniques can simulate the *stiffness* of a system, usually predicting the displacements and stress of the system. In contrast, for an assembly consisting of rigid parts, *stability* analysis is required since the parts are joined by non-permanent joints and thus do not behave like an elastic body under loads. Two types of stability are defined mathematically and solved computationally in the literature: static equilibrium and global interlocking.

A static equilibrium means that there exists a network of interaction forces among elements that can balance the external forces acting on each part, i.e. the net torque and force on each part is zero [WOD09]. Thus, an assembly's equilibrium can be formulated as a linear system with inequality constraints on the internal forces. This method is called the rigid block equilibrium (RBE) method. The strength of the RBE method is that it allows for a penalty formulation, which provides the users with the localization of unstable regions as meaningful structural hints, and thus enables different design options [WOD09]. One limitation of such a method is that it might falsely predict stability in certain sliding failure cases [YKGA17]. A variational static analysis method is proposed to amend the RBE method to exclude physically unrealizable forces in the sliding failure cases [YKGA17]. In addition, an extension of RBE is recently proposed to address equilibrium with friction when complex, non-planar interface geometry is involved [KIT+22]. [SPV+16] proves that a small modification to the linear elastic FEM makes it equivalent to the RBE method on static analysis problems.

In an interlocking assembly, there is only one moveable part, called the key, while all other parts as well as its subset of parts are not moveable relative to one another by their geometric arrangements [SFCO12]. The test for global interlocking tries to identify if there exists a motion that can take out any part(s) without any collision. An assembly is considered to be interlocking if such a take-out motion does not exist. There are two ways to test for global interlocking: (1) a direction-blocking graph-based method that considers sequential translational motion only [WSP18]; (2) an inequality-based method that includes infinitesimal rigid-body motions [WGZ+19], based on solving the non-penetration linear inequalities in a rigid body system [KSJP08]. However, these two methods can only assert whether an assembly is stable (a yes or no answer), but cannot give a quantitative measure of how stable it is. Borrowing ideas from tilt analysis for measuring a masonry structure's stability under lateral acceleration [Och02, Och02], [WGZ+19] generalizes the classic tilt analysis and proposes an algorithm to compute the measure based on the RBE theory.

Packing efficiency Packing is a classic problem in computational geometry and operational research, to place as many objects as possible in a given container without mutual collisions. In general, the packing problem is proven to be NP-hard [LMM02], but due to its practical relevance, extensive research efforts have been devoted to practical algorithms for solving 2D and 3D problems. 2D packing problems show up in applications like texture atlas layout [LFY+19], artistic primitive layout [RRS13], primitive packing on freeform 3D surfaces for architecture

[SHWP09] or personal fabrication [CML+17]. 3D Assembly packing involves packing elements of a given assembly in a given container, useful for saving spaces for storage and transportation, improving manufacturing efficiency [CZL+15] and minimizing cutoff material waste [KHLM17]. Existing work on assembly packing minimizes a variety of objectives, including the packed state's height [CZL+15], underlying free volume [VGB+14, Att15], while making sure that there is no collision among elements in the packed configuration and there exists a valid assembly plan for the packed state [YCL+15]. Proposed algorithms to solve the assembly packing problem can be divided into two categories: iterative and global. Iterative approaches place the given assembly elements one by one into the container while minimizing certain objectives using certain heuristics [VGB+14, Att15, CZL+15, KHLM17]. Global approaches try to find poses of the assembly elements in the packed state by solving a global optimization problem. Existing methods include optimization with random multi-restart [YCL+15] and a hybrid continuous-discrete optimization scheme [MCHW18]. Readers are referred to [CKPT17, LTO+20] for a comprehensive review in literature of packing.

Figure 2-23: Examples of packing: (1) art primitive packing [RRS13]; (2) tile decors packing [CML+17]; (3) Packing for zero-waste furniture design [KHLM17]; (4-5) Assembly packing for 3D printing [Att15, YCL+15]. Images by corresponding papers.

2.4.2 Computational design of assemblies

Computational assembly design methods are classified into two categories: design for fabrication 2.4.2.1 and design for performance 2.4.2.2.

2.4.2.1 Design for fabrication

Existing design-for-fabrication techniques aim to facilitate the use of certain fabrication techniques, addressing the fabrication methods' limitations and constraints. Existing work proposes shape decomposition and approximation algorithms of a given 3D volumetric shape, converting them into assemblies to address these fabrication constraints.

Discrete shape decomposition for 3D printing and CNC milling 3D printing can fabricate complex objects with fine details but has some well-known limitations:

- The maximum size of the printed object is usually limited by the machine size.
- A support structure is needed for objects with overhangs.
- The printing process is slow, and the printing material is expensive.
- It's hard to obtain high surface finishing quality in 3D printing objects.

Much work has focused on developing techniques that improve the user-specified object for support, surface finishing, etc. [LEM+17]. In contrast, assembly-based techniques try to address these limitations by decomposing a 3D shape into multiple printable parts. While the original 3D shape is the target geometry that users care about the most, the shape decomposition itself also leaves its aesthetic mark on the finished product and thus should be conceived as an essential part of the design. Following the seminal work of [LBRM12] to address the size limitation of 3D printing, multiple papers use clipping plane-based methods to minimize the total amount of support required for printing the parts [YYT+17, WQZ+18, KFW19, HLZC14]. Recent work devise decomposition algorithms for 3D printers that have more degrees of freedom, further reducing the number of supports required [GWYN19, WDF+20].

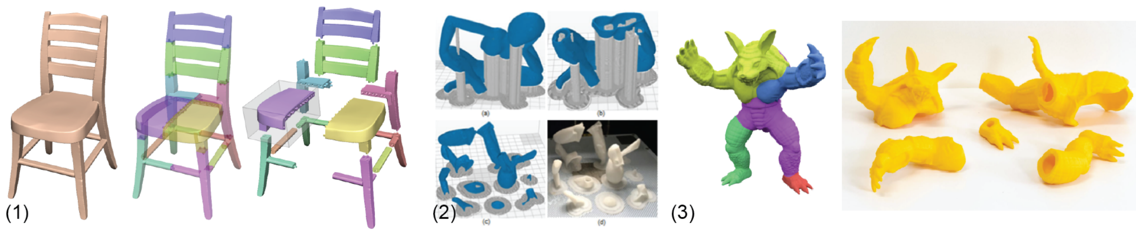


Figure 2-24: Examples of design for 3d printing: (1) shape decomposition to overcome printing size limitations [LBRM12]; (2) shape decomposition for support-free printing [WQZ+18]; (3) shape decomposition for hiding cutting seams [FAG+20]. Images by corresponding papers.

Besides the packing-efficiency-driven shape decomposition strategies that are summarized in Section 2.4.1, research efforts went into improving the final appearance of 3D printed assemblies. Existing research includes placing cutting seams in self-occluded areas of the model [LBRM12, FAG+20] or making the cutting seams align with the surface details [YCL+15]. To minimize the staircase effect coming out of the layer-based printing technology, some approaches decompose an object into multiple parts and find optimal printing direction for each part [HBA13, WZK16, WDF+20]. Beyond decomposition-based methods, other methods use adaptive layer height [WCT+15] or curved, non-horizontal slices [ERP+19] to alleviate the staircase effects. Readers are referred to [LEM+17] for a comprehensive review in this direction.

As a subtractive manufacturing technique, CNC milling requires that the rotatory tool have access to all the surfaces of the shape. Among all the milling machine types, 3-axis CNC milling is the most prevalent, inexpensive, and easiest-to-use one. Since the milling bit is translated along a Cartesian system, the machinable shape is limited to height-field blocks, which are solids with a flat base and a height-field surface defined along the direction orthogonal to the base. To address this limitation, height-field decomposition methods aim to cover the outer surface of the input shape with a set of overlapping-free height-field blocks. Unfortunately, this height-field decomposition problem is proven to be NP-hard [FM01], because of the gigantic discrete-continuous search space that includes the number of blocks, the base configurations, axes, and the height-field surface of each block. Existing attempts include two-step approaches that first define the number of blocks, bases and axes and then solve for height fields [ACP+14, FCM+18]. Other methods formulate the problem as an optimization problem and use a certain hierarchy to find solutions, usually involving first generating patches, then optimizing the patches via discrete optimization procedures, and finally resolving the collision among the blocks [HMA15, MLS+18]. Recently, shape segmentation methods for 4-axis CNC milling are proposed as well [NTM+21].

Figure 2-25: Examples of height-field decomposition methods:(1) Two-step approach based on polycube mapping [FCM+18]; (2) Optimization-based approach [MLS+18]. Images by corresponding papers.

Shape approximation for laser cutting Compared to manufacturing techniques that produce volumetric objects, laser cutting can only produce 2D shapes with polygonal contours. The low cost, fabrication speed and precision of laser cutting make it the most accessible digital fabrication technology for prototyping and production. Design for laser cutting aims to abstract or approximate a given 3D shape with 2D planar parts. In these assemblies, joining 2D laser-cut parts to form a steady assembly is a challenging, and often physically daunting task due to the limited contact area among the planar parts. Connection can be made by using additional connectors [CSLM13] or wires [RA15], by gluing overlapping foldable paper strips [Leu18], or by creating integral joints among parts [WSP18]. For laser-cut parts with a certain thickness, creating integral joints among parts is the most effective way of connection, which not only stabilize the assembly but also simplify the assembly process by providing geometric features to help alignment.

Laser-cut parts can be designed as abstract approximations of a volumetric shape or a surface. Popular in toys and sculptures, laser-cut parts intersect with each other in 3D space to form a spatial assembly, offering a coarse, yet effective way to convey the essential geometric features (Figure 2-26-(1)). Computational tools are proposed for designing shape abstractions [MSM11, MUS14, HBA13] and designing furniture that includes rigidity and assembly constraints [SP13b] and waste minimization objectives [KHLM17]. In the architecture domain, [Sas07] was the first to propose using the CNC-cut panels with friction-only joints to assemble houses without the need of screw or glue (Figure 2-26-4). Recently, laser-cut paper ribbons can be composed to approximate a volumetric shape with locally varying compliance [SIS21].

Ribbon-shaped slices can be composed to form a hollow assembly that approximates a surface. Existing efforts include using (1) long, curved slices with prefabricated halved joints [CPMS14]; (2) short, straight slices joined with additional connectors [RA15]; (3) bent, woven ribbons [VZF+19, RPC+21].

Shell-like assemblies are designed to approximate a closed 3D surface for quick and low-res fabrication. Different from the previous types, each part in the assembly acts like a mesh face, resulting in a geometrically water-tight surface (Figure 2-26-3). Finger joints are widely used for such shell-like assemblies [CSLM13, CS16, BSK+19]. Finally, origami patterns can be embedded in the design and used as laser cutting patterns, resulting in a spatial, economic assembly that does not need a complex, discrete assembly process [MKB, LPR20].

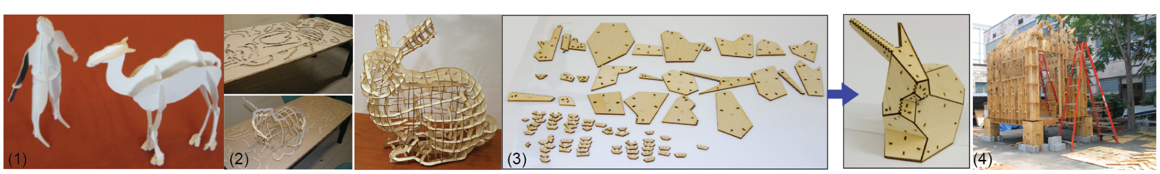


Figure 2-26: Examples of shape approximation method for laser cutting: (1) volumetric shape abstraction with spatial assembly [MSM11]; (2) surface abstraction with sparse assembly elements [CPMS14]; (3) surface abstraction with shell assembly [CSLM13]; (4) the instant house, made out of CNC-cut plywood panel and assembled with friction-only joints without screw or glue [Sas07]. Images by corresponding papers.

Design for mixed fabrication Mixed fabrication aims to combine multiple manufacturing technologies to assemble a single product, combining each technology's strengths or mitigating their drawbacks. Existing research focuses on computational methods that automatically generate geometries of parts and joints for mixed fabrication. One common goal for such design methods is to fabricate large objects at a lower cost and higher speed. One class of work aims to combine coarse laser-cut internal polygonal bases and 3D printed external shell for fine-grained details [GZN+15, SDW+16, CLF+18]. Recently, [WJVS21] considers using recycled material as infills for 3D printing, making fabrication not only faster but also more environmentally friendly. The other class of work involves joining custom-cut, standardized element like rods [TPCS16, CZS+19, Jac19] or recycled objects [KSW+17] with 3D printed joints to form complex spatial structures.

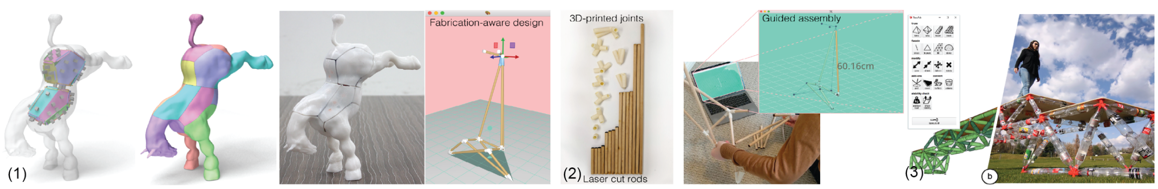


Figure 2-27: Examples of mixed fabrication: (1) cost-effective large object fabrication combining laser cutting and 3D printing [SDW+16]; (2-3) Truss fabrication with 3D printed joints [Jac19, KSW+17]. Images by corresponding papers.

Design for X-fab There is also a growing number of work in computational design for material systems and fabrication techniques that are less standard and cannot be put into the categories covered above. This includes but not limited to fabric concrete formwork [ZFS+19], wire mesh [GSD+14, LLZ+20], wire-bending [LFZ18, MLB16], molding [MPBC16, NAI+18, SJG19], shape approximation by developable [SGC18, IRHS20] and zippable patches [SPS18], thermo-forming [SPG+16], acoustic-effect driven milled glass surfaces [KEN+13, STTP14], quilting [LBDA21, LLG+21], and garment design workflows [NAH+18, Kas22]. Closer to architecture, computational techniques have been developed to support the design and rationalization of free-form architectural surfaces [PEVW15, LZZ+21a] inflatable structures [STK+14, PIC+21], auxetic shells [Luk19], and deployable structures

[PKI+19, RKP+22]. Finally, [Pee16] provides methods for designing and fabricating machines that make.

2.4.2.2 Design for performance

Design for stiffness When an assembly's parts are connected by permanent joints, it behaves like a continuum body when external loads are applied to it. Design-for-stiffness methods help designers to minimize the elastic energy stored by the structure under load, thus also minimizing displacements. [ZPZ13] proposes a method to help designers identify the weakest part of the object and strengthen it. [LSD+16, UMK17] consider shape optimization under load uncertainties. An inverse elastic shape design method to achieve the target shape after deformation is proposed in [CZXZ14]. Reinforcement design strategies are proposed to stiffen a design for materials that are weak in tension [SZB18, GPZ20]. [JTSW17, AJL+19] propose stiffness and volume optimization methods for trusses.

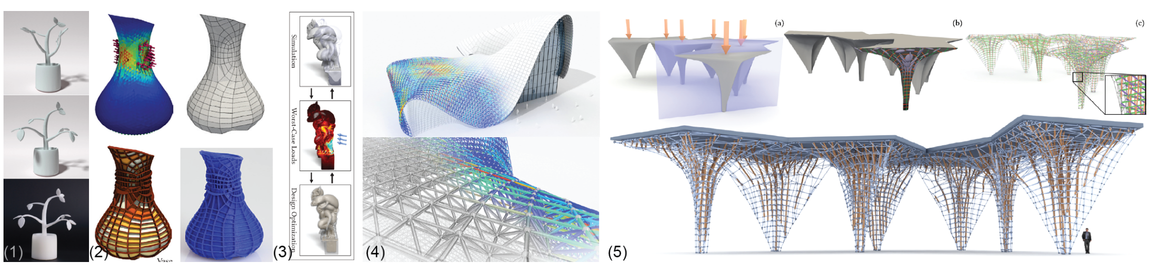


Figure 2-28: Examples of design for stiffness: (1) inverse elastic design of rest shapes [CZXZ14]; (2) shell rib reinforcement design [GPZ20]; (3-4) truss optimization [JTSW17, AJL+19]. Images by corresponding papers.

Design for equilibrium When an assembly's parts are connected only by contact forces, static equilibrium is maintained by pure compression force, counterbalancing external force, mainly gravity. RBE method (see Section 2.4.1) can not only test if a given assembly is in equilibrium under known external force, but also provide a measure of the assembly's infeasibility to be in equilibrium when it fails the test. Thus, optimization algorithms can be applied to navigate a given parametric design model to drive this infeasibility measure to zero [WOD09, WSW+12]. An interactive tool is devised to design assemblies in equilibrium by decomposing a shape into parts with planar cuts, with visualization of unwanted tensile forces to guide the users [FMB15]. Inspired by Thrust Network Analysis [BO07], the graphics community proposes geometry processing techniques to approximate a given free-form surface with self-supporting, compression-only assemblies [VHWP12, PBS13, MHS+19].

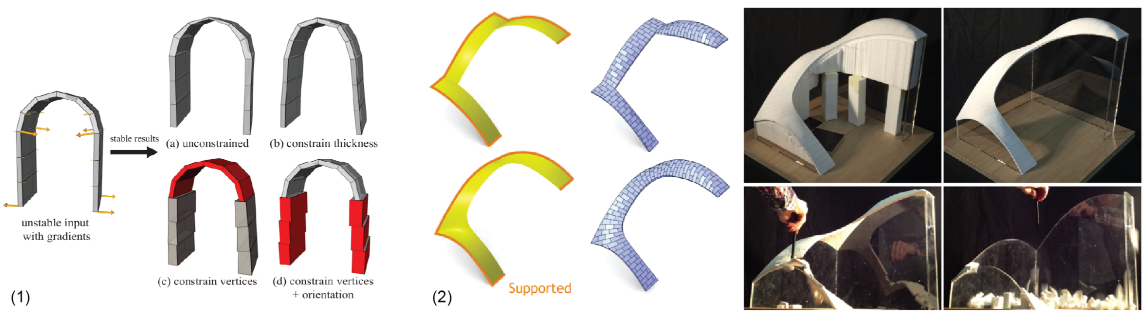


Figure 2-29: Examples of design for equilibrium: (1) gradient-based parametric assembly optimization for equilibrium [WSW+12]; (2) Approximating free-form surface with a self-supporting assembly [PBS13]. Images by corresponding papers.

Design for interlocking Designing interlocking assemblies is challenging because it requires addressing two conflicting properties at the same time: interlocking and disassemblability. Interlocking means the entire assembly is in equilibrium under arbitrary external force if the key part is held in place. Disassemblability demands at least one collision-free plan to separate the parts. The design problem can be classified into shape decomposition methods and joint planning problems, depending on the inputs. When the input is a target shape, the computational design of interlocking assembly can be formulated as a shape decomposition problem. Existing works attempt to limit the search space by restricting to a specific type of interlocking: recursive interlocking [SFCO12]. Given a voxelized 3D shape, this method iteratively extracts pieces while enforcing local interlocking conditions [SFCO12, SFLF15, TSW+19]. When the input is a set of initial parts without joints, designing interlocking assemblies can be formulated as a joint planning problem. The goal is to plan and construct a set of predefined joint types on the given parts to make them interlocking. Existing work extends the recursive interlocking idea to plan joints for plate furniture [FSY+15, SFJ+17]. Recent work unifies the treatment of the shape decomposition and joint planning problems based on a directional blocking graph representation, enabling the algorithm to search over the full space of interlocking configurations and supporting a wide range of overall assembly geometries [WSP18].

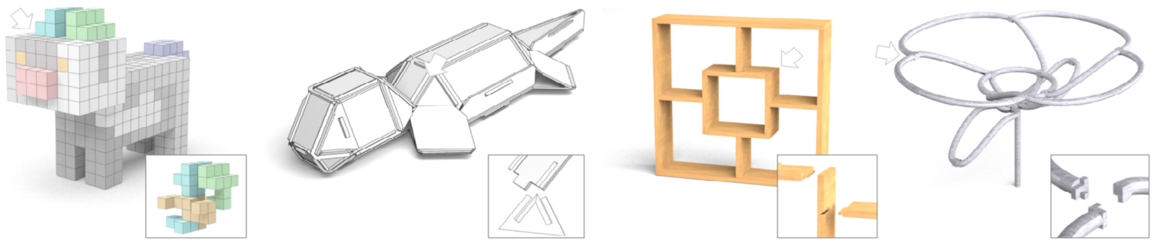


Figure 2-30: Examples of interlocking structures: voxelized puzzle, plate structure, furniture, and frame structure. Images by [WSP18]

Design for X Finally, design-for-X methods aim to expand the geometric possibilities of a certain type of assembly by answering the question: "Given any shape, how can we approximate it with an XXX assembly?" Existing research investigates designing assembles that are built by specific types of tiles, like LEGO blocks [TSP13, KKL14, LYH+15], topological interlocking blocks [WSIP19], or customized interlocking voxel tiles [ZB16]. In addition, other research tries to answer the question: "How can we modify the user-input shape so that it can have certain functionality?" Existing attempts include using micro-structures to control the global elastic behavior of objects [SBR+15, PZM+15], design assemblies for nesting [Jac17], shape optimization for desired metallophone sounds [BLT+15], designing mechanisms for creating animated mechanical characters [CTN+13], and creating spinnable toys from arbitrary input geometries [BBWSH17].

2.4.3 Reflection

Research in computer graphics and human-computer interface features advanced algorithms to address design-related problems for various aesthetic and functional

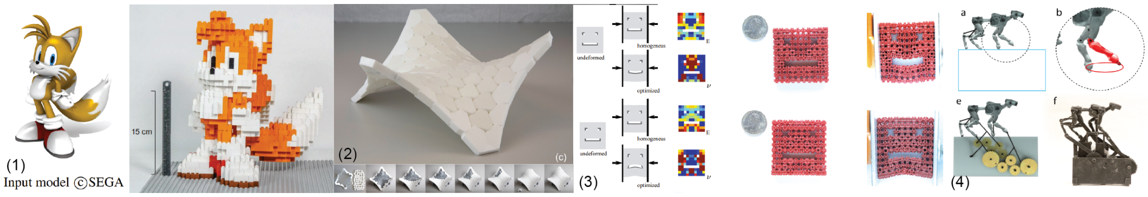


Figure 2-31: Examples of : (1) "Legolization" of a given shape [LYH+15]; (2) topological interlocking assembly that is stable under different orientations [WSIP19]; (3) controlling global elastic behavior through micro-structure design [PZM+15]; (4) computational design of animated mechanical characters [CTN+13]. Images by corresponding papers.

objectives. Its impacts have gone far beyond the domain of animations and the visual effect industry, with increasing relevance from applications in manufacturing and architecture. In graphics literature, design problems are usually formulated as optimization problems with objectives and constraints expressed as quantifiable energies. This thesis aims to devise automated planning techniques that can facilitate a measure of constructability so that it can be used in a design optimization framework. In the next section, we review existing efforts in the field of automated planning and highlight why existing research is not adequate as a satisfactory solution for the planning of robotic construction tasks.

2.5 Automated planning

There has been an interest in automated planning since the earliest days of robotics, which concerns the development of algorithms to decide what sequences of commands the robot should execute to accomplish some goals [FN71][7]. The first class of planning problems, called motion planning, is to move the robot from one configuration to another configuration without colliding with the objects in the world. Motion planning is first formulated by [LPW79] as a search for paths in the robot's configuration space, which is a continuous, bounded space with dimensions representing the controllable joints of the robot. Collision-free robot motions are impor-

[7] Part of this section is adapted from the literature review chapter of the author's master thesis[Hua18].

tant but do not enable robots to interact with the world. For the robot to, for example, move objects by picking them up and placing them, planning needs to encompass a much bigger state, which includes objects that the robot is manipulating, the grasp it is using, and the poses of other collision objects. To keep the search algorithmically feasible, planning for robotic manipulation (technically known as multi-modal motion planning) is best viewed as a hybrid discrete-continuous search problem of selecting a finite sequence of discrete action types (e.g. which objects to pick and place), continuous action parameters (such as object poses to place and grasps), and continuous motion paths. The artificial intelligence (AI) community has addressed problems of planning in very large discrete domains [GNT04]. Task planning (or action planning) refers to the composition of a sequence of symbolic actions to achieve certain high-level goals (e.g. the detailed coordination of factory production and shipment to deliver goods to customers). Research in task and motion planning (TAMP) aims to combine the AI approaches to task planning and the robotics approach to motion planning to derive automated planning systems that can reason both symbolically for discrete, "high-level" robot actions and geometrically for continuous, "low-level" robot motions. Among all the planning problems above, TAMP is the most computationally difficult in theory [DKLP16] and requires algorithmic sophistication to derive usable planning systems in practice. Figure 2-32 provides an overview of the relationship between all the planning problems discussed above, based on the involved search spaces' characteristics.

This section provides a brief overview of key research in automated planning. This thesis draws on ideas from the vast literature in this area, including motion (Section 2.5.1) and manipulation planning (Section 2.5.2), task planning from AI (Section 2.5.3), and task and motion planning (Section 2.5.4). We end this section with a reflection on why existing automated planning methods are not sufficient for robotic assembly in construction, which motivates the new research presented in this thesis. Readers are referred to [GCH+21] for a recent, comprehensive review on the TAMP-related research.

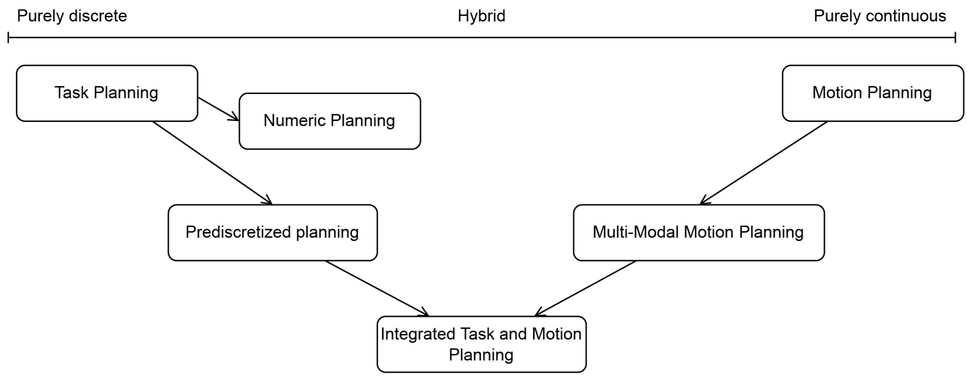


Figure 2-32: Taxonomy of automated planning approaches based on their search spaces' characteristics. Image inspired by slides from Caelan Garrett.

2.5.1 Motion planning

The problem of planning motions for a robot with d degrees of freedom can be formulated as finding a continuous trajectory for a point representing the robot's configuration through the d-dimensional configuration space, subject to various constraints. The simplest form of motion planning (MP) problems involves free-space motion, in which the robot simply needs to move through the space without colliding with itself and the environment.

Although the MP problem has been proven to be PSPACE-hard [Can88], there exist efficient algorithmic approaches that are used widely in practice, with many algorithms well implemented in open-source software like OMPL [SMK12]. The two most popular approaches are sampling-based MP [KSLO96, LKJ01] and trajectory optimization [RZBS09, KCT+11, SDH+14]. Despite their popularity in practice, both classes of algorithms are not complete, meaning that they cannot identify infeasible problems. However, many sampling-based MP algorithms can be shown to be probabilistically complete, meaning that the probability of their failure to find a solution converges to zero as running time increases, if such a solution exists. [LaV06] provides a comprehensive overview of MP algorithms.

In constrained motion planning, the robot is subject to additional constraints other than being collision-free, e.g. keeping its end effector in a straight line or maintaining certain end-effector orientations. These constraints confine the feasible configuration space into a much lower-dimensional subspace. As a result, naive sampling in the full configuration space will have a near-zero probability of producing a feasible sample in the subspace, rendering many sampling-based MP algorithms ineffective. The most general approach for constrained MP develops sampling and connecting operations that project values onto the constrained surface. See [KMK18] and [KMK19a] for a comprehensive survey on these techniques.

2.5.2 Manipulation planning

In manipulation planning, the goal is not only to move the robot without colliding with objects in the environment, but also to contact, operate, and interact with objects in the world to achieve high-level goals. This problem has been addressed from the earliest days of algorithmic motion planning [LP81, LPJM+87, Wil91]. Alami et al. pioneered the modern treatment of this problem, which involves continuous grasps as well as continuous placements [ASL90, ALS94]. They introduced the concept of manipulation graphs that breaks the problem of one robot moving one object in a potentially complex environment into several problems of moving between connected components of the combined configuration space where each component shares the same grasp. A solution is an alternating sequence of transit and transfer paths corresponding to the robot moving while its hand is empty and the robot moving while holding an object. Siméon et al. [SLCS04] expanded this work to a more realistic setting that requires multiple re-grasps, by using probabilistic roadmaps [KSLO96] to approximate each component of the robot's configuration space. However, maintaining an explicit characterization of the free configuration space can be prohibitively expensive in high-dimensional planning problems.

Hauser et al. identified a generalization of manipulation planning as multi-modal motion planning, i.e. planning for systems with multiple modes, representing different sub-manifolds of the configuration space under constraints [HL10, HNTH11]. The key insight in this approach is that, as in the manipulation graph, one can conceptualize the planning process as alternating between moving in a single mode, where the constraints are constant and switching between modes. So the solution requires being able to plan within a single mode and identifying configurations where modes can change, which is in general specific to the task. Hauser et al. provided a probabilistically complete algorithm to solve this type of problem assuming that effective single-mode planners and mode-transition samplers are available. However, this pure uni-directional sampling-based algorithm has trouble solving high-dimensional problems, e.g. problems that involve more than tens of objects. Barry et al. defined a bidirectional rapidly-exploring random tree (RRT) search of the combined configuration space [BKLP13]. Recently, [SNF+17, VBR20] extend these ideas to asymptotically optimal multi-modal motion planning.

Stilman et al. addressed a version of manipulation problems called navigation among moveable obstacles (NAMO), where the robot must reach a specified location among a field of moveable obstacles [SK08, SSKA07]. To solve monotonic NAMO problem instances, where each object can be moved at most once, they plan backward from the goal and use swept volumes to recursively determine which additional object must be moved recursively. Van Den Berg et al. [BSK+09] provided a probabilistically complete algorithm for NAMO. However, the algorithm assumes the ability to fully characterize the connected components of the configuration space of the robot at each planning step, which is computationally prohibitive for robotic configuration space with more than two dimensions.

Rearrangement planning is a special instance of pick-and-place planning where all objects have explicit goal poses. These problems are very similar to robotic assembly planning problems addressed in this thesis, where object goal poses are specified in the input design model. Extending existing work on NAMO [SSKA07] to non-monotone problem instances, Krontiris and Bekris provided an algorithm that constructs a probabilistic roadmap (PRM) in the combined configuration space [KB15, KB16]. To overcome the inefficiency caused by backtracking search, they proposed a faster approximation by performing a topological sort on a constraint graph between objects using minimum removal paths (MCR) [Hau12, Hau13]. The use of the PRM recovers completeness for problems that could not be solved by a greedy backtrack search planner, but the lack of search guidance forces the planner to explore many arrangements. Dogar et al. propose an algorithm for multi-robot grasp planning using a constraint satisfaction problem (CSP) formulation [DSBR15]. They use re-grasp, a domain-specific operation, to remove constraints in the constraint graph, balancing between solvability and the number of re-grasps performed.

2.5.3 Task planning

The AI community has had a long-standing focus on planning in discrete domains with very large state spaces but made tractable by using representations and algorithms that exploit the underlying structures of the domains. [GNT04] provides a comprehensive discussion of task planning from the AI perspective, and [KM20] surveys task planning in robotics.

The simplest formulation of AI planning is to specify the problem by four components: (1) a set of states, (2) a set of transition rules that describes legal changes to the state, (3) an initial state, and (4) a set of goal states. The objective for a planner is to find a plan that consists of a sequence of transitions that advances the initial state into a goal state. The problem can be reduced to a graph traversal problem, where the vertices are states, and directed edges are transitions, and solved by using standard graph-search algorithms. However, the state spaces considered are very large, and the branching factor of each transition is massive. Thus, one focus of AI planning is to devise functional and compact representations of the planning problems. To better benchmark the algorithms, the AI planning community worked on defining languages for specifying the problems, for example, SAS+ [BN95] and the most widely used Planning Domain Definition Language (PDDL) [McD91]. PDDL can be seen as a transition system where state variables are Boolean facts. Operating upon these shared, universal problem-definition languages, the AI planning community has developed domain-independent algorithms that can operate on any problem expressed in these formats, without any additional information about the problem. When expressed in these formats, the problem descriptions become factored and thus make it easier for generic search algorithms to take advantage of their combinatorial structures. For example, many efficient algorithms solve a relaxed, simplified version of the original problem first and use the solutions as heuristics to estimate the distance to the goal state [Hel06].

The basic formulation of task planning is purely discrete. Numeric planning aims to extend task planning to support real-valued variables such as fuel and battery charges. Several numeric planners extend discrete task planning algorithms by adding heuristics that obtain more accurate estimates by separating the treatment of numeric and Boolean variables [Hof03, CFL13]. While most of these planners are limited to problems with linear or polynomial dynamics [Hof03, BGMG15, CFLM16], some planners can handle non-polynomial dynamics by discretizing time [PFL+16] or support convex dynamics by requiring a convex decomposition of the robot's configuration space [FGWK18]. Although these planners have many use cases, they cannot be directly applied to robotic problems because they assume the set of actions is finite.

Another method for extending the capacities of PDDL planners is semantic attachments [DEK+09, Dor15]. Semantic attachments are functions computed by an external module, which include condition-checker modules that test Boolean action preconditions and effect modules that modify numeric state variables. In robotics domains, semantic attachments can be used to, for example, model reachable kinematic conditions that are defined on an external geometric representation of the state that is not directly accessible to PDDL planners. Planning Modulo Theories (PMT) [GLFB12] generalizes semantic attachments by supporting the composition of several modules through custom first-order theories. However, like many numeric planning techniques above, these techniques are still limited to domains with a finite set of actions. In the robotics context, this means that they can only handle domains that are pre-discretized, where a finite set of object poses, object grasps, and robot configurations are pre-defined manually to be considered by the planner.

2.5.4 Task and motion planning

Recent work in task and motion planning (TAMP) combines discrete task planning (Section 2.5.3) and continuous motion planning (Section 2.5.1) to simultaneously plan for discrete objectives as well as robot motions. TAMP aims to enable robots to operate in applications such as cooking, which require discrete choices of which objects to grasp or cook as well as continuous choices of which joint angles and object poses can physically perform each task. Readers are referred to [GCH+21] for a comprehensive survey in TAMP.

A key challenge in TAMP is that often physical constraints such as collision, kinematic, and visibility constraints can restrict which high-level actions are feasible. The pioneering work aSyMov system conducts an interleaved search at the symbolic and geometric levels [CAG09]. Their approach can be viewed as using the task planner as a heuristic to guide the motion planning approach. Plaku and Hager took a similar approach but used a task-planner-derived heuristic to bias the sampling instead of guiding the search [PH10]. However, since the task-level planner does not know about geometry, its value as a heuristic is limited.

Lozano-Pérez and Kaelbling introduced the Hierarchical Planning in the Now (HPN) approach, a regression-based symbolic planner that uses generators that perform fast approximate motion planning to select geometric parameters [KLP11]. Garrett et al. gave an algorithm for planning in hybrid spaces by using an approximation of the planning problem to guide the backward generation of successor actions to be considered in the forward search [GLPK15]. Both of these two approaches require an inverse model to specify the generators to be compatible with their backward searches.

The FFRob algorithm of Garrett et al. samples a set of object poses and robot configurations and then plans with them using a search algorithm that incorporates geometric constraints in its heuristic [GLPK18a]. An iterative version of the algorithm has been proposed to have a probabilistically complete guarantee and exponential convergence. Their recent work generalizes this strategy of iteratively sampling and searching from pick-and-place domains to domains with arbitrary conditional samplers [GLPK18b].

[PSSA12] and [dSPGA13] use Hierarchical Task Networks (HTNs) [EHN94] to guide the search over plan skeletons, which are discrete action sequences with unbound continuous variables, using knowledge about the task decomposition. The search over plan skeleton backtracks when it is unable to bind the free variables of a skeleton. Lagriffoul et al. proposed a constraint-satisfaction approach to interleave the symbolic and geometric searches and focus on limiting the amount of geometric backtracking [LDB+14]. They generate a set of approximate linear constraints on robot configurations and object poses that allow them to efficiently determine which assignments are feasible and rule out many useless branches and thus significantly limiting backtracking. Viewed from a constraint satisfaction perspective, their approach can be thought of doing backtracking with a forward checking of the kinematic constraints and a fixed value ordering. Lozano-Pérez and Kaelbling took a similar approach but leveraged constraint satisfaction problem (CSP) operating on discretized variable domains to bind free variables [LPK14].

Erdem et al. planned at the task-level using a Boolean satisfiability (SAT) solver, initially ignoring geometric constraints and then attempting to produce motion plans satisfying task-level actions [EHP+11]. If an induced motion planning problem is infeasible, the task-level description is updated to indicate motion infeasibility using a domain-specific diagnostic interface. Dantam et al. extended this approach by formulating task and motion planning problem more generally as a Satisfiability Modulo Theories (SMT) problem [TDKKCEK16]. They used an incremental constraint solver to add motion constraints to the task-level logical formula when a candidate task plan is found. Their approach adjusts to motion planning failure automatically and allows previously failed motion planning queries to be reconsidered. The algorithms proposed in [EHP+11, TDKKCEK16] both assume a priori discretization of all continuous values apart from configurations. Srivastava et al. [SFR+14] remove this restriction by using symbolic references to continuous parameters. Lagriffoul and Andres proposed to use Answer Set Programming

(ASP) [Lif08] to enable a richer failure explanation mechanism at the interface between symbolic and geometric search spaces [LA16]. They used a domain-specific interface to bind values for symbolic references and update the task-level description when none is available.

Toussaint et al. used a nonlinear constrained optimization to solve for values that satisfy constraints and minimize the costs along a plan skeleton [Tou15]. They incorporated this optimization into a higher-level forward search over plan skeletons called Logic-Geometric Programming (LGP), which also prunes branches in its search using a hierarchy of bounds on the nonlinear programs. In addition to geometric constraints, their approach has also demonstrated the ability to plan in the presence of dynamical [TAST18] and forceful constraints [THD20]. Recently, Hartmann et al. applied LGP-based techniques to plan the construction sequence and motions of multiple robot agents [HOD+20, HOD+21], marking the first few attempts at applying TAMP techniques in solving construction-related planning problems.

Recently, Garrett et al. proposed PDDLStream that extends the PDDL planning language (see Section 2.5.3) to support a generic, declarative specification for specialized sampling procedures that are treated as black-boxes, which allows the system to express and solve a wide variety of planning problems [GLPK20]. Such a modularized system can be extended to planning with learned models of skills [WGKLP21], planning in partially-observable domains by reasoning in belief space [GPL+20], and planning with imperfect understandings of the world by incorporating state-of-the-art perceptual tools in the planning model [CFK+22].

2.5.5 Reflections

Robotic assembly is a subclass of the generic planning problems considered by TAMP. Although existing TAMP research has demonstrated effective planning algorithms that can handle hybrid reasoning at both symbolic and geometric levels, their applications in robotic construction are still quite limited. The reasons are two-fold. First, many TAMP systems require the users to formulate their planning problem in the format of symbolic states and actions (e.g. [GLPK20]) or objectives and constraints for nonlinear optimization (e.g. [Tou15]), both of which are formalisms rarely used in the architectural community. Second, a key challenge in robotic assembly for construction is that they often require manipulating many more objects, which leads to substantially longer planning horizons. Solutions to most TAMP benchmarks involve fewer than 30 high-level actions while many construction problems may require assembling several hundreds of elements [Lag16]. This thesis attempt to address these challenges by providing the formalization and solving techniques of robotic assembly planning with repetitive (Chapter 3 and Chapter 4) and non-repetitive plan skeletons (Chapter 5).

2.6 Conclusions

This literature review has highlighted the breadth of methods related to the design and automated construction of structures. We observe that our current computational design capabilities have exceeded what the current construction paradigm can offer. While construction robotics promises a cost- and time-effective paradigm for building diverse geometries demanded by efficient structures, the technical challenge of planning is limiting the potential and impacts of this technology. In manufacturing and computer graphics, functional objectives and fabrication-related constraints are formulated in mathematical forms, so they can be included in algorithm-driven design methodologies. Finally, automated planning literature offers advanced algorithms to plan both robot's high-level actions and low-level motions to achieve certain goals, but these ideas have not yet been applied to generalized robotic assembly problems.

To broaden the impacts of robotic assembly in construction, we need better planning tools. Existing planning methods either generate robot programs that do not have any guarantee that they will work, or it takes too long and too much human labor to generate. An ideal planning tool should have the following characteristics:

- Being able to generate robot programs that are guaranteed to work.
- Off-load tedious, unnecessary human programming work.
- Be versatile for diverse types of assembly tasks.
- Involve human expertise when relevant.

To build such a planning tool, existing work in each area covered in this chapter is encouraging, but still has important shortcomings that limit their adoption and effectiveness. These shortcomings echo the challenges outlined in the introduction and the contributions of this thesis seek to address them. This dissertation has several specific goals:

- Provide a unified formulation of robotic assembly planning with a repetitive plan skeleton and devise scalable planning algorithms to solve the assembly sequence and robot motions without any human intervention. Work in this area is presented first for robotic extrusion in Chapter 3 and then generalized to pick-and-place assembly in Chapter 4.
- Provide a computational workflow for architectural and construction process designers to communicate intent and knowledge to robotic task and motion planners. Work in this direction targets general-purpose robotic assembly planning with non-repetitive plan skeletons and is presented in Chapter 5.
- Demonstrate the use of planning algorithms to derive a constructability measure and use this measure to drive structural design exploration. Chapter 6 presents work in this area.

Chapter 3

Scalable sequence and motion planning for robotic spatial extrusion

We start our investigation in algorithmic planning of robotic assembly in the domain of robotic spatial extrusion. This chapter lays down the mathematical foundation of planning for robotic extrusion of large and complex models, resulting in scalable planning algorithms without any human intervention[1].

3.1 Introduction

Spatial frame structures are used extensively in architecture to represent objects that cannot be easily captured by surfaces or volumetric solids (*e.g.* the Klein bottle in Figure 3-1). These structures are useful due to their high strength-to-weight ratios [TMG+18, HCM18]. Extrusion-based methods, such as 3D printing, can effectively fabricate these geometrically and topologically complex structures. Most existing printing systems deploy a 2.5D strategy where melted materials are accumulated layer upon layer along a fixed direction. These systems are unable to print general 3D frame structures due to their inability to print in arbitrary directions. Robot manipulators have proven to be viable alternatives for fabricating these structures due to their additional capabilities afforded by extra degrees-of-freedom

[1] A version of this chapter has been published in [GHLPM20].

(DOFs) [HL14, YHL+16, TMG+18, PM20b]. However, robotic spatial extrusion has only been applied in limited capacities due to the planning challenges imposed when fabricating large, irregular structures. The robot must respect both collision and kinematic *geometric* constraints present in manipulation tasks, and each partial-structure must respect *structural* constraints that ensure feasible construction. In extrusion planning, a *stiffness* constraint, which prevents significant structural deformation, is the primary structural constraint. Existing algorithms both require strong *human* guidance to solve these problems [HGM18] and lack completeness guarantees [WPGM16, HZH+16].

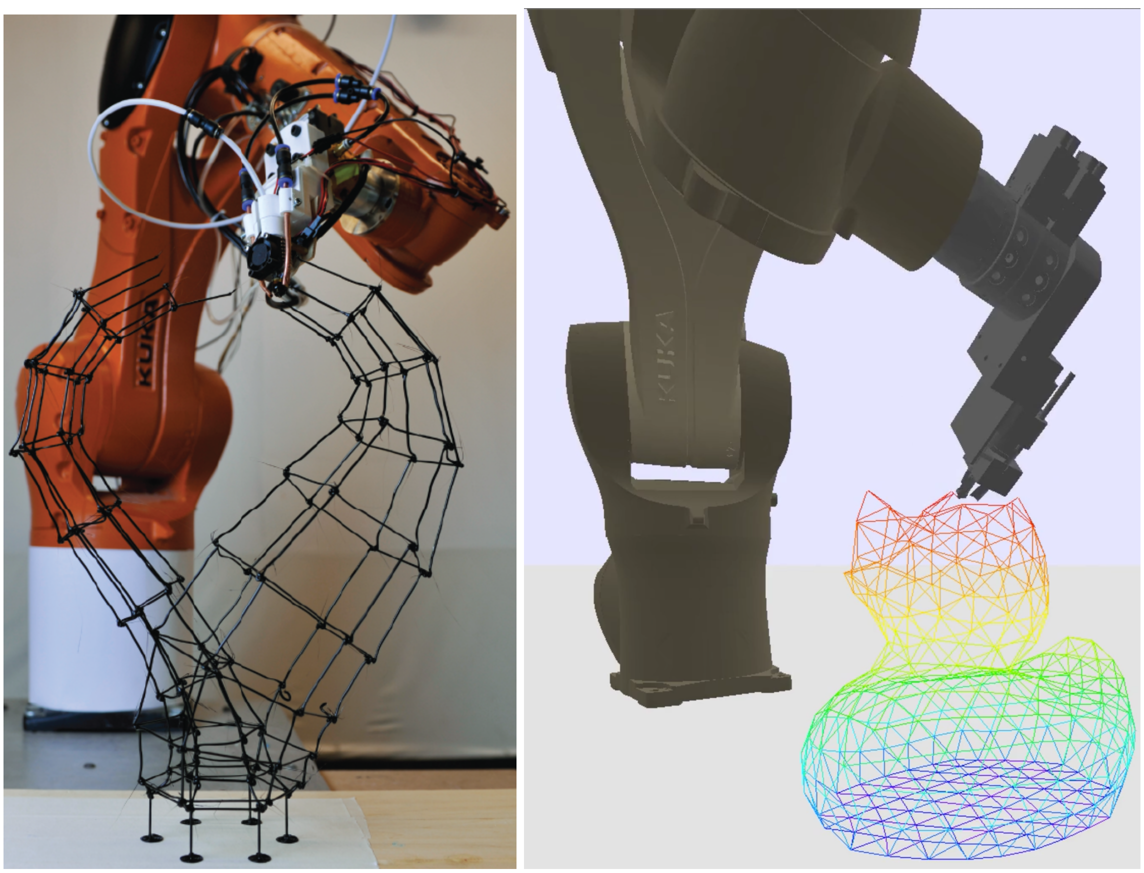


Figure 3-1: *Left*: Real-world Klein bottle extrusion (246 elements). *Right*: Simulated duck extrusion (909 elements).

We present an algorithmic treatment of extrusion planning that focuses on its mathematical form, probabilistically complete algorithms, and algorithms that scale empirically. We identify a dichotomy between satisfying geometric and structural constraints; stiffness most significantly impacts decisions at the *beginning* of construction while collisions most significantly limit actions towards the *end* of construction. In isolation, forward search is most effective for stiffness constraints but backward search is most effective for geometric constraints. We provide algorithms that efficiently plan in the presence of both constraints by globally performing a greedy backward search, using forward reasoning to bias the search towards stiff structures. The contributions of this paper are:

1. A formalization of robotic spatial extrusion in the presence of stiffness and geometric constraints;
2. Efficient and probabilistically complete forward and backward state-space search algorithms;
3. Prioritization heuristics that guide both stiffness and geometric decision-making;
4. An investigation of the failure cases of these methods;
5. Validation of our methods both on long-horizon simulated and real-world extrusion problems.

3.2 Related work

Most existing work on extrusion planning only addresses planning for a free-flying end effector. Wu *et al.* gave an algorithm for planning without stiffness constraints that considers a fixed discretization of end-effector orientations. It performs backward peeling [WPGM16] and computes a partial-ordering of elements that respects collision constraints. Then, it orders elements in a manner that preserves connectivity and the partial ordering. However, this procedure is incomplete because it rigidly commits to a single partial ordering. Huang *et al.* proposed a constrained graph decomposition algorithm to guide the extrusion sequence search [HZH+16]; however, their algorithm is also incomplete. Gelber *et al.* presented a complete forward search algorithm for a 3-axis printer that minimizes the deformation of a structure [GHB18]. Choreo is the first extrusion planning system using a robot manipulator [HGM18]. Choreo decomposes extrusion planning into a *sequence planning* phase, where it plans each extrusion, and a *transit planning* phase, where it plans motions between each extrusion. Because of this strict hierarchy, Choreo is incomplete as it is unable to backtrack in the event that transit planning fails to find a motion plan. Choreo performs a forward search during sequence planning, using constraint propagation to prune unsafe end-effector orientations. To make sequence planning tractable, Choreo requires a user-generated partial ordering on elements.

Task and Motion Planning (TAMP) involves planning both the high-level objectives as well as the low-level robot motions required to complete a multi-step manipulation task [SFR+14, TL17, GLPK18b]. For extrusion planning, the high-level decisions are the extrusion sequence, and the low-level motions are the extrusion and transit trajectories of the robot. A key challenge of extrusion planning when compared to typical TAMP problems is that its planning horizon is often substantially longer. Solutions to most TAMP benchmarks involves fewer than 50 high-level actions [LDG+18], while extrusion problems may require over 900 extrusions (Figure 3-1). At the same time, extrusion planning is less general than TAMP in several ways: 1) there is a single goal state 2) the robot's configuration is the only continuous state variable 3) every solution is an alternating sequence of movements and extrusions of a known length. Similar to how specializing to pick-and-place subclasses of TAMP enables the design of efficient algorithms [KB15, HSK+18], we take advantage of these restrictions and structural properties to develop efficient algorithms that scale to large problems.

Extrusion planning can framed as *Multi-Modal Motion Planning* (MMMP) [**?**, HNTH11], motion planning subject to a sequence of *mode* constraints σ on the feasible configuration space of the robot $\mathcal{M}(\sigma) \subseteq \mathcal{Q}$. Often times, $\mathcal{M}(\sigma)$ might be a lower-dimensional submanifold of an ambient space $\mathcal{Q}$. A critical component of MMMP is identifying *transition configurations* $q \in \mathcal{T}(\sigma, \sigma') \subseteq (\mathcal{M}(\sigma) \cap \mathcal{M}(\sigma'))$ between modes σ, σ', which allow for a discrete *mode switch* from $\sigma \to \sigma'$. Hauser and

Ng-Thow-Hing provide an algorithm for MMMP that performs a forward state-space search through the space of modes [HNTH11]. They prove that their algorithm is *probabilistically complete* [KKL98, LaV06], namely that it will solve any *robustly feasible* [KF11] MMMP problem with probability one. However, their algorithm blindly explores the state-space, which is intractable for the problems we consider.

3.3 Extrusion sequencing

We begin by formulating spatial extrusion planning in the *absence* of a robot. A *frame* structure is an *undirected geometric graph* $\langle N, E \rangle$ embedded within $\mathbb{R}^3$. Let the graph's vertices N be called *nodes* and the graph's edges be called *elements* $E \subseteq N^2$ where $m = |E|$. Each node $n \in N$ is the connection point for one or more elements at position $p_n \in \mathbb{R}^3$. Each element $e = \{n, n'\} \in E$ occupies a volume within $\mathbb{R}^3$ corresponding to a cylinder of revolution about the straight line segment $p_n \to p_{n'}$. A subset of the nodes $G \subseteq N$ are rigidly fixed to *ground* and thus experience a reaction force. Each element $e = \{n, n'\}$ can either be extruded from $n \to n'$ or $n' \to n$. Let *directed* element $\vec{e} = \langle n, n' \rangle$ denote extruding element $e = \{n, n'\}$ from $n \to n'$. We will use the set $P \subseteq E$ to refer to a set of printed elements, representing a partially-extruded structure. Let $N_P = G \cup \{n, n' \mid \{n, n'\} \in P\} \subseteq N$ be the set of nodes spanned by ground nodes G and elements P. Extrusion planning requires first finding an *extrusion sequence*, an ordering of directed elements $\vec{\psi} = [\vec{e}_1, ..., \vec{e}_m]$. We will use ψ to denote the undirected version of $\vec{\psi}$. Let $\vec{\psi}_{1:i} = [\vec{e}_1, ..., \vec{e}_i]$ give the first i elements of $\vec{\psi}$ where $i \leq m$.

3.3.1 Stiffness constraint

The key structural invariant that must hold throughout the extrusion process is a *stiffness constraint* requiring the maximal nodal deformation to be below a given tolerance. Each element experiences a self-weight load due to gravity, which causes the structure to bend. If the displacement is too large, elements might not successfully connect at the intended nodes. We approximate uniformly-distributed self-weight loads by applying half the load at each end of the element and using the fixed-end beam equation for moment approximation [GT97]. The deformation of all the nodes is calculated using finite element analysis of linear frame structures [MGZ99]. For a 3D frame structure, each node has six degrees of freedom (DOF) $(u_x, u_y, u_z, \theta_x, \theta_y, \theta_z)$, which correspond to the translational and rotational nodal displacements in the global coordinate system. Using linear basis functions and the local-to-global frame transformation, we can derive the beam equation to link the nodal load to nodal displacement in the *global* coordinate system [MGZ99] $K_e \left(\mathbf{u}_n, \mathbf{u}_{n'}\right)^T = \mathbf{f}_e$. Then, by concatenating all nodal DOF into a vector $\mathbf{u} = (..., u_{x,n}, u_{y,n}, u_{z,n}, \theta_{x,n}, \theta_{y,n}, \theta_{z,n}, ...)$ for $n \in N$, the system stiffness matrix K is assembled using:

$$K_{ij} = \begin{cases} \sum_{e \sim (i,j)} K_e(\text{e-dof}(i), \text{e-dof}(j)) & \text{if } i \sim j \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

where $i \sim j$ indicates that the nodal DOFs $i, j \in \{1, ..., 6|N|\}$ are connected by an element, $e \sim (i, j)$ indicates that element e connects DOFs i, j, and e-dof(i) gives the corresponding index of the DOF i in the local element system. The support condition specifies a set of fixed nodal DOF indices $\{s_1, \cdots, s_{6|G|}\} \subset \{1, \cdots, 6|N|\}$. The assembled system stiffness equation $K\mathbf{u} = \mathbf{F}$ is rearranged in the form:

$$\begin{pmatrix} K_{ff} & K_{fs} \\ K_{sf} & K_{ss} \end{pmatrix} \begin{pmatrix} \mathbf{u}_f \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{F_f} \\ \mathbf{F_s} \end{pmatrix} \tag{3.2}$$

The submatrix K_{ff} is positive definite (PD) if all elements are transitively connected to a ground node. Then, the nodal displacement under the structure's load can be obtained by solving the following sparse PD linear system: $K_{ff}\mathbf{u}_f = \mathbf{F}_f$. Let the procedure Stiff(G, P) test whether a partially-extruded structure P with ground nodes G satisfies the given maximum displacement tolerance.

Definition 1. An extrusion sequence $\vec{\psi} = [\vec{e}_1, \vec{e}_2, ..., \vec{e}_m]$ is *valid* if $\{e \in \psi\} = E$ and

$\forall i \in \{1, ..., m\}$. $\textsc{Stiff}(G, \vec{\psi}_{1:i})$ and $n_i \in N_{\vec{\psi}_{1:i-1}}$ where $\vec{\psi}_i = \vec{e}_i = \langle n_i, n_i' \rangle$.

3.4 Robotic extrusion

We consider extrusion planning performed by a single articulated robot manipulator with d DOFs. Let $\mathcal{Q} \subset \mathbb{R}^d$ be the bounded configuration space of the robot where $q \in \mathcal{Q}$ is a robot configuration. The robot executes continuous trajectories $\tau : [0, 1] \to \mathcal{Q}$ where $\tau(\lambda) \in \mathcal{Q}$ is the robot's configuration at time λ for $\lambda \in [0, 1]$. The robot must adhere to its joint limits as well as avoid collisions with itself, the environment, and the currently printed elements. Let $Q : P \to \mathcal{Q}$ be a function that maps a set of printed elements $P \subseteq E$ to the collision-free configuration space of the robot $Q(P) \subseteq \mathcal{Q}$. When no elements have been printed, $Q(\emptyset)$ is the collision-free configuration space of the robot when only considering environment collisions, self-collisions, and joint limits. Each additionally printed element weakly decreases the collision-free configuration space, *i.e.*

$$P \subseteq P' \implies Q(P') \subseteq Q(P). \tag{3.3}$$

To ensure τ can be safely executed given printed elements P, $\forall \lambda \in [0, 1]$. $\tau(\lambda) \in Q(P)$. Finally, let $f_p(q) = x_p \in \mathbb{R}^3$ and $f_o(q) = x_o \in \text{SO}(3)$ be the forward kinematic equations for the position and orientation of the end effector when the robot is at configuration q.

3.4.1 Extrusion

The robot extrudes material at the position of its end effector while executing an *extrusion trajectory* τ_e, which prints the continuous curve $l(\lambda) = f_p(\tau(\lambda))$. Thus, element $\vec{e} = \langle n, n' \rangle$ can be extruded by following a trajectory $\tau_{\vec{e}}$ if $\forall \lambda \in [0, 1]$:

$$||\lambda p_n + (1 - \lambda) p_{n'} - f_p(\tau_e(\lambda))|| = 0. \tag{3.4}$$

To prevent the end effector from colliding with the element while it is being extruded, the orientation of the end effector x_o is constrained be within the hemisphere $X_o(\vec{e})$, the set of orientations opposite to the direction of $p_n \to p_{n'}$:

$$X_o(\langle n, n' \rangle) = \{x_o \in \mathrm{SO}(3) \mid (p_{n'} - p_n)^\intercal (x_o \cdot [0, 0, 1]^\intercal) \leq 0\}.$$

Additionally, we enforce that the end-effector orientation x_o remains constant while extruding the element, $\forall \lambda \in [0, 1]$, $||x_o - f_o(\tau(\lambda))|| = 0$ to prevent the extruded material from inducing a twisting force. In practice, we also require the robot to perform *retraction* motions that move into and out of contact with the extruded element without extruding any material. Let $\rho \geq 0$ be an end-effector retraction distance hyperparameter. Then, the retraction position for node n at end-effector orientation x_o is: $r(n, x_o) = p_n + (x_o \cdot [0, 0, -\rho]^\intercal)$. Thus, the end effector moves from $r(n, x_o) \to p_n$ before extruding $\vec{e}$ and from $p_{n'} \to r(n', x_o)$ after extruding $\vec{e}$. We will treat retraction as a component of an extrusion motion. See Figure 3-2 for a visualization of each motion type.

3.4.2 MMMP formulation

Viewing extrusion planning under this lens of MMMP is valuable for understanding the geometry of the problem and its impact on completeness. Extrusion planning has two *mode families*, parameterized mode forms. A single *transit mode* (denoted as α) governs the robot's movement while *not* extruding [**?**, SLCS04]. The only active constraint is trivially that $q \in \mathcal{Q}$. Any probabilistically complete motion planner PlanMotion, such as a Rapidly-Exploring Random Tree (RRT) [LaV98, KSL+19], can be used to plan within transit modes.

An *extrusion mode* $\sigma_{\vec{e}} = x_o \in X_o(\vec{e})$ governs the robot's motion while extruding element $\vec{e} = \langle n, n' \rangle$ by starting at point p_n and ending at $p_{n'}$. Here, x_o is a continuous *coparameter* that defines the end-effector orientation constraint. Because of the position and orientation constraints on the end-effector, $\mathcal{M}(\sigma_{\vec{e}}) \subset \mathcal{Q}$ is a $(d-5)$-dimensional submanifold of the ambient space $\mathcal{Q}$. As typical in constrained motion

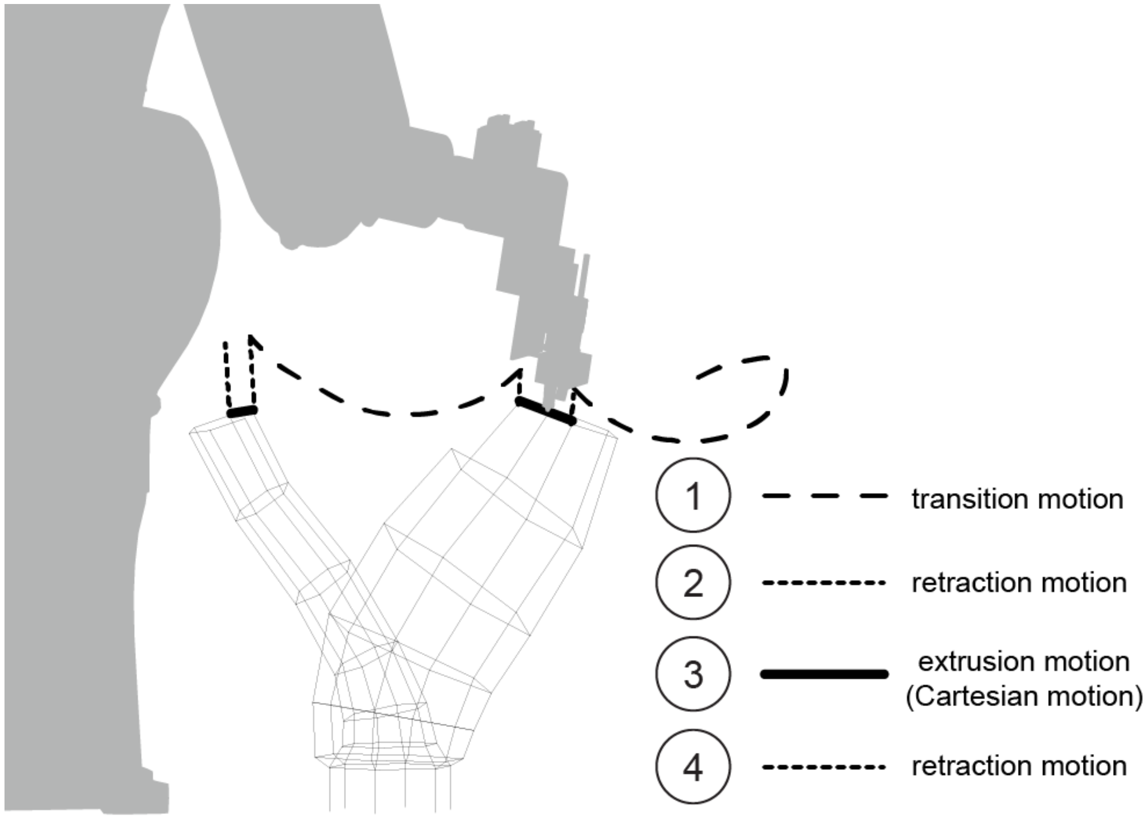


Figure 3-2: Transition, retraction, and extrusion motions for two elements.

planning, we enforce that any trajectory τ operating subject to mode σ stays within an ϵ-neighborhood of $\mathcal{M}(\sigma)$ [Sti10]. Let $\delta(q, \mathcal{M}(\sigma)) = \inf_{q' \in \mathcal{M}(\sigma)} ||q - q'||$ be minimum distance from configuration q to $\mathcal{M}(\sigma)$ and $\gamma(\tau, \mathcal{M}(\sigma)) = \sup_{\lambda \in [0,1]} \delta(\tau(\lambda), \mathcal{M}(\sigma))$ be the maximum distance from trajectory τ to $\mathcal{M}(\sigma)$. We enforce that the maximum constraint violation $\gamma(\tau, \mathcal{M}(\sigma))$ is below a given $\epsilon > 0$. Any probabilistically complete single-mode *constrained motion planner* [Sti10, BSK11, KMK19b] Plan-Constrained can be used to plan within extrusion modes. Finally, let $\mathcal{T}(\alpha, \sigma_{\vec{e}}) = \{q \in \mathcal{Q} \mid f_p(q) = p_n, f_o(q) = x_o\}$ denote the set of *unidirectional* transition configurations from the transit mode to extrusion mode $\sigma_{\vec{e}}$, and $\mathcal{T}(\sigma_{\vec{e}}, \alpha) = \{q \in \mathcal{Q} \mid f_p(q) = p_{n'}, f_o(q) = x_o\}$ denote directed transition configurations from extrusion mode $\sigma_{\vec{e}}$ to the transit mode.

3.4.3 Extrusion problems

Definition 2. An *extrusion problem* $\Pi = \langle N, G, E, \mathcal{Q}, q_0 \rangle$ is defined by a set of nodes N, ground nodes G, elements E, configuration space $\mathcal{Q}$, and configuration $q_0 \in \mathcal{Q}$ specifying both the initial and final robot configuration.

Definition 3. For a given error threshold $\epsilon > 0$, a *solution* to an extrusion problem Π is a valid extrusion sequence $\vec{\psi} = [\vec{e}_1, \vec{e}_2, ..., \vec{e}_m]$ (Definition 1), a sequence of extrusion mode coparameters $\vec{\sigma} = [\sigma_{\vec{e}_1}, ..., \sigma_{\vec{e}_m}]$, and an alternating sequence of $m+1$ transit and m extrusion trajectories $\pi = [\tau_{t_1}, \tau_{\vec{e}_1}, ..., \tau_{t_{m+1}}]$ such that:

- $\tau_{t_1}(0) = \tau_{t_{m+1}}(1) = q_0$
- $\forall i \in \{1, ..., m\}$.
  - $\tau_{t_i}(1) = \tau_{\vec{e}_i}(0)$
  - $\forall \lambda \in [0, 1].\ \tau_{t_i}(\lambda), \tau_{\vec{e}_i}(\lambda) \in Q(\psi_{1:i-1})$
  - $\gamma(\tau_{\vec{e}_i}, \mathcal{M}(\sigma_{\vec{e}_i})) < \epsilon$
- $\tau_{\vec{e}_m}(1) = \tau_{t_{m+1}}(0)$
- $\forall \lambda \in [0, 1].\ \tau_{t_{m+1}}(\lambda) \in Q(E)$.

3.5 Algorithmic tools

We present state-space search algorithms for solving extrusion planning problems. States $s = \langle P, q \rangle \in \mathcal{P}(E) \times \mathcal{Q}$ consist of the set of currently printed elements and the current robot configuration where $\mathcal{P}(E)$ denotes the power set of E. The initial state is $s_0 = \langle \emptyset, q_0 \rangle$ and the goal state is $s_* = \langle E, q_0 \rangle$. The PROGRESSION algorithm (Section 3.7) performs a forward search from $s_0 \to s_*$, and the REGRESSION algorithm (Section 3.7) performs a backward search from the goal state $s_* \to s_0$. Both PROGRESSION and REGRESSION perform a greedy best-first search [RN16] guided by a priority function $k(\eta)$ defined over *search nodes* η. On each iteration, the search node η in the *open list* O that minimizes $k(\eta)$ is expanded.

The key trade off when designing these algorithms is the impact on satisfying stiffness and geometric constraints when searching forwards versus backwards. For each constraint in isolation, it is advantageous to search from the *most constrained* state to the *least constrained* state. At a less constrained state, the planner has more options and may prematurely make a decision that limits the legal options later in the search. In contrast, the forward or backward branching factor is generally small at the most constrained state, limiting the availability of poor choices. Additionally, if the constrainedness either provably or empirically decreases over time, the pool of options will grow as the difficulty decreases. Our algorithms leverage this principle, to search in directions that reduce the presence of *dead ends*, because in many extrusion problems, escaping dead ends can require an enormous amount of *backtracking* due to the long planning horizon. We begin by developing common infrastructure for both the PROGRESSION and REGRESSION algorithms.

3.5.1 Sampling extrusions

The key subroutine within each algorithm is SAMPLEEXTRUSION (Algorithm 1), which leverages PLANCONSTRAINED to sample extrusion plans for an element e. First, it samples a start node n_1 based on the currently printed nodes N_P. This governs the extrusion direction $\vec{e} = \langle n_1, n_2 \rangle$. Next, it samples an extrusion mode coparameter $\sigma_{\vec{e}} = x_o$ using SAMPLEORIENTATION. This orientation produces the initial end-effector pose $\langle p_{n_1}, x_o \rangle$ and final end-effector pose $\langle p_{n_2}, x_o \rangle$. Then, we use SAMPLEIK, an inverse kinematics procedure, to sample robot configurations q_1, q_2 that are kinematic solutions for these poses. Finally, we call PLANCONSTRAINED to find a trajectory from $q_1 \to q_2$ that satisfies mode constraints $\sigma_{\vec{e}}$ and does not collide with printed elements P.

3.5.2 Deferred evaluation

Standard state-space searches evaluate all feasible successor states $s' = \langle P \cup \{e\}, q' \rangle$ when expanding a state $s = \langle P, q \rangle$. For extrusion planning, this requires planning

Algorithm 1 Extrusion Sampling Algorithm

```
procedure SampleExtrusion(e, P; i)
    n_1 ← sample({n ∈ e | n ∈ N_P})
    {n, n'} ← e
    n_2 ← n' if n_1 = n else n
    x_o ← SampleOrientation(n_1, n_2)
    q_1 ← SampleIK(p_{n_1}, x_o); q_2 ← SampleIK(p_{n_2}, x_o)
    return PlanConstrained(q_1, q_2, x_o, P; i)
```

both an extrusion trajectory τ_e, where $q' = \tau_e(0)$, and a transit trajectory τ_t from $q \to q'$ for each remaining candidate element $e \in (E \setminus P)$. In the worst case, the number of successor (*i.e.* the branching factor) could be $\mathcal{O}(|E|)$. This is exacerbated due to the fact that SampleExtrusion and PlanMotion are both computationally expensive due to collision-checking. To mitigate this problem, we adopt a *deferred evaluation* [Hel06, RH09] strategy by planning extrusion and transit trajectories *after* popping a search node off the open list instead of *before* pushing the node on the open list. To enable this, search nodes in the open list are state and element pairs $\eta = \langle s, e \rangle$ where e serves as "action type" that specifies the next element to be extruded. This strategy dramatically reduces computation time, particularly in a greedy search, because it often avoids checking the feasibility of printing each successor element. Once a feasible successor s' is identified, the yet-to-be evaluated successors are deferred until the greedy search backtracks.

3.5.3 Heuristic tiebreakers

Because search nodes are state and element pairs, the priority function $k(s, e)$ can take the next element e into consideration. We propose priority function $k(\langle P, q \rangle, e) = \langle r(P), h(e) \rangle$ that first orders search nodes by the number of *remaining elements* $r(P) = |E \setminus P|$ and lexicographically breaks ties using a *heuristic function* $h(e)$ defined on each individual element e. By prioritizing search nodes where few elements remain to be planned, the search greedily explores the state-space in a *depth-first* manner. Because all successor states s' of state s have the same number of remaining elements r, the heuristic tiebreaker decides the order in which successors are considered. This local ordering can have strong global effects on the sequence of partially-extruded structures considered. We consider four implementations of $h(e)$: (1) *Random*, (2) *EuclideanDist* and *GraphDist*, and (3) *StiffPlan*.

3.5.3.1 Random heuristic

The *Random* tiebreaker is a baseline where ties are broken arbitrarily. It orders elements by assigning each a value sampled uniformly at random $h(e) \sim U(0, 1)$.

3.5.3.2 Distance heuristics

The *EuclideanDist* and *GraphDist* heuristics prioritize elements that are close to ground, each according to a particular geodesic. The *EuclideanDist* heuristic computes the Euclidean distance from the midpoint of element $e = \{n, n'\}$ to the ground plane. When the ground plane is the xy-plane, this is simply the z-coordinate of the element's midpoint $h_e(e) = (p_n + p_{n'})/2 \cdot [0, 0, 1]^\intercal$. The *GraphDist* heuristic computes the minimum graph distance from any ground node $n \in G$ to the midpoint of element e within the weighted frame geometric graph $\langle N, G \rangle$, where the weight of edge $e = \{n, n'\}$ is the Euclidean distance $||p_n - p_{n'}||$. We precompute these distances upfront once by calling Dijkstra's algorithm starting from the set of ground nodes G. Intuitively, both of these heuristics guide the search through structures where the element load force has a short transfer path to ground because these structures are often stiff. Additionally, these heuristics improve the sample complexity of SampleOrientation because they often ensure end-effector orientations opposite to the z-axis remain feasible.

3.5.3.3 Stiffness heuristic

The *StiffPlan* heuristic solves for a valid extrusion sequence $\vec{\psi}$, *ignoring* the robot, and uses the index j of each element e in the sequence ($\vec{\psi}[j] = e$) as its value $h_s(e) = j$. Intuitively, because $\vec{\psi}$ is known to be stiff, it attempts to adhere to $\vec{\psi}$ as closely as possible subject to the additional robot constraints. We compute a valid extrusion sequence $\vec{\psi}$ using a greedy forward search that is equivalent to Progression in Algorithm 3 if all robot planning is skipped. We use the *EuclideanDist* heuristic h_e (Section 3.5.3.2) as the tiebreaker for this search.

Algorithm 2 gives the pseudocode for PlanStiffness that implements the *StiffPlan* heuristic. It performs a greedy forward search similar to Progression in algorithm 2, with the exception that the search is finite and does not involve the robot. It uses the *EuclideanDist* heuristic h_e (section 3.5.3.2) as its tiebreaker. PlanStiffness is *complete* and will solve the extrusion sequencing problem in a finite (but not necessarily polynomial) amount of time. In the event that PlanStiffness returns **None**, the extrusion planning problem is proved to be infeasible.

Algorithm 2 Stiffness Planning Algorithm

procedure PlanStiffness(N, G, E)
    $O = [\langle\langle |E|, h_w(e)\rangle, \emptyset, e, [\,]\rangle$ **for** $e \in E$ **if** $e \cap G \neq \emptyset]$
    while $O \neq [\,]$ **do**
        $\langle r, \_\rangle, P, e, \vec{\psi} \leftarrow$ **pop**(O)
        $P' \leftarrow P \cup \{e\}$
        if not Stiff(G, P') **then**
            continue ▷ No successors
        $\vec{\psi}' \leftarrow \vec{\psi} + [e]$
        if $P' = E$ **then**
            return $\{\vec{\psi}'[j] : j$ **for** $j \in \{1, ..., m\}\}$ ▷ Solution
        for $e' \in (E \setminus P')$ **do**
            push$(O, \langle\langle r-1, h_w(e')\rangle, P', e', \psi'\rangle)$
    return None

The *EuclideanDist*, *GraphDist*, and *StiffPlan* heuristics each perform a *forward* computation from ground to produce their values. As we will see in Section 3.7.2, moving in a forward direction proves to be advantageous for satisfying the stiffness constraint. Finally, these heuristics can be seen as applying "soft" partial-ordering constraints that steer the search but do not limit completeness. This is in contrast to the hard partial-ordering constraints in prior work [WPGM16, HZH+16, HGM18] (Section 3.2).

3.5.4 Persistence

The procedures SampleExtrusion and PlanMotion use sampling-based algorithms and thus are are unable to prove infeasibility. As a result, both procedures must be reattempted indefinitely and with an increasing number of samples i. In order to ensure that Progression and Regression are probabilistically complete, they both are *persistent* [GLPK15] searches, meaning that they repeatedly expand each search node in a round-robin fashion. Let $i \geq 0$ denote the number of times a search node has been expanded. We implement persistence by simply using the pair $\langle i, k(s, e) \rangle$ as the key for search nodes in the open list O. This ensures that the search node with the fewest attempts is always expanded first. After a search node is expanded, it is re-added to the search queue O with priority $i + 1$. This search node will not be re-expanded until all other nodes in O have been expanded i times.

3.6 Progression

Algorithm 3 displays the pseudocode for Progression. Let π be the currently planned trajectories for a search node. After popping a state $\langle P, q \rangle$ and next element e from the open list O, Progression first checks whether the new structure $P' = P \cup \{e\}$ is stiff, taking advantage of the computational cheapness of Stiff. If not, the search node can be pruned altogether. Otherwise, SampleExtrusion samples an extrusion trajectory τ_e for element e. The initial configuration $\tau_e(0)$ then becomes the goal for a transit motion that is found using PlanMotion. If $P' = E$, then the structure is fully printed, and all that remains is for the robot to return to q_0. Otherwise, all remaining elements $e' \in (E \setminus P')$ are added to O as successor search nodes. Finally, search node $\langle P, q \rangle, e$ is re-added to O with sampling timeout $i + 1$ to be re-expanded in the future (Section 3.5.4). In Appendix A.2 - theorem 3, we prove Progression is probabilistically complete.

Progression is geometrically sensitive to the extrusion sequence ψ. By equation 3.3, when elements are added to $P = \{e \in \psi\}$, the collision-free configuration space $Q(P)$ weakly decreases, causing SampleExtrusion and PlanMotion to be-

Algorithm 3 Progression Algorithm

```
procedure PROGRESSION(N, G, E, Q, q_0; h)
    O = [⟨0, ⟨|E|, h(e)⟩, ⟨∅, q_0⟩, e, [ ]⟩ for e ∈ E if e ∩ G ≠ ∅]
    while True do
        i, ⟨r, _⟩, ⟨P, q⟩, e, π ← pop(O)
        P′ ← P ∪ {e}
        if not STIFF(G, P′) then
            continue                                            ▷ No successors
        τ_e ← None
        if FORWARDCHECK(E, G, P′; i) then                       ▷ Optional
            τ_e ← SAMPLEEXTRUSION(e, P; i)                      ▷ Extrusion
        if τ_e ≠ None then
            τ_t ← PLANMOTION(q, τ_e(0), P; i)                   ▷ Transit
            if τ_t ≠ None then
                π′ ← π + [τ_t, τ_e]
                if P′ = E then                                  ▷ All printed
                    τ_t ← PLANMOTION(τ_e(1), q_0, E; i)
                    if τ_t ≠ None then
                        return π′ + [τ_t]                       ▷ Solution
                s′ ← ⟨P′, τ_e(1)⟩
                for e′ ∈ (E \ P′) do
                    push(O, ⟨0, ⟨r − 1, h(e′)⟩, s′, e′, π′⟩)
        push(O, ⟨i + 1, ⟨r, h(e)⟩, ⟨P, q⟩, e, π⟩)               ▷ Persistence
```

come more constrained. In the worst case, P may prevent some of the unprinted elements $E \setminus P$ from admitting any safe extrusions. For example, Figure 3-3 demonstrates that PROGRESSION becomes trapped in a dead end near the end of the horizon because it printed the left tail of the Klein bottle (Figure 3-1) before the black diagonal element.

3.6.1 Forward checking for dead-end detection

In order to help PROGRESSION avoid making poor geometric decisions, we developed a *forward-checking* (look ahead) Algorithm [HE80, Dec03] that is able to detect dead ends earlier in the search. Intuitively, the robot must extrude every element in the structure eventually. If there is ever an element that cannot be extruded given the partially-extruded structure P, then this state is a dead end. Thus, FORWARDCHECK eagerly evaluates the viability of many successors. However, this acts oppositely to deferred evaluation (Section 3.5.2), and thus achieves better dead-end detection at

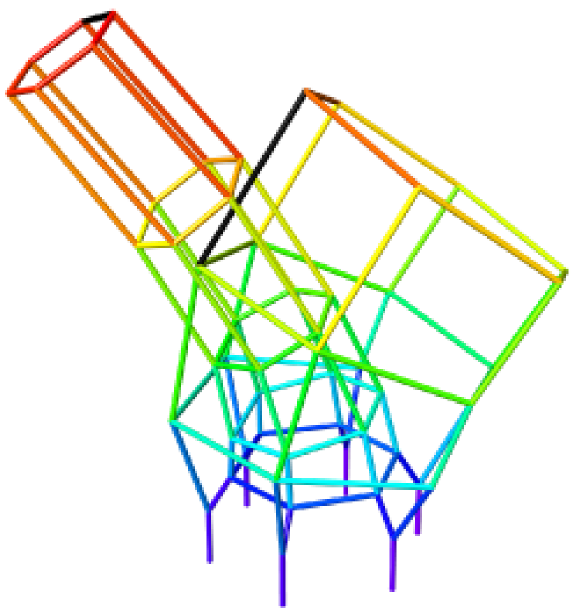


Figure 3-3: *Left*: The first state where Progression-*EuclideanDist* backtracks (black elements are unprinted). *Right*: Regression-*EuclideanDist* finds a solution without backtracking. In our structural figures, elements are colored by their index in a planned extrusion sequence. Purple elements are printed first, red elements are printed last, and black elements have yet to be printed.

the expense of worse computational overhead. As a compromise, we plan *extrusion trajectories* for only the elements e that can *currently* can be printed given P, (*i.e.* $e \cap N_P \neq \emptyset$). Intuitively, these elements are close in proximity to the printed structure and thus are most likely to be affected by a proposed geometric decision.

Algorithm 4 displays the pseudocode for ForwardCheck. It maintain a global *cache* of extrusion trajectories in order to reuse previously computed trajectories if possible. Because ForwardCheck invokes SampleExtrusion, it cannot prove that a search node is a dead end. Thus, ForwardCheck also uses the increasing sampling timeout i to search for longer extrusion trajectories. Figure 3-4 demonstrates an instance where ForwardCheck detects, and thus avoids, a dead end early in the search. The element with the pink sphere is the candidate element e to be printed. However, printing e prevents the diagonal black element from being printable. As a result, the search defers expanding e at this time.

ForwardCheck performs a one-step look ahead to detect dead ends. However,

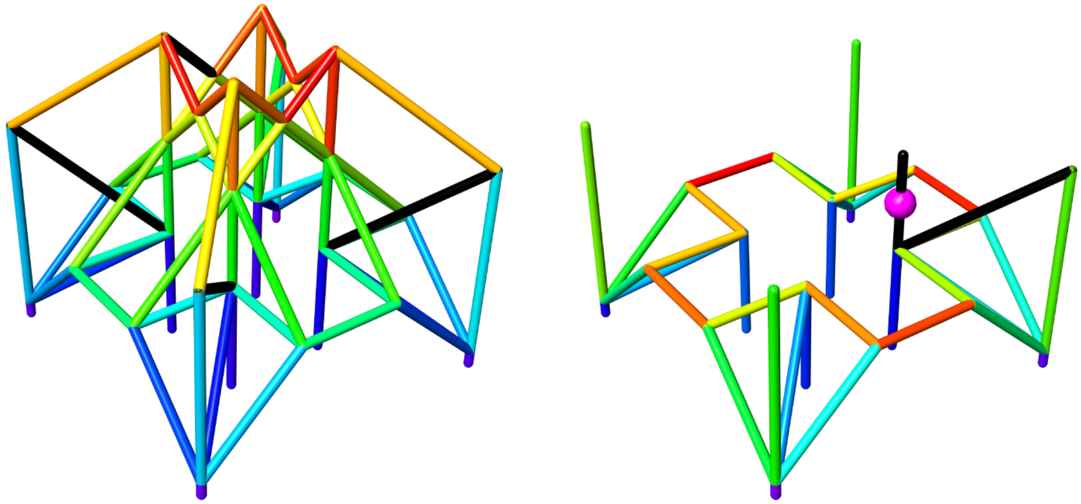


Figure 3-4: *Left*: the first state where PROGRESSION-*GraphDist* backtracks (black elements are unprinted). *Right*: FORWARDCHECK detects that printing the element indicated by the pink sphere prevents the diagonal black element from being safely extrudable.

it might the case that while each element can be printed individually, a *pair* of elements together cannot be printed. If so, FORWARDCHECK will not be able to detect the dead end until much later in the search, such shown in Figure 3-5. Here, extruding any black element prevents at least one other nearby element from being safely printable. An *arc-consistency* look ahead that considers pairs [SF94] could detect these cases at the expense of even greater expansion overhead.

Algorithm 4 Forward Checking Algorithm

procedure FORWARDCHECK$(E, G, P; i)$
    cache $\leftarrow \{e : [\,]$ **for** $e \in E\}$ ▷ Global cache
    for $e \in (E \setminus P)$ **do**
        if $e \cap N_P = \emptyset$ **then** ▷ Printable
            continue
        if any(SAFE(τ_e, P) **for** $\tau_e \in$ cache$[e]$) **then**
            continue ▷ Reuse existing
        $\tau_e \leftarrow$ SAMPLEEXTRUSION$(e, P; i)$ ▷ Extrusion
        if $\tau_e =$ **None then**
            return False
        cache$[e] \leftarrow$ cache$[e] + [\tau_e]$
    return True

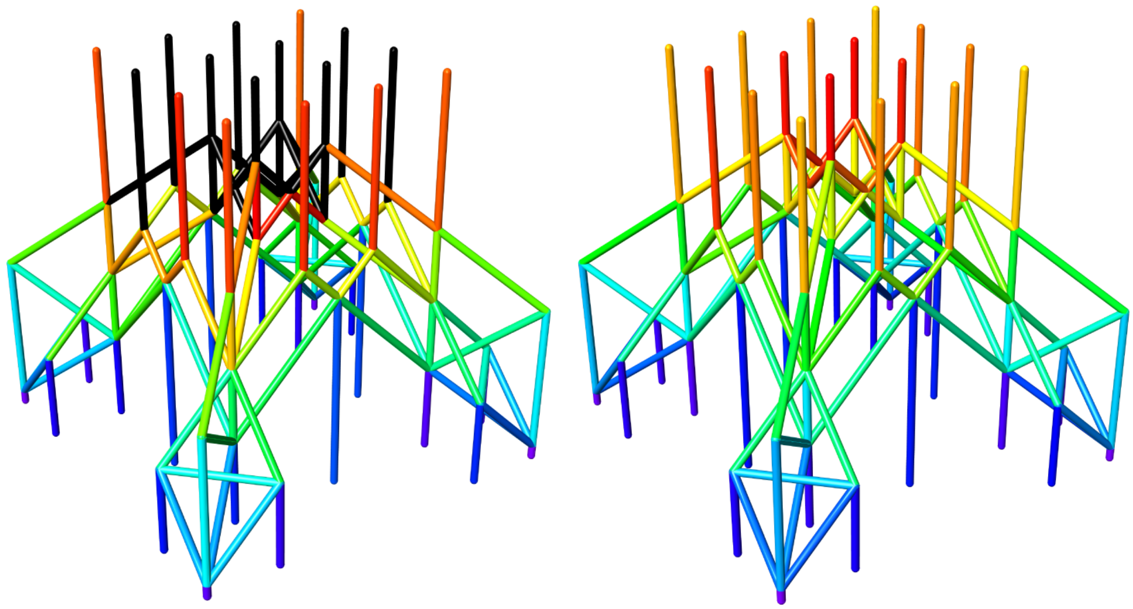


Figure 3-5: *Left*: the first state where ForwardCheck-*GraphDist* backtracks (black elements are unprinted). *Right*: Regression-*EuclideanDist* finds a solution without backtracking.

3.7 Regression

Regression performs a backward search from the goal state to the initial state [Nil14, Wel94, McD91, GNT04]. In many planning domains, the goal conditions are under-specified, and as a result, there are many goal states. Because of this, the initial branching factor can be quite large. Furthermore, some goal states might not be reachable from s_0, creating more opportunities for dead-end branches [BG01]. Because extrusion planning has a single goal state s_*, these problems are avoided. Algorithm 5 displays the pseudocode for Regression. The key differences from Progression in Algorithm 3 are that we negate $-h(e)$ in order to expand elements in the reverse order, the final extrusion configuration $\tau_e(1)$ is the start of each transit motion planning problem, and trajectories $[\tau_e, \tau_t]$ are prepended to plan π. In Appendix A.2 - theorem 3, we prove Regression is probabilistically complete.

Algorithm 5 Regression Algorithm

procedure REGRESSION($N, G, E, \mathcal{Q}, q_0; h$)
  $O = [\langle 0, \langle |E|, -h(e) \rangle, \langle E, q_0 \rangle, e, [\,] \rangle$ **for** $e \in E]$
  while True do
    $i, \langle r, \_ \rangle, \langle P, q \rangle, e, \pi \leftarrow$ **pop**(O)
    $P' \leftarrow P \setminus \{e\}$
    if not STIFF(G, P') **then**
      continue ▷ No successors
    $\tau_e \leftarrow$ SAMPLEEXTRUSION($e, P'; i$) ▷ Extrusion
    if $\tau_e \neq$ **None then**
      $\tau_t \leftarrow$ PLANMOTION($\tau_e(1), q, P; i$) ▷ Transit
      if $\tau_t \neq$ **None then**
        $\pi' \leftarrow [\tau_e, \tau_t] + \pi$
        if $P' = \emptyset$ **then** ▷ All printed
          $\tau_t \leftarrow$ PLANMOTION($q_0, \tau_e(0); \emptyset, i$)
          if $\tau_t \neq$ **None then**
            return $[\tau_t] + \pi'$ ▷ Solution
        $s' \leftarrow \langle P', \tau_e(0) \rangle$
        for $e' \in P'$ **do**
          push$(O, \langle 0, \langle r-1, -h(e') \rangle, s', e', \pi' \rangle)$
    push$(O, \langle i+1, \langle r, -h(e) \rangle, \langle P, q \rangle, e, \pi \rangle)$ ▷ Persistence

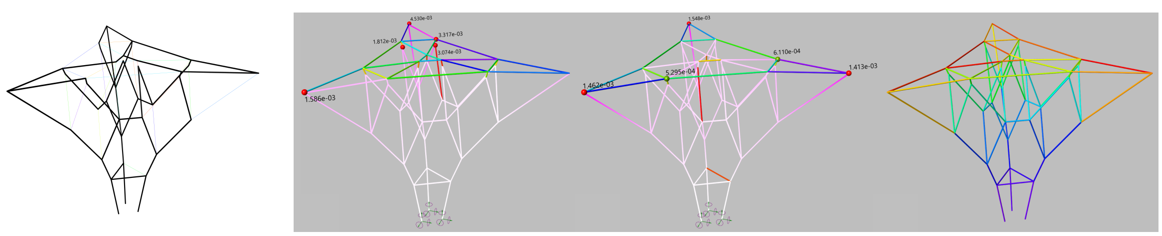


Figure 3-6: *From left to right*: 1) the unassigned substructure at the first state where REGRESSION-*Random* backtracks. 2) the first state where REGRESSION-*EuclideanDist* backtracks. The element *deflection* is colored from white to pink. The five most deformed nodes are red and their translational displacements are annotated in meters 3) the first state where REGRESSION-*GraphDist* backtracks 4) REGRESSION-*StiffPlan* finds a solution without backtracking.

3.7.1 Geometric constraints

REGRESSION can be seen as *deconstructing* the structure by sequentially removing elements. From equation 3.3, removing an element weakly increases the collision-free configuration space $Q(P)$. Thus, the robot is the most geometrically constrained at the beginning of the search, limiting which elements can be initially extruded.

As a result, REGRESSION's options with respect to geometry increase as the search advances, preventing it from being trapped in a geometric dead end. To motivate using backward search to efficiently satisfy geometric constraints, we analyze a simplified *geometry-only* version of the extrusion problem that both omits stiffness and transit constraints as well as assumes a given set of possible extrusion trajectories T. Given these simplifications, extrusion planning simply requires identifying a totally-ordered subset of T that extrudes each element exactly once. We consider a modified version of REGRESSION in Algorithm 5 for extrusion-only problems. Trivially, for all inputs, let $\text{STIFF}(G, P) = \textbf{True}$ and $\text{PLANMOTION}(q, q', P; i) = [q, q']$. Additionally, $\text{SAMPLEEXTRUSION}(e, P; i) = \textbf{sample}(\{\tau_e \in T \mid \text{SAFE}(\tau_e, P)\})$ arbitrarily selects a safe trajectory $\tau_e \in T$ for element e if one exists. Otherwise, **sample** returns **None**. Under these conditions, REGRESSION will solve feasible problem instances in polynomial time (see theorem 1 in Appendix A.1).

3.7.2 Stiffness constraints

Although REGRESSION makes geometric planning easier, it increases the difficulty of satisfying the stiffness constraint. At the beginning of the backward search, there are many elements that can be removed without violating the stiffness constraint. However, later in the backward search (closer to the structure's supports), there are fewer opportunities for supporting the structure, making the search more likely to arrive at a dead end caused by stiffness. Figure 3-6 image 1) shows the remaining-to-be-printed structure at the first dead end encountered by REGRESSION-*Stiffness*. As can be seen, arbitrarily removing elements sparcifies the structure and reduces its structural integrity. To combat this, we use the heuristic tiebreakers in Section 3.5.3 to bias the search to remain stiff.

To understand the impact of these tiebreakers, we experimented on the extrusion problems in Section 3.8, comparing the success rate of the PROGRESSION and REGRESSION algorithms when *only* the stiffness constraint is active (*i.e.* ignoring the robot). For PROGRESSION, this is equivalent to PLANSTIFFNESS in Section 3.5.3.3. We performed 6 trials per algorithm, heuristic, and problem. Each trial had a 5 minute timeout. Figure 3-7 displays the success rate of each algorithm. PROGRESSION was able to find an extrusion sequence for all problems, regardless of the heuristic. REGRESSION failed around 40% of the time when randomly breaking ties. However, REGRESSION was able to solve all problems when using the *StiffPlan* heuristic; although, this is not surprising given that *StiffPlan* explicitly uses a stiff plan. The *EuclideanDist* and *GraphDist* heuristics perform quite well but still have failure cases, such as in Figure 3-6. There, both heuristics prioritize removing the top of the structure, which is designed to provide tensile forces to hold the cantilevered elements [Lee18], causing the red vertices to deform significantly.

3.8 Results

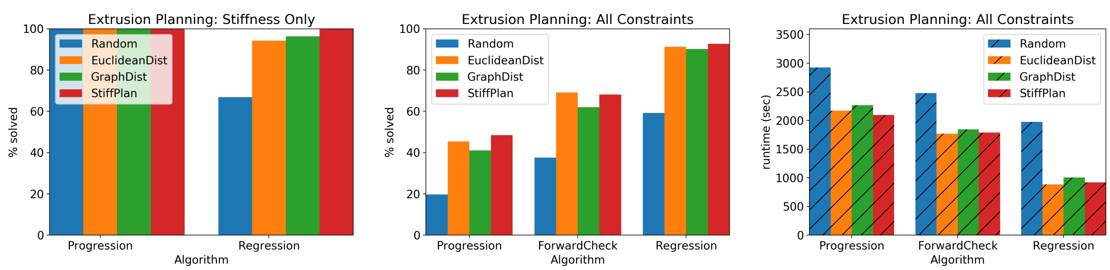


Figure 3-7: *Left*: the success rate of each algorithm (except FORWARDCHECK) and heuristic pair subject to *only the stiffness constraint*. *Center*: the success rate of each algorithm and heuristic pair. *Right*: the average runtime in seconds of each algorithm and heuristic pair with a timeout of 1 hour (3600 seconds).

We experimented on 41 extrusion problems with up to 909 elements (the duck problem in Figure 3-1). See the extended manuscript[1], for a picture of each problem. We experimented using all combinations of our 3 algorithms (PROGRESSION, FORWARDCHECK, and REGRESSION) and 4 heuristics (*Random*, *EuclideanDist*, *GraphDist*, and *StiffPlan*). We performed 4 trials per algorithm, heuristic, and problem, each with a 1 hour timeout. We used PyBullet [Cou15, CB16] for collision checking, forward kinematics, and rendering. Because each element can only be in one pose, we preprocess the structure by computing a single, static axis-aligned bounding box (AABB) bounding volume hierarchy (BVH) [Eri04, KIS+12] for use during broadphase collision detection with each robot link. We implemented PlanMotion using RRT-Connect [KJL00], SampleIK using IKFast, an analytical inverse kinematics solver [Dia10], and PlanConstrained using Randomized Gradient Descent (RGD)[YG07, Sti10]. See https://github.com/caelan/pb-construction for implementations of our algorithms. All the tested problem instances are available at https://github.com/yijiangh/assembly_instances.

Figures B-1, B-2, and B-3 display the extrusion problems that we considered. For each problem, we ran one trial of Regression+*StiffPlan* and recorded the extrusion sequence it produced. For successful trials, elements are colored by their index in a extrusion sequence, where purple elements are printed first and red elements are printed last. All elements in the structure are black an unsuccessful trial. Some problems are the result of a linear transformation, such as a rotation or scaling, applied to the same original frame structure. Other problems are discretized version of the same object but with varying degrees of topological complexity.

Figure 3-7 displays the success rate (*Center*) and the average runtime (*Right*) for each algorithm. We assign a runtime of 1 hour for trials that failed to find a solution. The *EuclideanDist*, *GraphDist*, and *StiffPlan* heuristics outperform *Random*, regardless of the algorithm. The improved performance for both Progression and Regression indicates that the heuristics provide both stiffness and geometric guidance. ForwardCheck is able to solve more problems than Progression, indicating that it is able to avoid some dead ends. However, ultimately Regression performed the best in terms of both success rate and runtime. The best performing heuristic was *StiffPlan* followed closely by the *EuclideanDist*. Our best-performing algorithms are able to solve around 92% of the problems and have an average runtime of about 15 minutes. Figure 3-8 displays the runtime of each trial per problem size when each algorithm uses the *EuclideanDist* heuristic. Although ForwardCheck is able to solve more problems than Progression, it comes at the expense of longer runtimes.

We experimented on two extrusion problems considered by Choreo [HGM18].

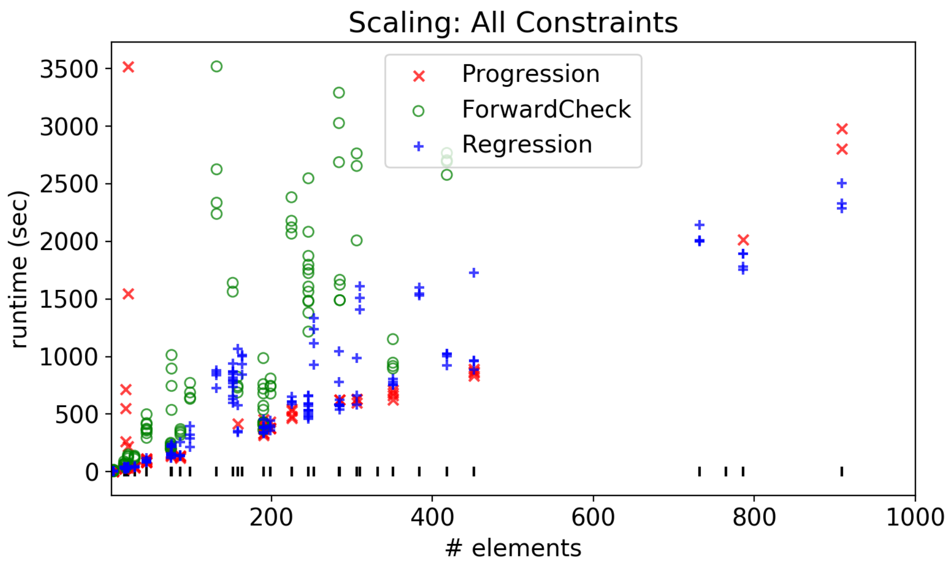


Figure 3-8: The runtime of each algorithm when using the *EuclideanDist* heuristic. The x-axis ticks denote the distribution of problem sizes.

Choreo solves the "3D Voronoi" and "Topopt beam (small)" problems in 4025 and 3599 seconds whereas Regression-*EuclideanDist* solves the problems in 742 and 2032 seconds. Our planner outperforms Choreo despite the fact that Choreo had access to additional, human-specified information (Section 3.2). We validated our approach on three real-world extrusion problems. See https://youtu.be/RsBzc7bEdQg for a video of our robot extruding each structure. The largest of the three is the Klein bottle (Figure 3-1), which took about 10 minutes to plan for and 6 hours to print.

3.9 Conclusion

We investigated 3D extrusion planning using a robot manipulator. Here, structural constraints are often at odds with geometric constraints. Our algorithmic insight was to use backward search to plan geometrically feasible trajectories and to use forward reasoning as a heuristic that guides the search through structurally-sound states. Future work involves extending our approach to general-purpose construction tasks.

Chapter 4

Unified sequence and motion planning for robotic additive construction of bar structures

In this chapter, we generalize the planning algorithms that we developed in the last chapter on extrusion planning to demonstrate how to solve a variety of additive assembly planning problems using the same search strategy. As a result, we obtain a unified approach for modeling and solving sequence and motion planning for generic robotic assembly problems that involve a repetitive robotic action pattern [1].

4.1 Introduction

Robotics-based fabrication promises to substantially improve construction in the built environment, offering benefits such as speed, quality, material efficiency, worker safety, and eventually cost reduction. In addition, robotic additive construction opens up new fabrication possibilities beyond automating manual processes, leading to an expansion of design possibilities and formal expressions. Recent years have started to see these decades-old promises finally come to fruition, primarily in the form of impressive prototypes, proofs-of-concept, and demonstrator projects

[1] A version of this chapter has been published in [HGT+21].

at the "pavilion" scale, showcasing the potential of robotic fabrication and assembly in architecture. However, key challenges still remain to be tackled before the impact of these technologies are broadly felt on real construction sites. This paper addresses one such challenge: automated methods for planning and programming the robotic systems that will assemble building systems. In particular, this paper focuses on a broad class of building structures, bar structures, which are composed of discrete linear elements and used broadly and in great variety in the built environment. Trusses, frames, and more complex structural hierarchies involving beams and columns are all examples of bar structures, and can use a range of materials such as metals, timber, precast concrete, plastics, etc.

3D bar structures can have complex geometries and topologies, for reasons related to aesthetics, structural efficiency, material availability, or site constraints. Because of this complexity, robotic assembly, i.e. using industrial robotic arms to pick and place structural elements, is an attractive construction approach due to the dexterity and geometric range of industrial robots; there is the potential to precisely and efficiently construct highly tuned structures designed in response to a range of priorities and constraints. However, because of this design complexity, the challenge of robotically assembling such structures is high, and one can not typically fall back on rules of thumb or previous solutions [EGK17, SAE+16]. Previous work has demonstrated great potential for additive construction of bar structures using robots with many degrees-of-freedom (DOFs) [GMW14, WKB+16], but also significant hurdles: planning the sequence and motion of these robots must simultaneously account for collision avoidance in dense bar networks and structural stiffness during construction. These computational challenges, coming from both the high DOFs of the machine and the large-scale structural behavior of the partially constructed structures, distinguish robotic additive construction from other layer-based additive manufacturing techniques. While transitioning between a volumetric digital design model and machine code for a 3-axis gantry machine is relatively straightforward, searching for a feasible construction sequence and trajectories for robotic arms is much less obvious, and thus requires a different planning strategy and new design methodologies.

Recent previous research has proposed scalable planning algorithms to address these constraints in the context of robotic spatial extrusion [GHLPM20], finally empowering solutions to large robotic planning problems without the use of human intervention. However, more generalized assembly planning for additively assembled structures has not been addressed.

In this paper, we generalize previous work on extrusion planning [GHLPM20] to demonstrate how to solve a variety of additive assembly planning problems using the same search strategy. In particular, we use the concepts of an *action template* and *plan skeleton*, which together describe a sequence of robot's motion primitives in these two different applications. Formalizing additive construction methods such as extrusion and assembly planning through the lens of a plan skeleton allows us to use the same planning algorithm to solve a range of construction problems efficiently, improving automation of additive construction planning in general. We demonstrate the effectiveness of our approach through simulation and real-world construction examples.

4.2 Related work

Automating the construction of bespoke, irregular bar structures using programmable robots has been studied in the rapidly advancing field of architectural robotics in the past two decades, specifically due to a robot's capacity for performing precise spatial movements. Spatial extrusion and robotic assembly are the two main types of methods that use a robot to directly distribute (by extruding or positioning) individual, often standardized linear elements in positions designated by the structure's design. Existing pilot studies have demonstrated the great potential of deploying this technology to the scale of a building [EGK17, HL14, HWT+15, TAH+18], allowing both formal variations [SRJG17, YMYZ16] and structural efficiency [SAE+16, TMG+18]. However, much of this early work adopts a trial-and-error method for planning, by manually conjecturing a construction sequence and end effector workspace poses for the robot. Software packages exist to support these methods by performing point-wise kinematics checks [BBC11, Sch12] or configuration-to-configuration motion planning [GPR+18, SC18]. However, these tools currently require a sub-optimal manual planning process because they cannot reason about the construction sequence and trajectories simultaneously. This manual requirement limited early work to bar structures with repetitive topological patterns.

Several sequence and motion planning algorithms have been proposed to robotically extrude bar structures with arbitrary geometry and topologies. Early work in this direction addressed planning for a free-flying hotend end effector [HZH+16, WPGM16, YHL+16]. By rigidly committing to a partial ordering on the construction sequence, these algorithms are incomplete. Gelber et al [GHB18] presented a complete forward search algorithm for a 3-axis 3D printer that constrains the deformation of the structure but does not address higher-DOF systems. Choreo was the first extrusion planning system that uses a robot manipulator to make robotic extrusion planning for arbitrary topologies possible [HGM18]. However, Choreo separates the extrusion planning into separate sequence and transit planning phases, which can prevent the discovery of a feasible solution in certain cases. Additionally, to make the sequence planning tractable, Choreo requires a user-specified ordering on elements to guide the search algorithm.

In most prior work in robotic assembly, the construction of bar structures has also followed a layer-based approach, simplifying particularly the sequence planning and reachability challenge such that it can be intuitively solved [AKGK16]. Recent projects have demonstrated robotic assembly for more complex geometries [SAE+16, HKK+17], but motion planning has remained a key challenge, addressed by pre-defining the robot's path through a trial and error process. To construct large-scale differentiated space-frame structures, recent research has focused on integrating motion planning and sequence definition into the design process. One approach to addressing the duality of sequence- and motion-planning is to define the sequence of a spatial assembly simultaneously with the geometry generation, within the design process. Parascho et al. [PKC+18] propose a constructive system relying on tetrahedral configurations, which, if assembled in the order that the bars are generated, provides local structural stability and support during construction. While this method removes the sequencing challenge from the fabrication process, it still relies on motion planning algorithms to identify collision-free trajectories, which are not guaranteed to be readily found. In order to find such trajectories, a trial-and-error process was needed to iterate through multiple design possibilities that alter both the sequence and the geometry, making the design and planning process very time-consuming.

In the robotic planning literature, task and motion planning (TAMP) involves planning both high-level objectives as well as low-level robot motions required to complete a multi-step manipulation [GLPK18b, SFR+14, Tou15]. For extrusion and assembly problems, the high-level decisions involve the construction sequence, and the low-level motions are the trajectories for motion primitives, like extrusion, pick, place, transit and transfer. Although TAMP includes a much broader class of problems than extrusion and assembly for construction, a key challenge in extrusion and assembly problems is that they often require manipulating many more objects, which, as a result, leads to substantially longer planning horizons. Solutions to most TAMP benchmarks involve fewer than 30 high-level actions while many construction problems may require assembling several hundreds of elements [Lag16]. Readers are referred to Garrett et al. [GCH+21] for an extensive review of the work in this area.

Some notable works consider construction sequencing with a structural stability constraint but in the absence of a robot manipulator. Beyeler et al. [BBW15] proposed a backward search algorithm to find stable deconstruction sequences of masonry structures. Deuss et al. [DPW+14] presented a divide-and-conquer algorithm to compute cable-assisted, self-supporting construction sequences for masonry structures. Connecting such approaches with robotic assembly constraints is more challenging algorithmically but necessary to support automated robotic assembly planning. The research goal of this paper's work is to build upon existing material systems and knowledge from the architectural robotics literature and algorithmic insights from the robotic planning literature, to develop generalized planning algorithms that automate the tedious task of programming robots to additively construct various bar structures.

4.3 Robotic additive construction

In this work, we provide a unified formulation of robot-enabled additive construction. Although we focus on spatially extruded systems and bar assembled systems, our framework can be applied to other additive construction tasks if an appropriate set of primitives are specified. This formulation allows us to describe and solve problems involving different kinds of systems using the same high-level algorithms. In this section, we show that, at a high-level, the same constraints are in play during both extrusion and assembly. Because of this, we only need to implement several manipulation-specific motion primitives to plan in different construction settings. We first describe two types of constraints: structural (Section 4.3.2) and robot-involved geometric constraints (Section 4.3.3). We use the concepts of action templates and plan skeletons as modeling abstractions that represent motion primitives in different applications. Then, we develop planning algorithms that are able to find feasible construction and robotic motion plans for any plan skeleton, which in turn results in a plan for the underlying bar structure construction task (Section 4.4).

4.3.1 Bar systems

A bar system can be described as a collection of n linear elements E. Each linear element corresponds to a bar. Each element can be connected to one or more other elements. In spatial extrusion, these linear elements are solid plastic cylinders and are connected exactly at the endpoint of the linear element (Fig. 4-1-left). At the connection point, an extra sphere of extruded plastic is applied to form a ball joint-like connection among elements after solidification. In bar assembly, elements can be connected anywhere along the boundary of the element (Fig. 4-1-right). In this case, extra adhesive material is applied between elements to form a rigid connection, such as welding metals, glue, etc. We assume that the connectors can transfer moment and torsional load to give the partially constructed structure the ability to cantilever, which is a reasonable assumption given the rotational stiffness of these joints. We assume that the robot does not exert additional external forces on the structure, and thus each partial structure only experiences forces induced by self-weight. Because of this, structural constraints only involve the partial construction, but geometric constraints involve the interaction between the robot and the structure it is building.

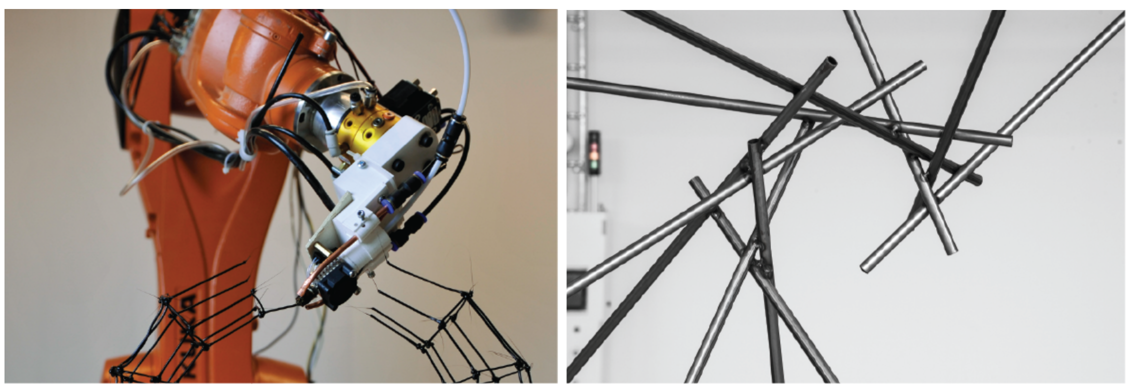


Figure 4-1: Connection examples between bar elements; *left*: two extruded plastic bars are connected exactly at their shared end point [GHLPM20]; *right*: Point-welded connections between steel bars [Par19].

For a bar system, we define a *construction sequence* to be an ordering of the elements that dictates the sequence for which they added to the structure. In the scope of this work, elements are not allowed to be removed and later re-introduced once they are placed. Furthermore, we do not allow extra scaffolding elements to be introduced. We call such a construction process a *monotonic* additive construction process because the structure only grows over time.

A construction sequence is *valid* if every partial structure along the sequence satisfies both the structural constraints that limit the deformation of the partial construction and the geometric constraints induced by the behavior of the robotic manipulator. Such geometric constraints ensure that the robot stays collision-free with the partial construction, as well as restricting its trajectory to follow certain motion primitives.

In the following sections, we will describe the details of the structural constraints and robotic geometric constraints. We shall see that through the lens of action templates and plan skeletons, we can unify the treatment of both extrusion and assembly planning under the same framework, and reduce their implementation as a switchable module in our general backward state-space search algorithm.

4.3.2 Structural constraints

In this paper, we focus on deformation-based stiffness constraints as the key structural criterion. A stiffness constraint requires that a partially constructed structure's maximal nodal deformation is below a given tolerance. Each bar element experiences a self-weight load due to gravity, which causes the structure to bend. Excessive deformation is undesirable since it leads to connection failures for subsequently constructed elements. The deformation of all nodes is calculated using a first-order linear elastic finite element analysis (FEA) of a 3D frame structure [MGZ99], and the norm of the x, y, z translational deflections at each node is compared to a maximum permitted tolerance. For extruded structures, the central axes of the extruded elements are directly used as linear beam elements. For assembled structures, there are two types of structural members, both modeled with linear beam elements: (1) bars which represent the rods and (2) connectors which represent the welded or glued connection between two bars [Par19].

4.3.3 Geometric constraints

The second key type of constraint involves the geometric interaction between the robot, the environment, and each partially constructed structure. First, the robot must always respect common motion constraints such as staying within its joint limits as well as avoiding collisions with itself, the environment, and the currently assembled elements. In addition, different types of motion primitives impose additional task-space constraints on the robot. In this section, we will examine several motion primitives and discuss the constraints they induce in extrusion and assembly problems.

We introduce the following terminology to discuss different motion primitives using the same language. Let *action template* be a parameterized robot skill that describes a type of robot motion. Additionally, let a *plan skeleton* be a sequence of action templates [GCH+21]. For extrusion and assembly, each plan has a corresponding plan skeleton that follows a known repetitive pattern and has a known fixed length. Through formalizing the robot's skill primitives and corresponding action templates, we identify constraints for different primitives and unify our algorithmic treatment of both extrusion and assembly planning.

In an extrusion process, the robot only has two skill primitives: extrude and transit. The robot alternates between performing extrude primitives, where the robot is extruding material, and transit primitives, where the robot is moving to another element instead of extruding. The plan skeleton for extrusion is an alternating sequence of the transit and extrude templates (Fig. 4-2 -left). Detailed descriptions of the constraints involved in these action templates and the primitive planners involved can be found in Garrett et al. [GHLPM20].

In an assembly process, the robot repetitively performs the following four skill primitives in order: (1) *transit*, (2) *pick*, (3) *transfer*, and (4) *place*. Unlike extrusion, where materials are distributed directly out of the hotend end effector, assembly involves re-positioning ready-made elements. Typically, the elements are initially placed on a material rack and must be picked up; however, in some applications, a human operator instead manually lifts and positions new elements in the robot's gripper. The pick and place primitives attach and detach the element, and the transit and transfer primitives move between each pick and place. The pick and place primitives both have two sub-procedures, approach and retreat, which move the robot in and out of contact with the element being manipulated.

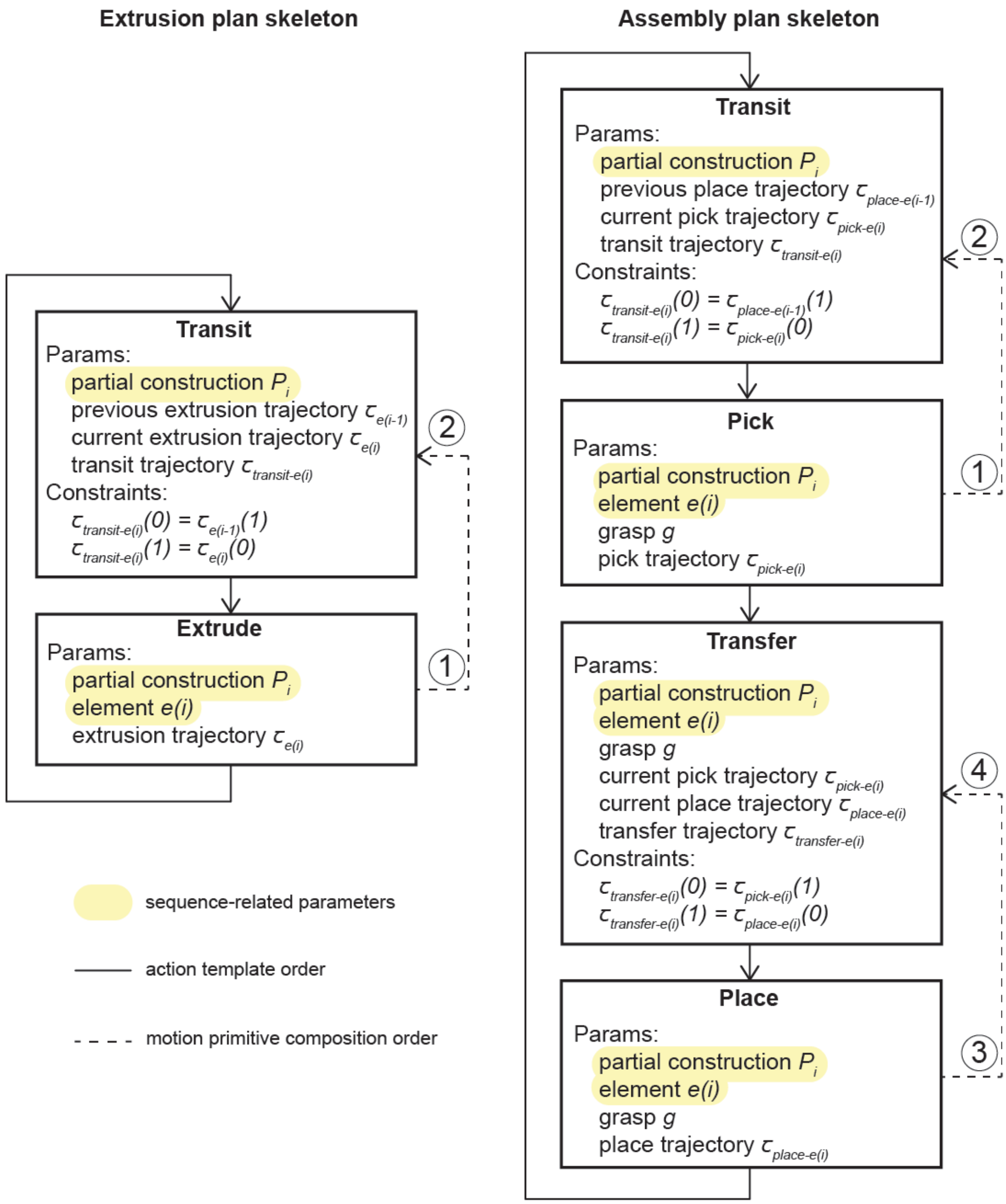


Figure 4-2: The plan skeleton for extrusion (*left*) and assembly (*right*). On each iteration $i \in \{1, ..., n\}$ of the plan skeleton, a new element $e(i) \in E$ is constructed where $e(i)$ is selected by a planner.

In a pick primitive, the robot approaches an element on the material rack (*pick-approach* in Fig. 4-3), grasps it, and moves it out of contact with the rack (*pick-retreat* in Fig. 4-3). The parameters for the pick action template include: (1) the partial structure P that specifies the elements that the robot must not collide with, (2) the element e being picked up, (3) a grasp pose g that describes the relative transformation between the robot's end effector and the element when attached, and (4) a pick trajectory τ_{pick-e}, which is the concatenation of the pick-approach and pick-retreat trajectories.

In a place primitive, the robot inserts the element into the structure (*place-approach* in Fig. 4-3), detaches it, and moves out of contact with the structure (*place-retreat* in Fig. 4-3). Because the held element during an insertion motion is in the close vicinity of the assembled elements, the place-approach motion usually requires the path of the grasped element to involve both translation and rotation. In contrast, the place-retreat motion is often just a translational motion that moves the end effector out of contact with the element after detachment. With the exception of the place trajectory parameter $\tau_{place-e}$, the place action template shares the same parameter values as its corresponding pick action template, including the partial structure P, the element e, and the grasp pose g.

The robot moves between two configurations using the transit and transfer primitives. In a transit primitive, the robot's end effector is empty. However, in a transfer primitive, the robot is grasping an element at the relative pose specified by the grasp pose parameter g of the adjacent pick and place action templates. Transfer primitives impose two additional constraints that (1) the grasped element remains rigidly attached to the robot's end effector and (2) the grasped element does not collide with the robot nor the currently built structure. As a result, the grasped element can be seen as a temporary component of the robot, and because it only increases the volume occupied by the robot, it decreases the collision-free configuration space of the robot. The transit action template's parameters include (1) the partial structure P, (2) the place trajectory $\tau_{place-e'}$ from the previous construction step, (3) the pick trajectory τ_{pick-e} from the current construction step, and (4) a tran-

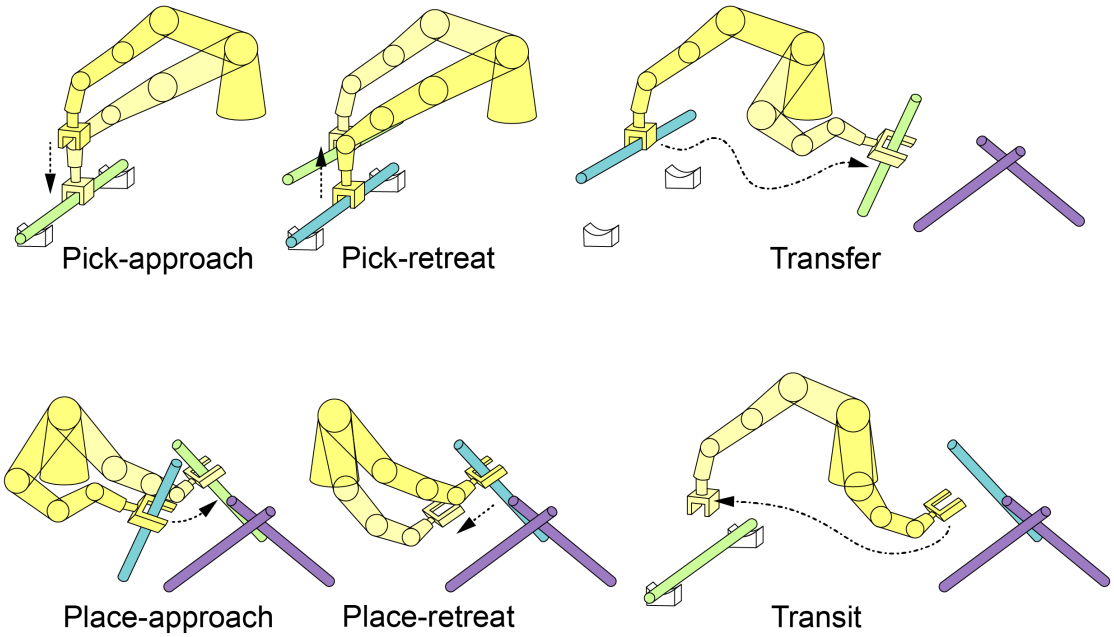


Figure 4-3: Motion primitives for assembling a single bar element.

sit trajectory $\tau_{transit-e}$. The transfer action template's parameters are (1) the partial structure P, (2) the grasped element e, (3) the grasp pose g, (4) the current pick trajectory τ_{pick-e}, (5) the current place trajectory $\tau_{place-e}$, and (6) a transfer trajectory $\tau_{transfer-e}$.

The plan skeleton for assembly is a repeating sequence of the four action templates in order: transit, pick, transfer, and place (Fig. 4-2-right). Within each iteration, a planner must select the partial construction P, assembled element e, and grasp pose g as well as valid robot trajectories. Similar to extrusion, because of parameter dependence, although the transit and transfer action templates come *before* pick and place in the plan skeleton, their parameters are determined *later* during planning.

Transit and transfer primitives can be implemented using any standard motion planner, such as the rapidly-exploring random tree (RRT) algorithm [LaV98]. Additionally, any constrained motion planner can be used to find collision-free paths for the pick and place primitives [BSK11, KMK19b, Sti10].

4.4 Planning for robotic construction

In Section 4.3, we observed that both extrusion and assembly require a planner to (1) find a structurally feasible construction sequence for the bar elements and (2) compute robotic motions for the corresponding action primitives while constructing each element. In this section, we will describe state-space planning algorithms for additive robotic construction. Additionally, we will describe why searching backward helps the algorithm avoid geometric deadends, which improves the planner's performance. Finally, we will explain the impact of different search heuristics on a planner's performance.

One common planning strategy is to hierarchically decompose (1) selecting the construction sequence and (2) planning robot motions into two separate sequential planning problems [HGM18]. In this strategy, the robot is not considered at all when planning the construction sequence in step (1), and the construction sequence is fixed when planning robot motions in step (2). However, excluding geometric constraints from the construction sequence planning can often lead to construction sequences that are infeasible for a robot to construct, causing the planner to fail [HGM18]. Instead, by jointly planning the construction sequence and robot motions, we can develop planning algorithms that are complete, *i.e.* are guaranteed to find a correct solution if one exists.

4.4.1 State-space planning algorithms

State-space planning algorithms search over possible sequences of world states. Here, a state represents the status of a step in the construction process. A state includes (1) the partially constructed structure (2) the robot's current configuration, and (3) any values pertaining to active constraints imposed by the current motion primitive, such as the constant orientation of the end effector for extrude motions and the rigid grasp pose of an element for transfer motions. Because both the *discrete* constructed structure and the *continuous* robot configuration are simultaneously parts of the state, the state is *hybrid*. Furthermore, the set of possible states is infinitely large, whereas the set of states when only considering sequencing is finite.

A state-space planning algorithm can either search *forward* in time starting from the initial state or *backward* in time from a goal state. In the context of construction, a forward search starts from the unbuilt structure and sequentially adds elements. Thus, it can be seen as exploring different ways of *constructing* the structure. In contrast, a backward search starts from the completed structure and sequentially removes elements, thus *deconstructing* the structure. We discuss the trade-offs when moving forward versus backward in Section 4.4.2.

4.4.2 Backward search

Forward search is the most intuitive way to search because it mimics how plans will be executed in the real world. However, our previous work on robotic extrusion planning found that forward search encounters many *dead ends*, states that do not lead to any plan, due to geometric constraints [GHLPM20]. For example, a state in the forward search might have the outer layer of the structure constructed with the interior remaining. Because the outer layer prevents the robot from reaching any element in the interior without colliding, the search cannot progress from this state. However, in a backward search, elements are removed instead of added, so the search could instead peel-off the exterior in order to reach the interior. As a result, we found that searching backward can significantly reduce the number of states explored to find a plan and thus dramatically reduce the computation time.

Fig. 4-4 illustrates an example of how the backward state-space search algorithm operates. The backward search starts from the completed structure state at the top rectangle (Fig. 4-4-a) and works its way downward towards the initial unbuilt state at the bottom (Fig. 4-4-d) through intermediate partially constructed states, which are circles. The partially constructed structure P is illustrated right next to each state. Every time the algorithm selects the next element to remove, it must evaluate the new partial construction's structural feasibility (Fig. 4-4-b) as well as geometric constructability (Fig. 4-4-c). Specifically, it is geometrically constructable if the motion primitive planners for the action templates in Section 4.3.3 succeed in finding trajectories. To proceed towards the next state, the algorithm must choose an element among the candidate elements that have not yet been removed, according to a ordering induced by the heurstic function $h(e)$ (illustrated as a normalized number $0 \leq h \leq 1$ next to each transition arrow between states in Fig. 4-4). If there is no feasible successor state, the algorithm encounters a dead end, and it will *backtrack* to the previous state, undoing the previous deconstruction action, to explore other options from there.

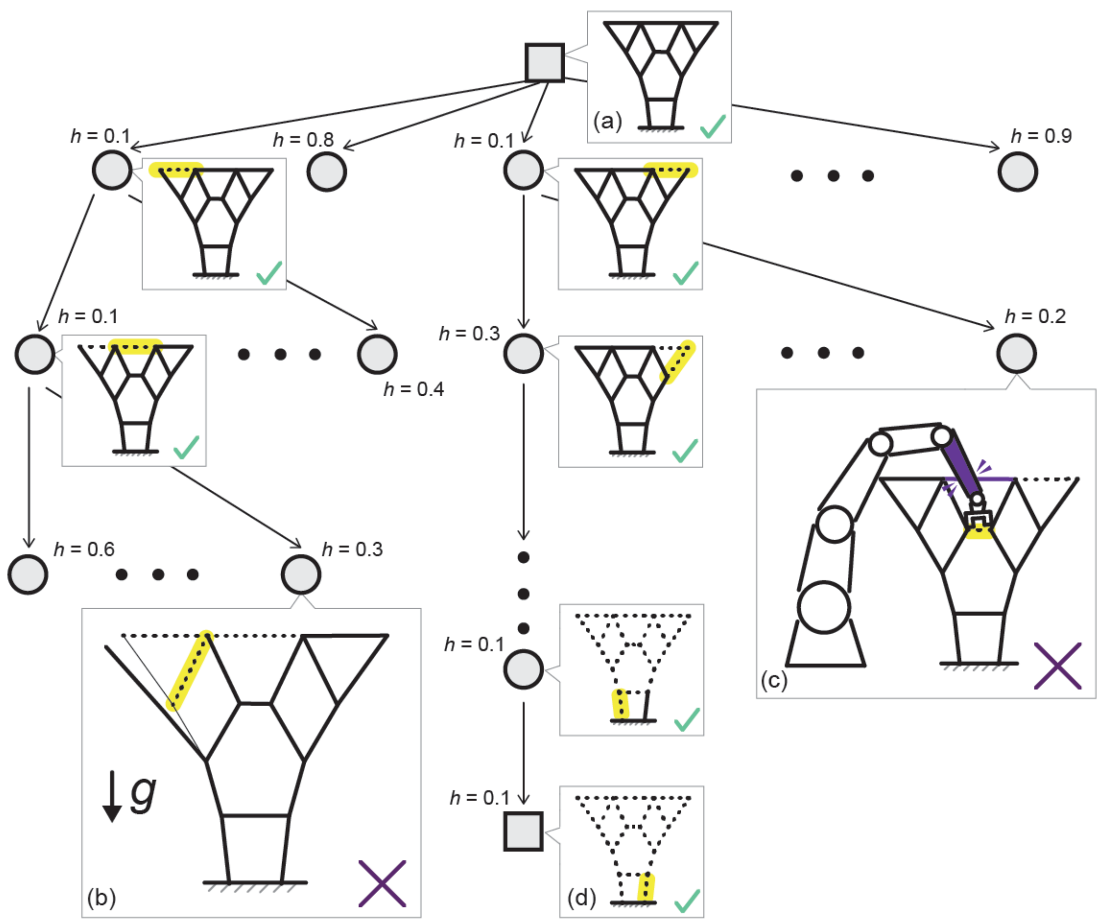


Figure 4-4: An example search tree for our backward state-space search algorithm.

We visualize the behavior of the backward search algorithm through the flow chart in Fig. 4-5. We use a priority queue to sort states according to the pair $(|P|, h(e))$, which consists of the number of remaining elements and a heuristic value $h(e)$ defined on element e. The priority queue is sorted in an ascending order by first considering element with lower $|P|$ and $h(e)$ values. Minimizing the number of remaining elements results in a *greedy* search that attempts to complete the plan as quickly as possible, akin to a depth-first search. At the start of each iteration, the element with the lowest number of remaining elements is popped, where the heuristic value $h(e)$ is a tie-breaker. This ensures that the search always prefers progressing towards removing more elements. The yellow-shaded diamond box in Fig. 4-5 signifies the structural constraint evaluation. The blue- and green-shaded diamond boxes correspond to the geometric constraint evaluations for extrusion and assembly. Both constraint evaluations are computationally expensive, because they rely on physical simulation and primitive-specific motion planning. Thus, the total efficiency of the algorithm depends on the number of states that are visited and checked, which can grow enormously if the search encounters dead ends.

Following this principle, the proposed backward search algorithm has its search direction designed to address geometric constraints, and its heuristic function tailored for the structural constraints. For the geometric constraints, removing an element increases the collision-free configuration space of the robot, since there are fewer collision bodies in the workspace. Thus, the robot is most geometrically constrained at the beginning of the backward search, limiting which elements can be initially constructed.

Although backward search makes geometric planning easier, it comes at the expense of making the structural constraints more difficult to satisfy. At the beginning of a backward search, there are many elements that can be prematurely removed without violating the structural constraints. However, as the search progresses, there are fewer options for supporting the structure, making the search more likely to reach a structural dead end. Thus, we propose a heuristic function that can bias the search to adhere to a structurally feasible construction sequence as much as possible while still satisfying the robot-related geometric constraints.

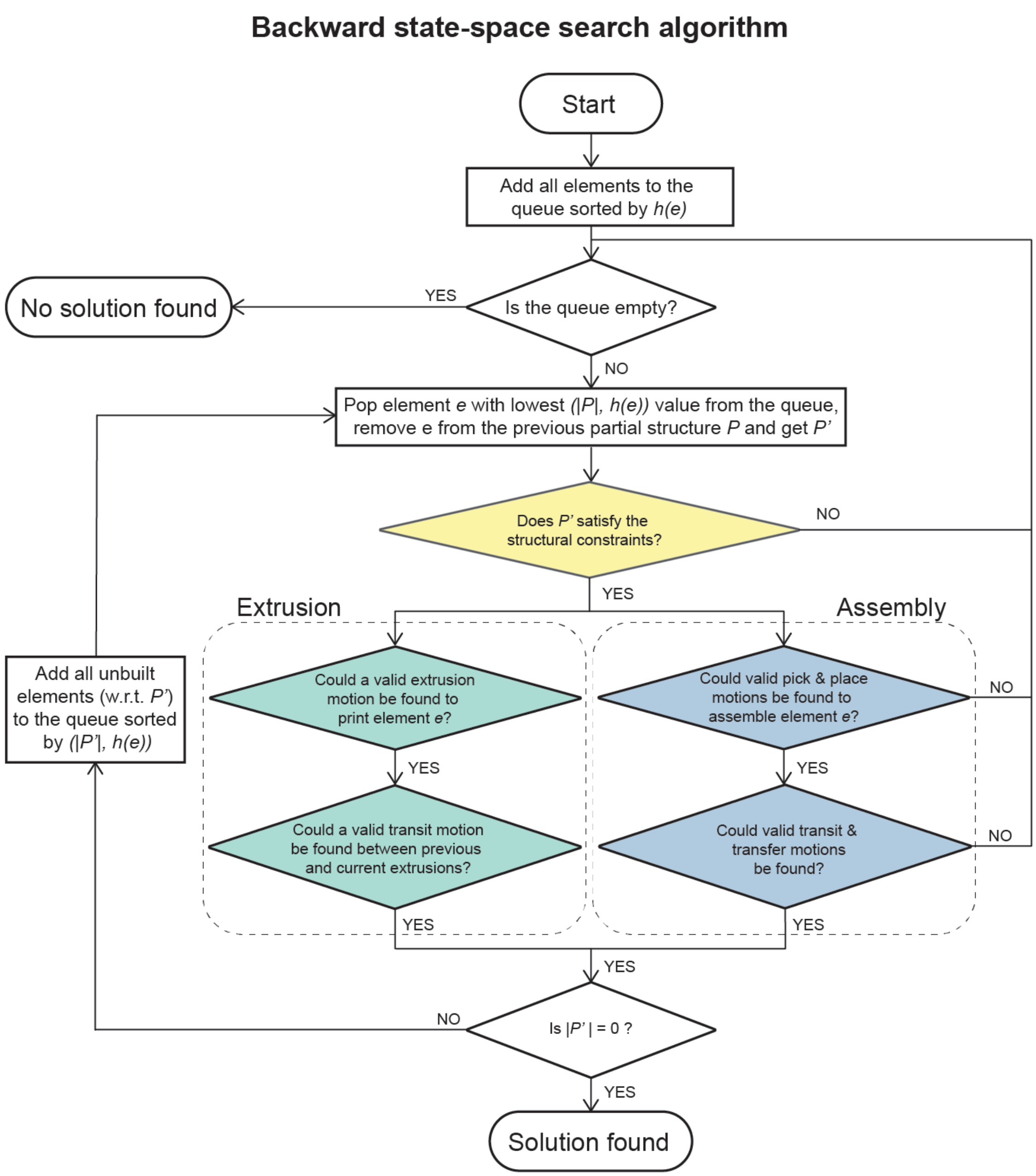


Figure 4-5: A flowchart describing our backward state-space search algorithm.

One intuitive heuristic is to remove elements based on their distance to the ground, where elements farther away from the ground are removed first. This heuristic is motivated by human builders' common construction practices. When building self-supporting structures, builders typically construct elements in a layer-by-layer fashion, in an order opposing gravity. This strategy originates from the physical reasoning that elements that are lower in the layer contribute to support the elements above them. Such heuristic functions can be seen as applying "soft" partial ordering constraints that steer the search. If the heuristic guides the search into a dead end, the algorithm will simply backtrack to previous states, thus preserving completeness. This is in contrast to the hard partial-ordering constraints used in prior work [HZH+16, WPGM16, HGM18].

Although effective in many practical situations, this distance-based heuristic might still lead to unnecessary state explorations in some special cases, such as when a structure is hung from the ceiling. To overcome some of these disadvantages, we propose a heuristic that directly takes into account the structural constraints. This heuristic performs a *forward* search for a valid construction sequence considering *only* structural constraints (*i.e.* ignoring the robot) that is then used to bias the overall *backward* search to remain as close as possible to this construction sequence. More precisely, it precomputes a valid construction sequence S using a greedy forward search and returns the negated index i of element e in the sequence. Namely, if $S[i] = e$ then its heuristic value $h(e) = -i$ (recall that an element with *lower* heurstic value will be removed first). Previous work has showed that this heuristic (*StiffPlan*) outperforms the distance-based heuristic (*Euclidean-Dist*) [GHLPM20].

4.5 Case studies

This section presents case studies of applying our planning algorithm to various extrusion and assembly problems. We present results demonstrating assembly planning for a 7-DOF robot system, with an IRB 4600 robot mounted on a 4-meter-long

ABB IRBT 4004 linear track (Section 4.5.1). A summary of these assembly case studies can be found in Table 4.1. More extensive studies on extrusion problems can be found in previous publication [GHLPM20].

Design name	# Elements	Connection type	Deformation tolerance (mm)	Dimensions (m)	Planning time (min)
Short Arch Fig. 4-10-row 1	39	double-tangent	5	1.1 x 2.5 x 1.4	2.6
Tall Arch Fig. 4-10-row 2	42	double-tangent	3	0.9 x 1.7 x 1.4	5.3
Column Fig. 4-10-row 3	39	double-tangent	5	1.0 x 2.7 x 1.3	3.3
Hydra Fig. 4-10-row 4	42	double-tangent	5	1.0 x 1.8 x 1.5	5.2
TopOpt Vault (assembly, Fig. 4-11-left)	68	ball joint	3	1.0 x 1.0 x 1.0	6.7
TopOpt Vault (extrusion, Fig. 4-11-right)	76	ball joint	1.5	0.35 x 0.35 x 0.35	4.4

Table 4.1: Overview of the assembly case studies for our additive construction method. The planning time results are for a consumer-grade laptop.

We implemented transit and transfer primitives using RRT-Connect [KJL00]. We used IKFast, an analytical inverse kinematic solver, to initialize the extrude, pick, and place primitives [Dia10] and Randomized Gradient Descent [Sti10, YG07] to plan full paths. We use PyBullet [Cou15] for collision checking, forward kinematics, and visualization during motion planning. An open-source Python implementation of our algorithm is available at `https://github.com/yijiangh/coop_assembly`.

4.5.1 Assembling double tangent bar systems

Spatial structures made of bar elements are particularly efficient solutions for spanning structures. We therefore focused on designs that showcase the advantages of spatial structures through their structural potential. Traditionally, space-frames are designed as a layer-based system with a regular grid of diagonals between horizontal elements. By utilizing robotic assembly, we can free up the design of space frames to allow for different geometric configurations that may result in free-form shapes.

Among the six case studies tested in this paper, we chose to physically demonstrate our method with the construction of an arched truss (called *Short Arch* here), consisting of 39 elements, to address a multitude of challenges that are inherent to the robotic assembly of spatial structures. An arching geometry presents a particularly challenging structural case, due to the different support conditions during and after construction. Traditionally, arches are built from both supports inwards as cantilevering structures (or using additional support) until they are connected in the middle to form a self-stable structure with two supports. Other assembly sequences are possible, but will strongly influence the structural behavior during construction and might lead to either the necessity of temporary supports or oversizing of the members to ensure structural integrity at every step of assembly. As a result, the assembly sequence plays a particularly crucial role in the stable construction of arching structures.

The Short Arch - along with the Tall Arch, Column, and Hydra designs - uses the double-tangent system proposed in Parascho et al. [PKC+18] and is based on regular tetrahedra (Fig. 4-6). This system presents a number of geometric constraints that need to be fulfilled (each bar needs to be connected to the structure in at least two points at every end) and results in complex nodes where only two elements connect at any single point. The bars' double tangent connections often necessitate a rotational insertion movement by the robot arm to avoid collisions, making it particularly difficult for the planning of the place motion primitive. As such, we chose to test the proposed construction planning algorithm on a design that combines the sequence challenges resulting from structural stability and robotic reachability.

4.5.1.1 Short Arch prototype

If a purely geometric planning process were used (*e.g.* robot only, not considering structural behavior), the structure would accordingly have been constructed from one support towards the other, following a pre-defined sequence ensuring local stability of the tetrahedra. However, the cantilevering of the entire structure before reaching the second support would result in large deformations and poten-

Figure 4-6: Short Arch; *left*: the completed structure; *right*: a detailed view of a joint.

tial connection failures. By using our planner, a custom sequence could be found which follows the construction practice of the arch-like structure from both supports. While this sequence is not optimal with regards to structural mechanics, it does provide a compromise that ensures structural stiffness and, at the same time, enables the required complex collision-free insertion trajectories to be found.

The generated construction plan for the Short Arch was tested in a physical prototype using the robotic setup described above (as shown in Fig. 4-6 and Fig. 4-7). We constructed the Short Arch structure at a one-to-one-scale using 3/4-inch Schedule 40 (26-mm diameter) PVC pipes joined by epoxy putty (material descriptions given in Table 4.2). The four bottom-most bars of the structure were affixed to wooden blocks connected to a plywood base, according to the structural modeling of these elements as rigid fixities.

Material	Details
PVC Pipe	Diameter: 26 mm Wall Thickness: 2.9mm Length: 629-945mm
Epoxy Putty	Oatey Fix-it Stick

Table 4.2: The materials used in the physical prototype of arched truss.

Epoxy putty was chosen as an adhesive due to its relatively quick setting time (about 20 minutes) and its high viscosity, allowing it to serve as a filler material

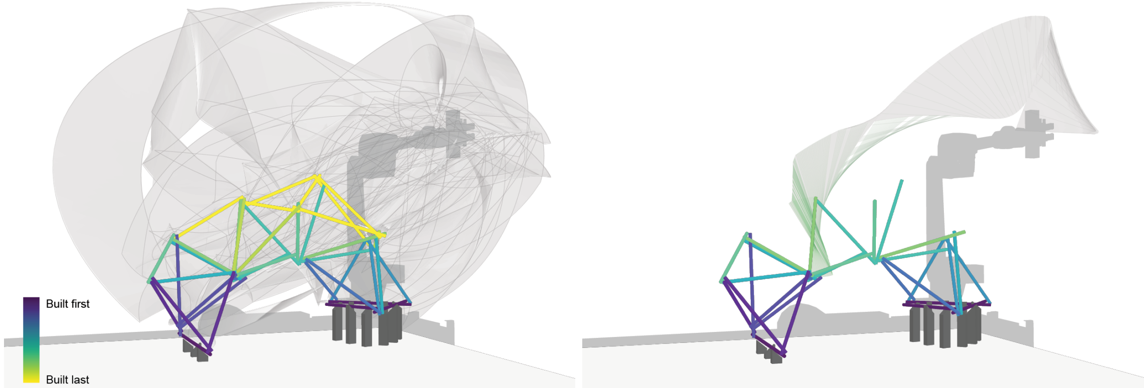


Figure 4-7: Short Arch; *left*: full sequence and trajectories displayed; *right*: a transfer and place motion for a particular bar.

for the tangent connections between elements and to bridge gaps caused by any combination of inaccuracies due to tolerances between the simulated and physical robot, as well as material deformation. The physical assembly process revealed the expected displacements during construction that are accounted for in our planning method's FEA simulation: the entire structure deflects upon release of each new element by the robot after joining to the structure. While this was predicted by our planning process, the amount of deflection was often different than predicted, likely due to a combination of inaccuracies of the robotic setup and material modeling. We observed that in many cases, this effect was self-reinforcing: larger than expected deflections caused larger than expected gaps in subsequent bars, requiring larger epoxy joints that therefore deflected more. This geometric nonlinearity of the deformation was not captured in the FEA simulations, which were based on linear elastic assumptions. Future implementations of this approach could replace the linear elastic FEA with a nonlinear solver.

The maximal deformation tolerance for checking the structural constraint was set in the planning algorithm to be 5mm, though in the majority of cases the real deformation was within the range of 30mm. A larger deformation of approximately 30mm occurred between the placement of bars 19 and 20 of the construction sequence (see frames 19-20 in Fig. 4-8). Although these were the steps predicted to have a relatively large deformation (Fig. 4-9), structural joint deformations caused a more dramatic displacement. This discrepancy between our structural simulation and the real behavior during construction can be attributed to material modeling inaccuracy, robotic imprecision, and the interrelated compounding effect described above. Additional sources of error may include the observed material creep of the epoxy putty, which led to large deformations overnight after the initial curing of the material. Finally, the spring and joint stiffnesses of the bar-putty interfaces were likely not modeled with complete accuracy.

Figure 4-8: Construction timelapse collage for the Short Arch.

Despite this unexpected behavior, the resulting misaligned joints were able to be fixed in place using additional epoxy putty, and the spatial positioning of each bar by the robot helped to prevent a catastrophic compounding of misalignments in the overall structure. Further refinement of material stiffnesses in the structural simulation and attention to robotic precision would improve the fidelity of the predicted displacements.

Construction plans for the other three double tangent bar case studies were also generated using our algorithm (although they were not built), as shown in Fig.

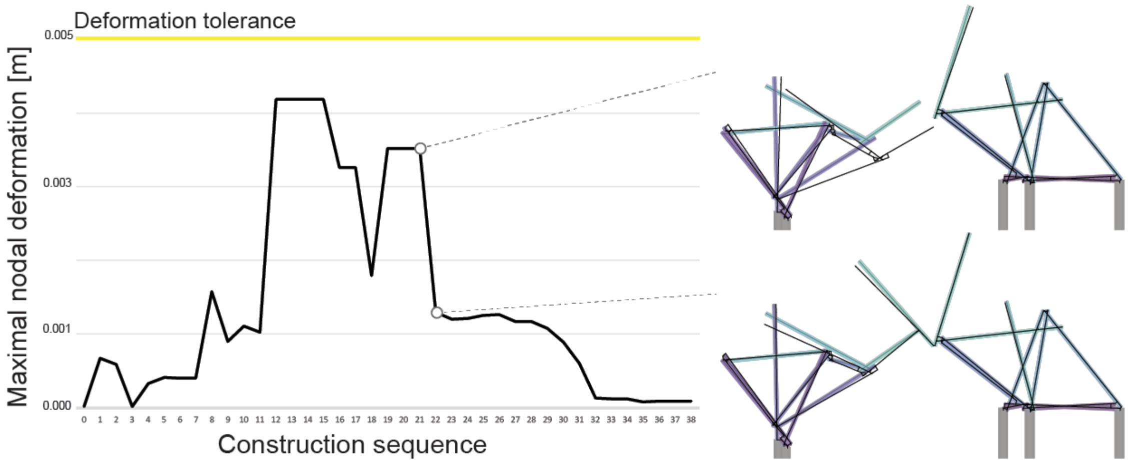


Figure 4-9: FEA-simulated deformation history of the in-progress structure over the construction sequence of the short arch. The two called out steps show the deformation reduces once a bridging element is placed between the two sides of the construction. The deformation in the called-out steps is magnified 50 times for clarity.

4-10. See supplementary materials for timelapse videos of the real construction experiment. Note that in the video, the place-approach motions were executed twice, with the first attempt as a dry-run to help identify the positions to apply the epoxy putty.

4.5.2 Topology-optimized vault

Finally, to further demonstrate the benefits of a general-purpose robotic planner for additive construction, we show planning results of extrusion and assembly of the same design, a topology optimized vault structure, constructed with different materials at different scales (Fig. 4-11), presented in a previous paper [GHLPM20] in detail for the extrusion case (see Table 4.1 for key data). It is noteworthy that the construction sequences for the two versions are quite similar (but not identical), but the motion paths are completely different. In the extrusion version, the collision-free space is slightly larger, since the robot is not holding the next bar to be placed. Additionally, the material is lighter, making it easier to meet the structural constraints. The difference in motion is also due to the difference in scales and motion primitives between extrusion and bar assembly.

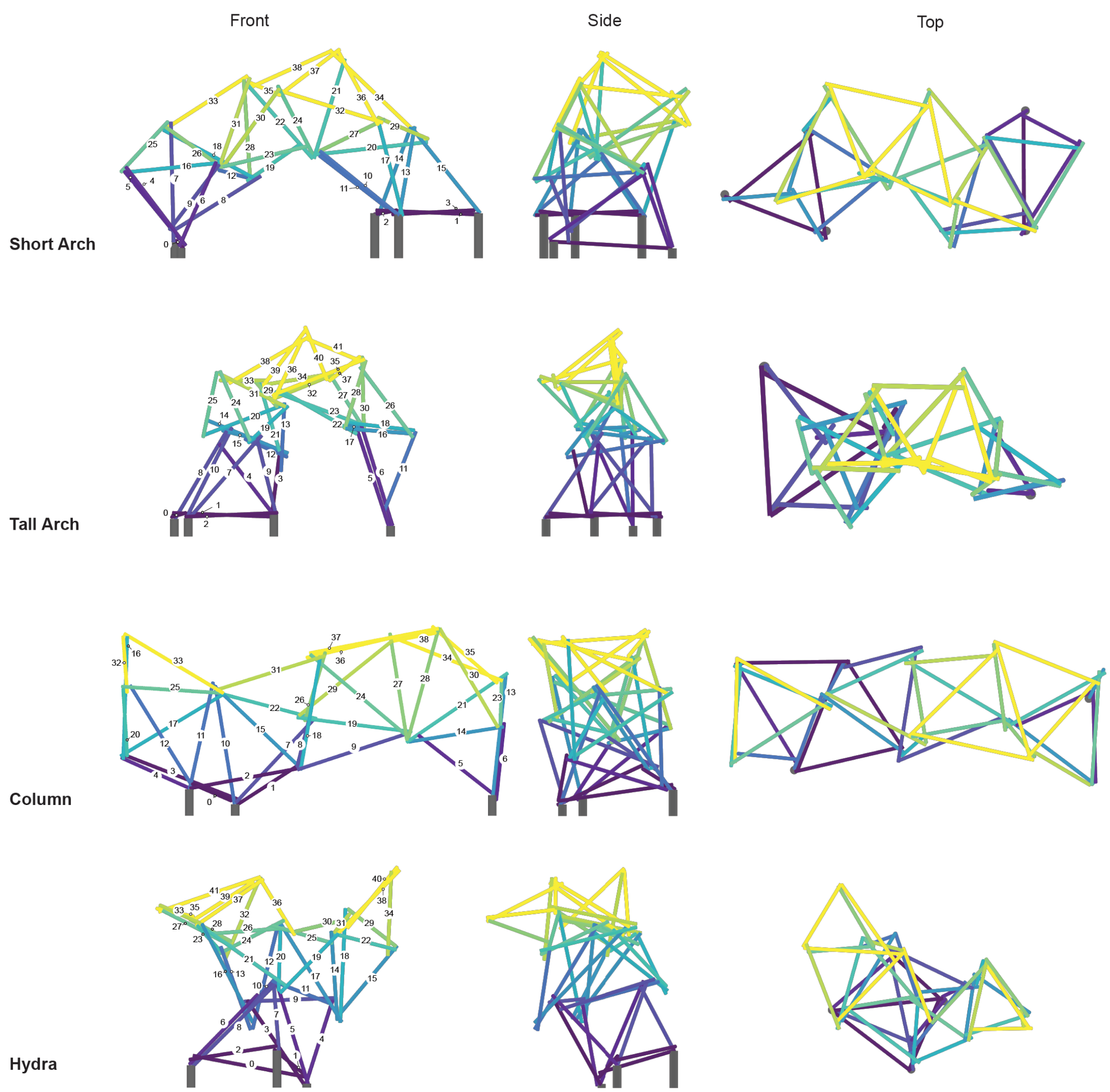


Figure 4-10: Planned construction sequences (indicated by labeled steps) for the four double tangent assembly case studies.

4.6 Conclusions, potential impact, and future work

This paper has presented a newly unified approach for planning additive construction of bar structures that are made by either spatial extrusion or discrete assembly. By extending previous research, we have leveraged a rigorous state-space search

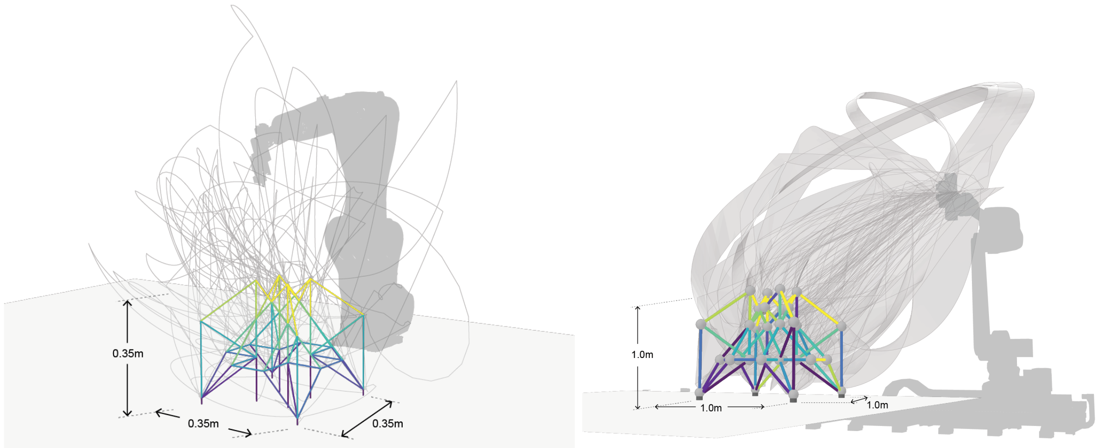


Figure 4-11: Planned trajectories for extrusion *left* and assembly *right* of the same design at different scales.

algorithm to jointly find feasible construction sequences and motion plans automatically. In order to efficiently plan in the presence of both geometric and structural constraints, the algorithm performs a backward search to find geometrically feasible trajectories and uses forward reasoning as a heuristic to guide the search through structurally-sound states. We have demonstrated the effectiveness and versatility of this approach on six diverse case studies, five of which are newly presented in this paper and involve bar assembly construction methods. A human-scale physical prototype further tests the method, with a successful result built using the sequence and motion plan generated by our algorithm.

One key advantage of our planning tool is the potential for adaptability to new construction processes, demonstrated here by the application on both extrusion and assembly, thanks to the flexible plan skeleton formulation. With the growing popularity and accessibility of digital fabrication, we are witnessing the development of new methods for construction at an ever-growing pace. In order to propose a relevant planning tool for these processes, it is crucial to consider flexibility with regard to material processes and robotic tasks. We aim to address this by generalizing our approach and making the adjustment to new fabrication methods as easy as possible. Our approach can be extended to new robotic construction systems simply by defining the plan skeleton and associated action templates. In the future, we imagine being able to provide similar sequence and motion-planning tools for applications with multiple machines, or those beyond additive processes, such as material manipulation, disassembly, or reconfiguration.

The automated nature of our approach also suggests directions for future work regarding design methods. In its current iteration, the method leads to an intuitive design setup, in which a designer does not need to consider fabrication constraints or stability during construction, but can input any design and check its feasibility through the proposed algorithm. However, in developing the work presented in this paper, we sometimes encountered structures with configurations that were difficult, if not impossible, to build. This was often due to the bars not being reachable at all by the robot, which can be a consequence of dimensions, position and orientation of the bars. While our method simplifies the construction of buildable structures (and quickly identifies whether structures are buildable by given robotic setup) we believe that the next step is to achieve a closer integration of the sequence and motion planning into the design process that currently precedes these calculations. Rather than discussing the hierarchical position of the sequence planning - either as a part of or after the design process - we imagine it fully incorporated into the design process itself. The current algorithm runs in minutes on a consumer-grade laptop, which suggests the possibility for real-time construction feasibility feedback for designers. This could be harnessed to avoid post-processing problems and geometric deadends, either in manual design methods, optimization-assisted, or generative methods. For example, we imagine that our method can be incorporated into sequence-based design processes, by providing feedback on motion planning and structural stability at every construction step. This would expand our method by adding the possibility of using the sequence to inform the geometry, as an alternative to using the geometry to define the sequence.

Chapter 5

The new analog: A protocol for linking design and construction intent with algorithmic planning for robotic assembly of complex structures

The works presented in the previous two chapters solved the sequence and motion planning (SAMP) problems that have a repetitive robotic action pattern. However, many robotic assembly processes benefit from having a more flexible, non-repetitive action pattern, e.g. processes that require the robot to alternate between transporting elements and manipulating multiple tools. As we will reveal in this chapter, such planning problems pose new formulational and solving challenges that prevent the usage of the search strategies we have developed for SAMP. This chapter aims to address these challenges by proposing two strategies: (1) a new interface concept that allows designers to formulate a robotic process in a straightforward and flexible way and (2) an automated solving technique to solve a long chain of robotic movements effectively [1].

5.1 Introduction

An emerging trend in architectural construction is to use industrial robotic arms for discrete-element assembly tasks. This approach is used for different material systems, such as timber, steel, masonry, fiber, plastic, and can be applied on different scales, ranging anywhere from interior finishes to structural elements. These projects take advantage of the precision and accuracy of robot to create *non-repetitive* (i.e. irregular) assemblies that have complex architectural expression or high structural efficiency, which are difficult to assemble manually [GMW14].

Non-repetitive robotic assembly Programming robots to build non-repetitive assemblies requires a very different paradigm from what is typically considered in industrial robotics, where the robot performs repetitive tasks in a production line and programmers can manually program and optimize a routine (often referred to as teaching). In the context of creating non-repetitive architectural assemblies, it is desirable to have general-purpose and flexible robots that can quickly adapt to evolving designs and construction concepts. In a high variation assembly, where the assembly movements do not have similar pick-and-place patterns or where the workpiece varies substantially in geometry, the time required to manually program or dry-run each individual motion can often be longer than the execution process itself. Still, it is important to validate that all the robotic motions are feasible during the design stage of an architectural scheme.

In this chapter we focus on the most difficult and crucial component of validation: ensuring robot reachability and collision-free motion. Unlike a Cartesian robot (such as a 3D printer or a laser cutter), most robotic arms have six or more degrees of freedom (DOF) and their movements cannot be easily validated without performing full robotic motion planning for all of the involved steps. In fact,

[1] A version of this chapter has been published in [HVG+21].

validation planning is so detailed such that, upon completion, the robotic execution trajectories are produced as a by-product. This nontrivial process requires formulating the assembly process as a planning scene, which defines the involved robots, tools, constraints and collision objects, and developing planning algorithms, often referred to as *planners*.

Task and motion planning Within the robotics planning literature, there are predominantly two types of planners: (1) *task planners* plan discrete decisions such as the order in which to perform various types of robot and tool motions and (2) *motion planners* plan a trajectory for a single robotic motion. Recently, researchers in architecture-scale digital fabrication have started to use planners to generate instructions for robotic assembly. However, the currently available motion planning tools for the architectural community (such as compas_fab [RCP+18], moveit! [SC18]) are only able to plan trajectories connecting two configurations. In order to apply these tools to *multi-step manipulation*, which involves several movements (e.g. picking up an object, transferring it to a different location, and inserting the object into a hole), the user needs to perform many motion planning calls, while ensuring the overlapping configurations are the same.

In the robotics literature, each movement is referred to as an *action template* [HGT+21]. An action template contains three types of parameters, which can be assigned manually or automatically: (1) discrete parameters (e.g. which object to pick or place), (2) continuous parameters (such as poses and grasps of the movable objects), and (3) continuous motion paths. Chaining multiple action templates together results in a *plan skeleton* - a high-level description of a multi-step manipulation process [GCH+21].

Repetitive vs flexible plan skeletons Robotic processes such as material extrusion or pick-and-place assembly sometimes have repetitive plan skeletons. For example, in spatial extrusion, the robot alternates between two actions: (1) extrude and (2) transit. For pick-and-place assembly, the robot repetitively performs four actions in order: (1) transit, (2) pick, (3) transfer, and (4) place. In these situations, both a construction sequence and corresponding robotic motions can be planned together automatically[HGT+21].

However, many architectural assembly processes require more flexible plan skeletons. Some examples of this include when a tool-change is needed to accommodate different workpiece geometry or if the assembly actions change depending on a variable propriety of the workpiece. In these cases, the designer traditionally must manually create case-specific planning software in order to bridge the gap between this custom behaviour and the standard motion planners.

In our view, this approach wastes human time and expertise. Process designers bring high-level intentionality and knowledge about fabrication and assembly processes that should be communicated to planning systems without the need for low-level or case-specific programming. An alternative approach could take inspiration from other complex planning problems in robotics. In general, planning for non-repetitive plan skeletons that involve both high-level actions (e.g. tool-change) as well as low-level robotic motions (e.g. linear and free-space movements) is a sub-class of task and motion planning (TAMP) from the robotics planning literature [GCH+21]. While many algorithms have been developed by the robotics community to automatically plan for both actions and motions, the formulation of such problems requires domain-specific expert knowledge that can be unfamiliar to many users. We believe this expertise gap will be closed by empowering designers and engineers to formulate their problems using a protocol that bridges between high-level intentions and low-level planning algorithms. Such a formulation can be seen as a new *digital* version of the traditional, *analog* architect-engineer-contractor conversations of relevance suited for the new era of digital fabrication.

Contribution This chapter presents the formalization of general-purpose robotic assembly planning with non-repetitive plan skeletons. Our method uses abstracted actions arranged in a flowchart to enable designers to easily describe complex, non-repetitive assembly processes. The other benefits include:

- The process description is *decoupled* from the implementation of automatic solvers and motion planners.
- The process description is fully *parametric* (e.g. geometry, joints, neighbour relationships and number of elements). Different architectural schemes can be evaluated without reformulation.
- It establishes a *protocol* between architectural design, process design and planning, allowing better separation of work and promoting collaboration between different expertise.
- The formulation is compatible with *non-sequential* motion planning, allowing difficult motions to be planned first. Process designers can easily control the planning priority based on experience and intuition, which can dramatically improve planning efficiency.
- The outputted trajectories from motion planning can be directly used for robotic *execution*.

We use a recently published spatial timber assembly process [LAT+21] as a case study to demonstrate the benefit of the proposed flexible planning framework. In particular, this assembly process has multiple grippers and fastening tools, a non-repetitive plan skeleton, and requires manipulation of long timber elements in a dense, congested environment. This allows us, a team of architects and engineers, to use the plan skeleton formulation as a protocol to effectively collaborate and to perform this case study. The method and planning results are validated by the real-world robotic construction of a spatial frame structure (4.8 x 3.0m footprint, 3.4m tall) comprised of 40 pieces of 100x100mm profile timber elements (Figure 5-1).

5.2 Challenges and related work

Early work in assembly and manipulation planning Automatic assembly of mechanical parts or structures is among the first few envisioned applications of indus-

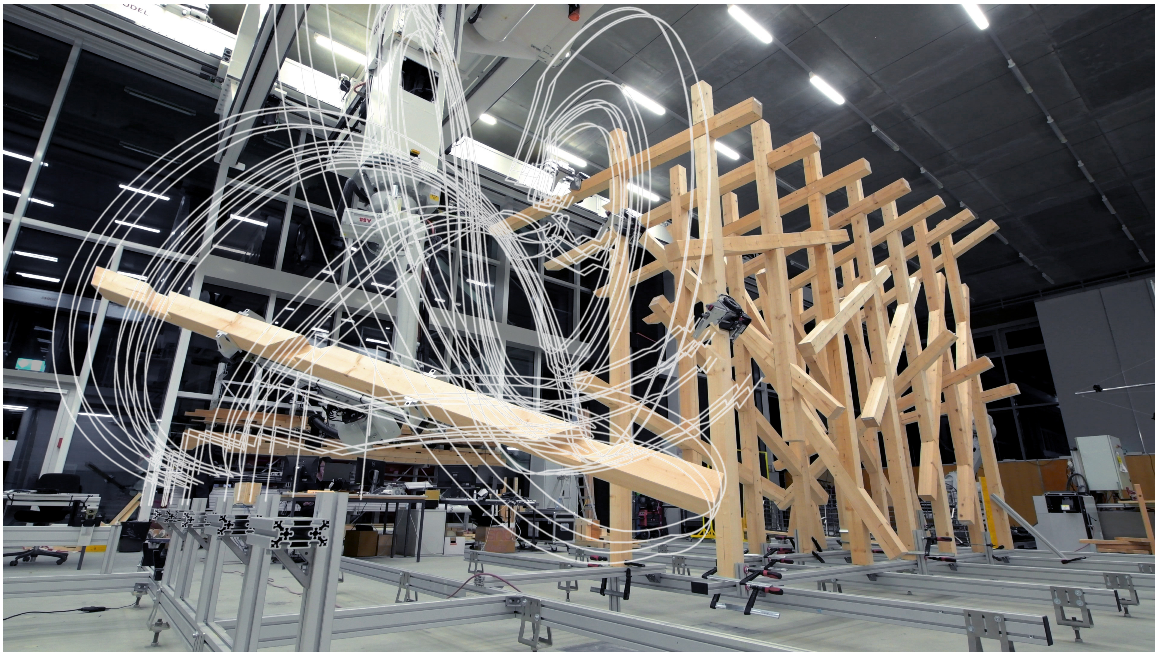


Figure 5-1: Image showing the final timber element being assembled in our case study. Three distributed clamps can be seen in the background already attached to the structure by the robotic arm. The assembly process is modeled with our flowchart and solved using our solver. Two linear and one free motion trajectory are used to bring the element from pickup to the clamps, shown here as overlaid white curves.

trial robots [AM83]. Investigations into generating assembly sequences that allow humans or robots to assemble mechanical parts based on design CAD files dates back to 1980s [DMS90, Wil92, DFW87]. This line of work focuses on low-level constraints such as mutual blocking relationship during assembly, but ignores the geometric constraints introduced by robot manipulators.

Manipulation planning problems in which the goal is not just move the robot without collision but also to operate on the objects in the world have been addressed from the earliest days of motion planning to this day, for example [LP81, ASL90, SLCS04, SK08, HNTH11, KB15, GLPK15]. However, the lack of open-source implementations of these algorithms and a proper modeling interface to connect them with practical design and construction problems makes their usage rare in architectural digital fabrication projects.

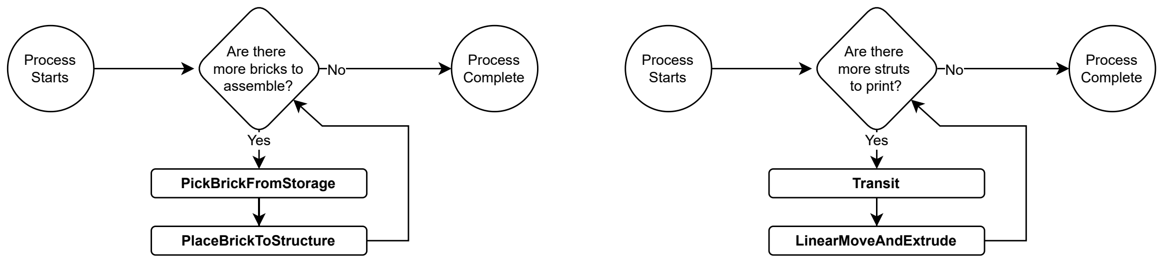


Figure 5-2: Plan skeleton for repetitive assembly processes: brick wall assembly (left) and 3D extrusion (right)

Early work in architectural robotics A number of architectural projects have used industrial robots to create bespoke spatial assemblies [EGK17, HL14, HWT+15, TAH+18]. However, many of these early works adopt a trial-and-error method for planning the actions and robotic motions, often by manually (1) assigning a fixed plan skeleton, (2) guessing a construction sequence, (3) guessing robot target configurations. Although existing software packages can support these basic operations through performing point-wise kinematics checks [BBC11, Sch12] and configuration-to-configuration motion planning [GPR+18, SC18], applying a strict ordering on the actions and solving linearly for trajectories can lead to a "stuck" situation in a dense, congested environment. For example, grasp poses or configurations that are feasible in the earlier steps might lead to infeasible situations in the subsequent actions. This leads to a highly inefficient solver that requires a lot of manual backtracking.

Sequence and motion planning In assembly problems that have a fixed plan skeleton, such as single-robot spatial extrusion and pick-and-place assembly, the robot repetitively performs certain action primitives in a fixed order (Fig. 5-2). The plan skeleton thus has a fixed length and pattern, which removes any challenge of high-level action planning. In these cases, the planning problems can be reduced to Sequence And Motion Planning (SAMP) problems, where the planner only needs to fill in the construction sequence, i.e. which element to assemble at each step, and the robotic motions. Algorithmic investigation of SAMP began in the robotic extrusion of bar structures with arbitrary geometries and topologies.

Early work along this direction addressed sequence planning for a disembodied end effector, ignoring the robot arm [HZH+16, YHL+16, WPGM16, GHB18]. One recent example is Choreo, which plans both the assembly sequence and robotic motions for extrusion processes [HGM18]. However, Choreo separates planning into separate sequence and transit phases, which can prevent it from finding solutions to feasible problems in some cases. Recent research has proposed scalable planning algorithms to solve large SAMP problems effectively without the use of human guidance [GHLPM20, HGT+21]. However, it is hard to generalize these specialized search algorithms to general assembly domains without a pre-assigned, fixed plan skeleton. This limits the problems that these algorithms can address to a small category that is overly simplified, compared to more realistic construction processes.

Task and motion planning Task and motion planning (TAMP) bridges both symbolic reasoning of actions to achieve goals and geometric reasoning in search of a collision-free robotic motions [GCH+21]. Research in this area seeks to combine discrete task planning from the artificial intelligence (AI) community [GNT04] and continuous motion planning from the robotic community [LaV06] to allow reasoning on both levels simultaneously [GLPK18b, SFR+14, Tou15]. In order to solve a broad class of TAMP problems, [GLPK20] proposed PDDLStream, a modular, domain-agnostic planning language for formulating robotic problems with symbolic task definitions. By using logical predicates to describe system states and symbolic operators to represent actions, PDDLStream and its solvers can automatically reason about the order of actions, while also planning valid robotic motions. However, PDDLStream modeling requires the users to formulate their planning problem in the format of symbolic states and actions, which is rarely used in the architectural community. In contrast, working directly with representations of high-level assembly and construction actions is more relevant to those working in architectural robotics. The missing gap is not one of solvers, but one of the challenge of problem formulation.

This chapter aims to help bridge the gap by demonstrating the TAMP-based plan skeleton formulation process with a realistic and complex case study, bringing robotic construction closer to the problem encoding used by the TAMP solvers, and thereby enabling flexible, efficient planning for a wide range of complex and realistic structures.

5.3 Formulating a construction process into a plan skeleton: a flowchart interface

In order to simplify the formulation of an assembly process for a wider user group, this chapter proposes the use of a two-step construction planning process, by first arranging high-level *actions* in a flowchart and then breaking down the actions into lower-level *movements*. This flowchart method is particularly intuitive for a designer to use and the result can be compiled automatically into plan skeletons and subsequently used to plan motions for the entire assembly process. An overview of the design to execution workflow is shown in Fig. 5-3.

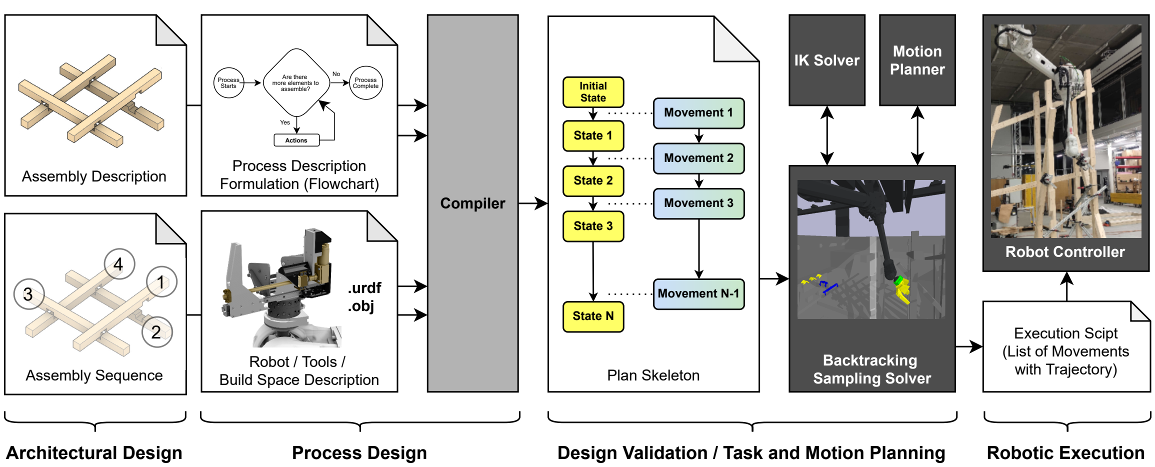


Figure 5-3: Generalized design, validation, and execution workflow for robotic assembly processes envisioned by the authors. The "Design Validation / TAMP" portion is highly automatic. Images are symbolic references to our case study.

5.3.1 High-level actions

The *actions* in our formulation are user-defined, high-level abstractions of robotic manipulations skills. Each action consists of one or more atomic *movements* that can involve a combination of robots and tools. For example, the brick stacking process in Fig. 5-2 consists of two high-level actions: PickBrickFromStorage and PlaceBrickToStructure. The two actions can be repeated to assemble as many elements as required for a given assembly sequence. During each iteration, only the parameters specific to that step need to be changed, for example, the location to pickup and place a specific brick.

In order to illustrate how actions play an important role in more complex scenarios, we will extend our discussions based on a prior spatial timber assembly process [LAT+21] (Fig. 5-1). This process utilized a group of distributed robotic clamps and grippers as well as a single 6-DOF industrial robotic arm inversely-mounted on a 3-DOF gantry. The process is designed to automatically assemble timber structures consisting of linear timber elements connected with carpentry lap joints. While in the previous publication [LAT+21] only used the robot for picking and placing the elements and had human operators to place the clamps, the case study in this chapter presents an updated version of the process, tested physically, by using the robot to automatically perform all the actions. Of particular interest for TAMP is that a variable number of robotic clamps are used during each step to apply large assembly forces while the robotic arm manipulates and supports the weight of a timber element in space.

Conditional statements Although a different number of clamps are needed for each step, the flowchart method can still be applied by adding extra conditional statements to create inner loops within each step. In our case study, an iterative loop is added (Figure 5-4.b) for the robotic arm to perform as many PickClampFromStorage and PlaceClampToStructure actions as necessary. Similarly, another conditional loop is added (Fig. 5-4.c) for the robotic arm to detach all the clamps after they have been used. On the other hand, conditional statements can also be used in the flowchart to trigger different actions based on specific properties of that step. In our case study, a conditional statement (Fig. 5-4.d) allow for automatically deciding whether to perform PlaceElementWithoutClamps or PlaceElementWithClamps depending on whether clamps are used for that element.

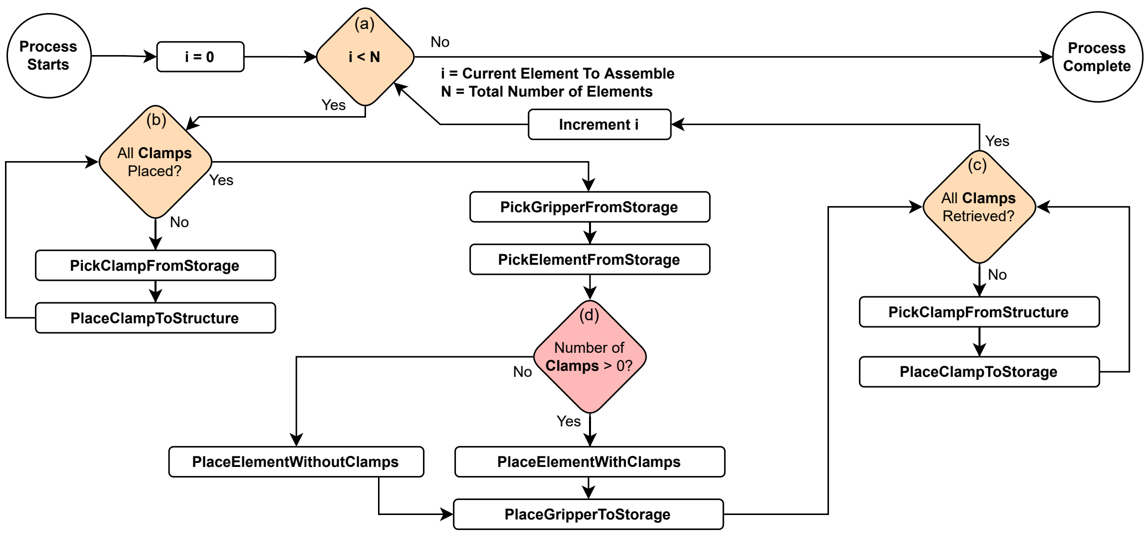


Figure 5-4: The flexible planning skeleton of the case study expressed as a flowchart, showing loops (a, b, c in orange) and conditional statements (d in red) that are evaluated for each element.

The use of a flowchart allows a process designer to focus on arranging high-level actions and defer their implementation details to a later stage. Combined with the use of conditional statements, the high-level actions reduce unnecessary specificity during process design and reduce code redundancy when later implemented. As a counterexample, it would be possible to list out all possible clamp and no-clamp scenarios in our case study as sequential scripts. However, this implementation would be very hard to maintain, reuse, and alter.

5.3.2 Low-level movements

The second step after creating the flowchart is to break down each action into a sequence of low-level movements. The movements refer to the primitive skills that a robotic system or the tools can perform. They should be formulated to be highly atomic for maximum modularity and reusability across different actions. The list below shows three common types of movements. In practice, custom movements can be formulated for a specific robotic setup (see Section 5.6).

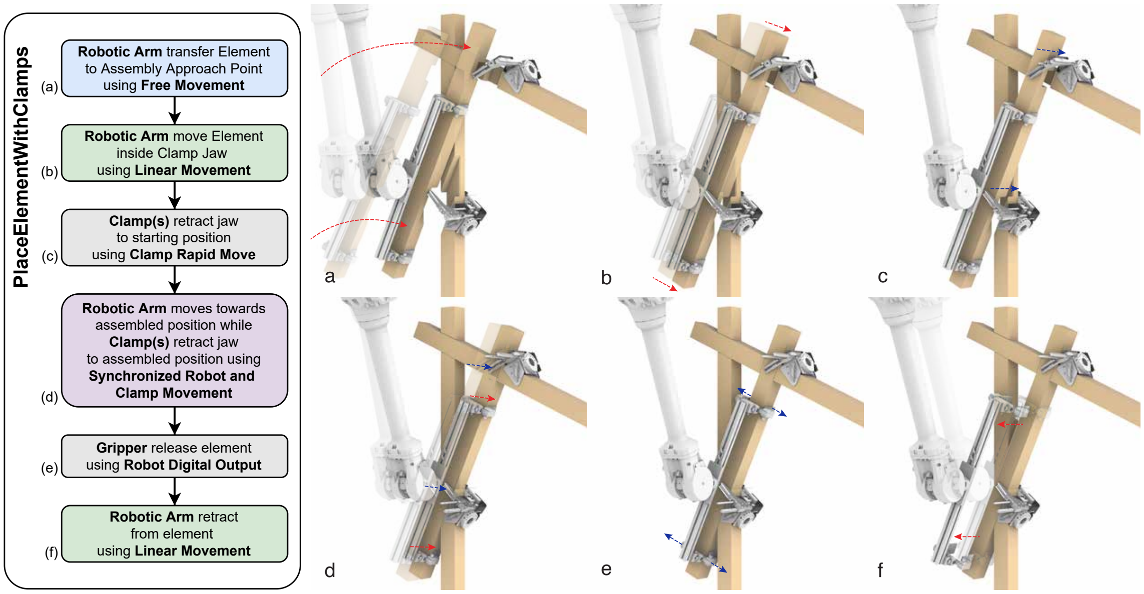


Figure 5-5: Movement decomposition of action PlaceElementWithClamps in our case study. Please refer to the legend of Fig 5-6 for the color coding of movements. Drawings on the right (a - f) shows an example of the planning scene after each Movement. Red arrows indicate movements caused by the robotic arm, blue arrows indicate tool movements. Note that step (d) is a custom movement that requires robot arm and clamps to move synchronously.

- **Robotic Movement** - actions of the robotic system that requires a motion planner for computing trajectory. During a robotic movement, tools and workpieces that are attached to the robot move together with the robot. It is possible to impose additional constraints on the robot, such as constraining the end effector to follow a linear path in 3D workspace (linear robotic movement). If no additional constraints are imposed, the movement is referred to as a free robotic movement. Section 5.4 provides more information on motion constraints.

- **Tool Movement** - discrete actions executed by tools that are either stationary or attached to the robotic system, such as opening or closing a gripper, locking or unlocking a tool changer, or turning an electric spindle on or off. These types of movements do not require computing a trajectory for the robot but may change the shape of a tool or change the attachment status of a workpiece or a tool (whether they are attached to the robot).

- **Manual Movement** - any other type of movements that are not executed automatically, such as a human manually fixing elements, making structural connections, or inspecting the structure. If these movements change the state of the objects in the scene, it is important to update the corresponding planning scene for the motion planner. Instead of executing machine code, Manual Movements simply trigger an Operator Stop and wait for human confirmation.

The decomposition of an action into movements require a level of granularity that is determined by the motion planners at hand, specifically, the motion constraint (see Section 5.3.4). For example, we cannot have a free motion and a linear motion combined as a single movement because they require two different planners to solve them. On the other hand, a free motion can be subdivided into several smaller chuck of concatenated free motions.

Fig. 5-5 shows the decomposition of one of the Actions (PlaceElementWithClamps) used in our case study. Note that a custom-formulated "Synchronized Robot & Clamp Movement" is used to model the synchronized movement unique to our case study. This movement requires a corresponding planner (in our case, similar to a linear motion planner) for motion planning.

5.3.3 Compiling a flowchart into a plan skeleton

The final step before performing motion planning is to compile the flowchart into a *plan skeleton*. A plan skeleton is a sequential list of movements (Fig. 5-6), each with an associated packet of information passed to the motion planner. The compiler is a piece of software that is created by the process designer to combine assembly description, assembly sequence, process description (flowchart, actions and movements) with a specific robotic setup. A typical compiling process involves the following tasks:

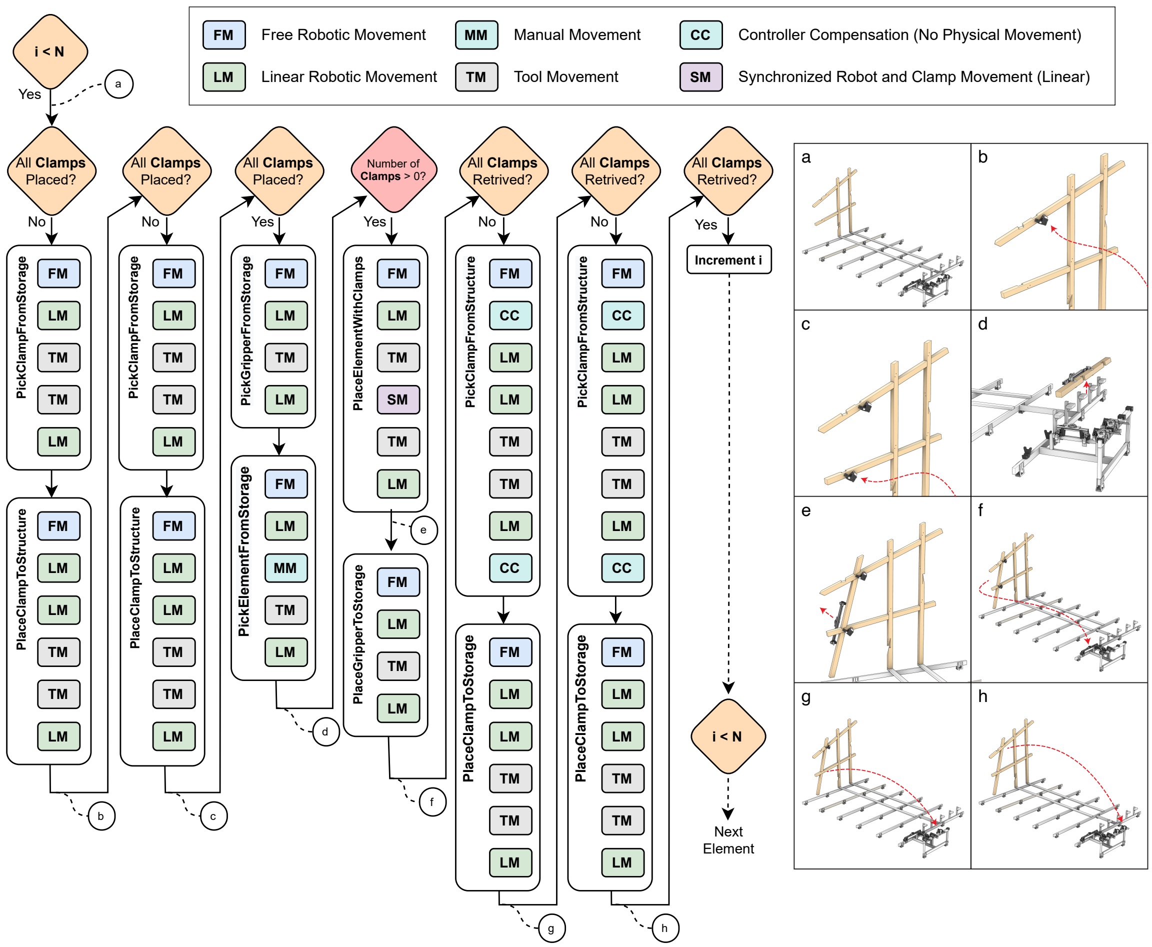


Figure 5-6: Diagrammatic output of the compiler: a full plan skeleton for one construction step (5th element in the sequence) in our case study, showing the 69 Movements created from 12 Actions. Images (a-h) on the right shows the state of the planning scene at selected moments. Red arrows indicate the general movement of the active tool before the state.

1. **Gather the assembly sequence**: Depending on the type of assembly, the sequence can be manually specified (our case study) or automatically computed from the assembly description using heuristics (e.g. a brick wall).

2. **Gather the sequence for other loops within each construction step**: For example, in our case study, we have two additional inner loops within each

construction step regarding the sequence of attaching and detaching clamps. These are automatically assigned based on available tools.

3. **Evaluate the conditional statements**: All the conditional statements in the flow chart can be evaluated from the assembly sequence and properties in the assembly description. The flowchart will therefore turn into a linear sequential list of Actions.

4. **Gather action-specific parameters**: Actions may contain parameters that are computed from the assembly description, such as the target frame for the robot or tool to reach or the grasp pose. These parameters can either be constant (for example, always holding a brick on its center from its top face) or variable (for example, in our case study, holding a timber element along various locations and directions along its longitude axis).

5. **Decompose actions into movements and gather movement-specific parameters** - Movements may contain parameters that are copied from its parent action (e.g. target frame, tool id) or computed directly from the assembly description (e.g. allowed collision pairs between the workpiece and its to-be-connected neighbors, see Section 5.3.4).

5.3.4 Planning constraints for motion planners

The primary purpose of compiling a high-level flowchart into a plan skeleton is to convert a multi-stepped planning problem into atomic motion planning tasks that an off-the-shelf motion planner (MP) can individually solve. In general, motion planners search for a robotic trajectory within a feasible configuration space (described in joint positions) defined by constraints.

- **Joint Limits** - In general, an MP will stay within the joint limits of the robotic system. This limit is typically non-changing.

- **Robot Collision** - The MP will prevent the robotic system (including any attached tool and grasped workpieces) from colliding with itself or the stationary environment. Extra *allowable collision pairs* can also be specified for the MP to ignore expected contacts. Note that the environment and the expected contacts may change from step to step due to the progression of the assembly task.

- **Motion Constraint** - Constraints can be specified by the process designer to achieve a more controlled motion. For example, a linear movement constraint requires the tool tip to stay on a linear path in the workspace. In our case study, only free and linear movements are used. However, other constraints such as twisting and rotational motion constraints exist [BSK11].

- **Targets** - The MP requires start and end targets to be specified as input. These targets can be defined as a loosely constrained tool pose (also called a tool frame) or a prescribed robot configuration (joint position). Typically, a tool pose is specified and the MP can freely sample a valid robot configuration by internally calling an inverse kinematics (IK) sampler. A fixed, rigid robot configuration can be used in cases where a specific configuration is preferred by the designer, such as tool change positions and workpiece pickup positions [2] .

Fixed constraint within one movement By convention of most existing MPs, these planning constraints are fixed within one planning call. Our movement formulation thus also follows this convention and will not allow changing constraints within one movement. However, many assembly processes (including our case study) require changing the allowable collision pairs, such as during a tool change or when grasping a workpiece. In order to overcome the limitation, it is possible to add a linear movement shortly before making contact (e.g Fig 5-5.b) and shortly after breaking contact (e.g Fig 5-5.f) to specify different allowable collision pairs.

[2] The exact positions of these targets often depends on the physical setup and are often acquired by manually jogging the robotic system to alignment and then reading out the robotic configuration to achieve maximum repeatability. This technique is used throughout our case study for all the tool storage positions and the element pickup positions.

Moreover, the linear motion constraint can be beneficial for creating a predictable trajectory for such operations.

5.4 Plan skeleton solver

A single motion planning task can be solved using our movement formulation by passing the robot description, tools descriptions, object geometries (meshes), and planning constraints to a corresponding MP (based on the required motion constraint). However, a more involved solver is needed to solve all the sequentially coupled movements in a plan skeleton. Specifically, the "intersection" between two adjacent movements trajectory must share a common robot configuration to ensure joint-space continuity during execution. Therefore, a solver needs to pass the ending robot configuration of the previous motion as the starting target for the next MP task.

5.4.1 Linear sequential solver

A simple method of solving all the movements in a plan skeleton is to solve individual movements in a fixed linear sequence. Starting from the first movement, a trajectory is planned by calling the corresponding MP. The robot configuration at the end of this trajectory is propagated as the starting target constraint for planning the next movement (Fig. 5-7). This is repeated for all the robotic movements (skipping over non-robotic movements) until all the movements are planned.

Due to the stochastic nature of MPs and the underlying IK samplers, a planning request may fail to find a solution within a reasonable time (i.e. timeout). In this case, the cause of the failure may lie in the planning result of previous movements. Specifically, the randomly sampled ending robotic configuration of the previous movement can create an unfavorable starting constraint for the current movement. Therefore it is necessary for the solver to adopt a backtracking or restarting mechanism for rewinding the planned movements and allow for another random attempt.

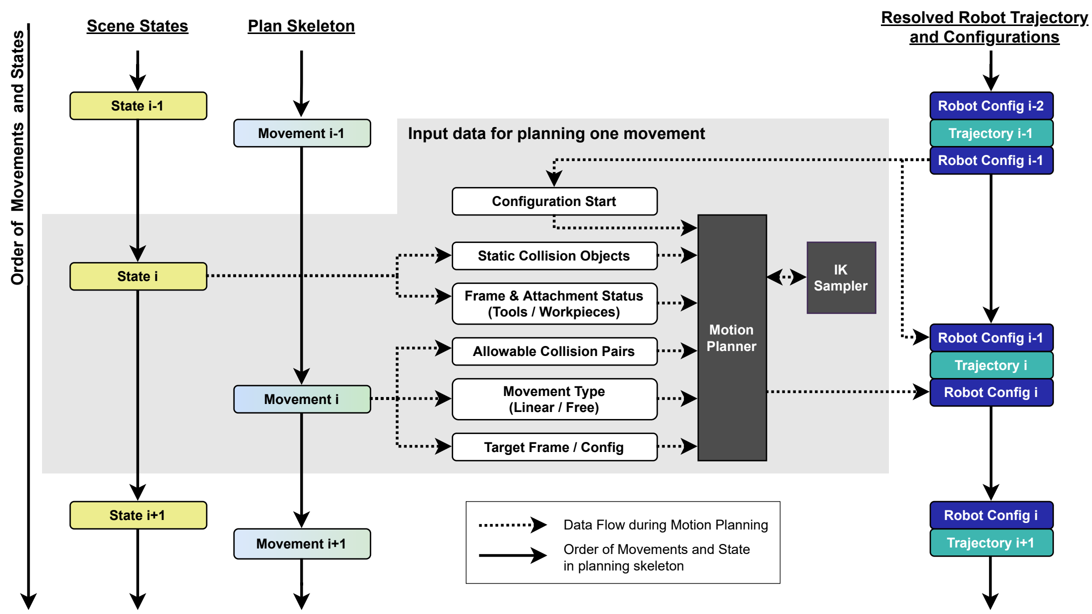


Figure 5-7: Diagram showing data flow for planning one movement within a plan skeleton. Note that the robotic configuration of the planning result is used for planning the next movement, this constraint ensures motion continuity.

5.4.2 Nonlinear solver

Unfortunately, the seemingly intuitive sequential planning strategy is highly inefficient at solving plan skeletons with scattered fixed robot configurations (e.g. for tool change and workpiece pickup) and with scattered difficult movements (e.g. narrow passages when manipulating long timber elements in congested environment). The backtracking or restarting method in a sequential solver will result in a lot of wasted effort solving the easier portion of the plan, only to be backtracked by a difficult movement.

One example of narrow passage manipulation can be seen in Fig 5-8. It shows two possible robot configurations after the robotic arm brings a clamp to the joint using free movement. However, one of them cannot proceed further due to an imminent collision. The second described difficulty, related to a fixed robotic configuration, can be seen in the tool changes in our case study. There, a free movement is used to bring the robot close to the tool storage and then a short linear movement is used to bring the robot to a predetermined fixed configuration. When solved linearly, the probability of the free movement to sample a configuration close enough to the fixed configuration is too low for the linear MP to bridge. In our case study, the success rate of solving the movements of a single element using a linear solver is practically zero (see results in Section 5.5).

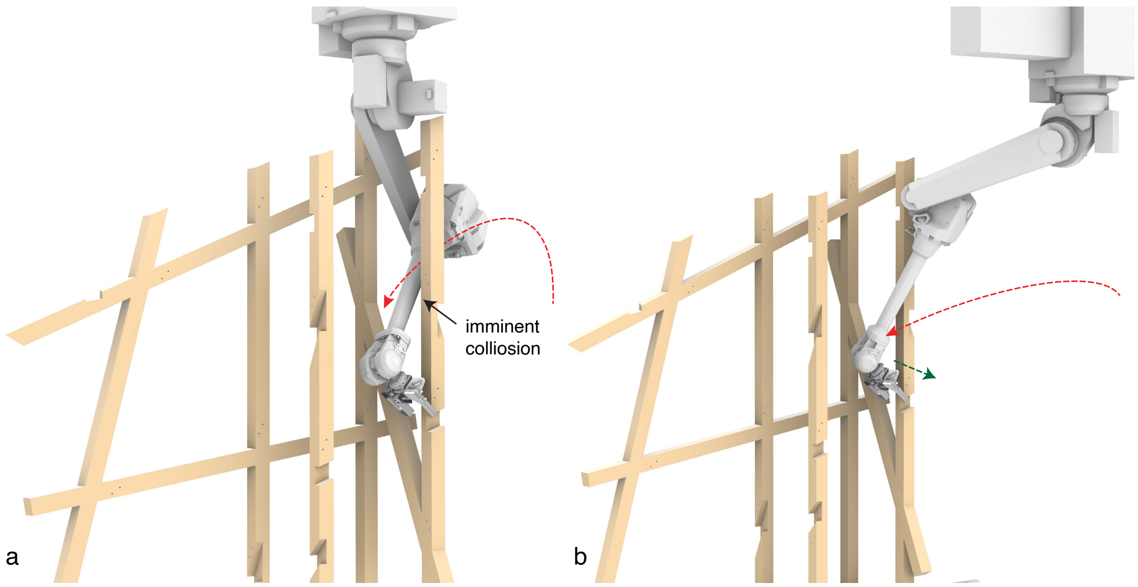


Figure 5-8: Comparison between a good and bad state after the robotic arm transfers a clamp to the structure (a free movement), before the final approaching movement (a linear movement). Due to the stochastic nature of the IK sampler, any good or bad scenarios are equally likely to happen. (*a*): The robotic arm body is sandwiched between the structure after its previous movement (red arrow). The linear MP for the next movement cannot find a collision free path. (*b*): The robotic arm holds the clamp in a different configuration that is possible for next movement (green arrow).

In order to overcome this problem, we introduce a non-linear solving method (Fig. 5-9.a) based on a priority heuristics that describe the "difficulty" of a movement. In general, movements with more constraints are more difficult to plan, since more constraints leads to smaller feasible robot configuration space [KMK18]. Similarly, movements with more, larger attached objects or collision objects have smaller collision-free robot configuration space, thus harder to plan. These heuristics allows the solver to plan difficult movements first, thereby failing quickly before spending time on the other movements. The starting robotic state constraint propagation works essentially the same way as the sequential planner, but it propagates both forward and backwards to neighboring movements. This intuition essentially allows the backtracking algorithm to eliminate unsuitable configuration candidates earlier and can substantially improve success rates.

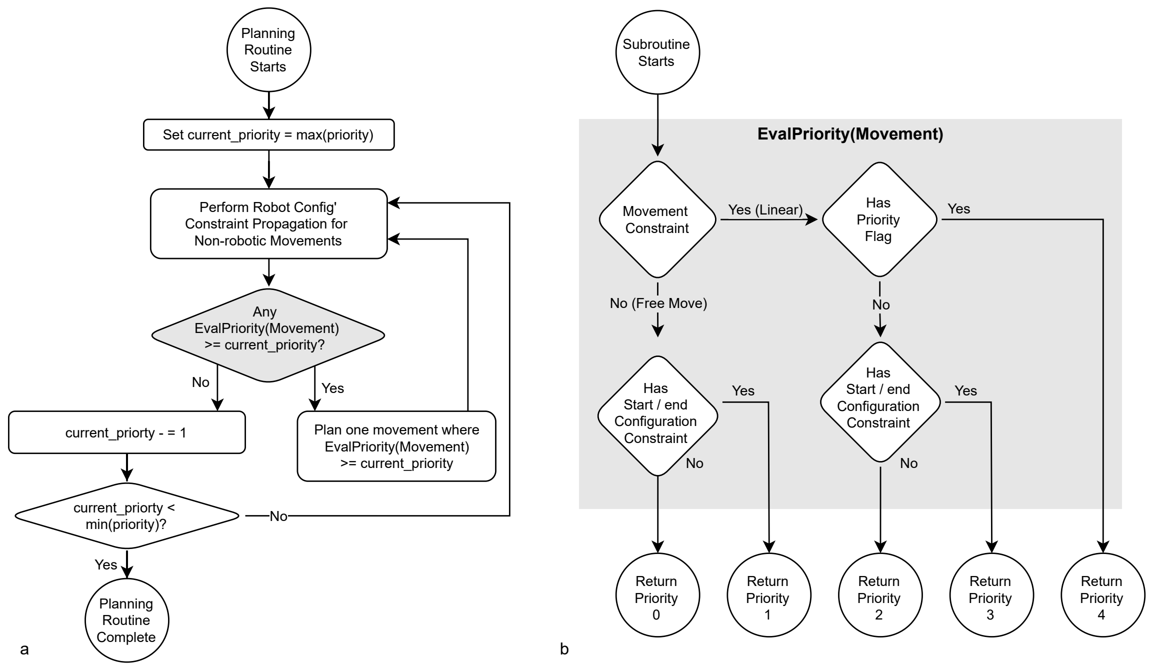


Figure 5-9: a: Nonlinear algorithm for solving plan skeleton based on a priority heuristics. b: Movement priority heuristics used in our case study, higher scores are considered more difficult and are planned earlier.

Case-study specific priority computation The priority heuristics used in our case study can be seen in Fig.5-9b. We first use the general rule of thumb to classify movements that have more constraints (motion constraints and start configuration constraints) to be more difficult. In addition, the process designer can assign a priority flag (a Boolean value) to specific movements of an action to denote higher priority. The inclusion of designer intuition into our heuristics allows a process designer to help the solver making informed decisions. We also found that it is natural for a process designer to speculate which movements are the most constrained and thus most difficult to plan a motion for.

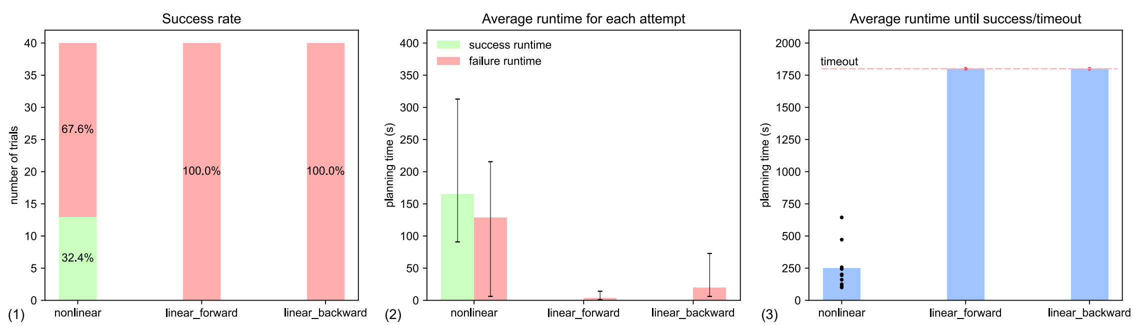
5.5 Software implementation and runtime results

In order to validate our method and to generate trajectories for executing the case study, we implement our algorithms and solvers using Pybullet [CB16] as a simulation platform, which takes care of collision checking, forward kinematics, and visualization during motion planning. Robot and tool description are based on Unified Robot Description Format (URDF) and meshes (.obj and .stl). Assembly description, robotic configuration and geometry classes are extended from the compas framework [Vmo21]. Flowchart, action-movement decomposition and compiler are implemented by the authors in Python, without the graphical visualization or user interface described in Fig. 5-4 or Fig. 5-6. Timber element geometry and robotic tool geometry are parsed from objects modeled in Rhinoceros 6 [Rob22]. Assembly sequence and assembly directions for individual element are specified manually using an interactive script within the Rhinoceros 6 canvas.

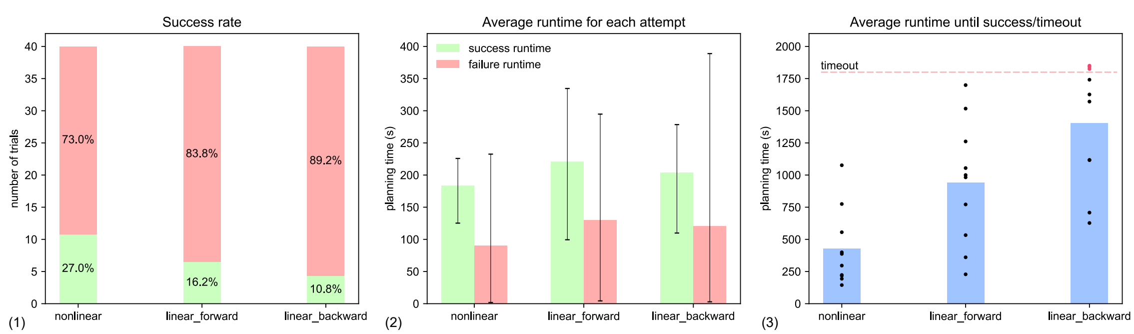
Our free motion planner implements the RRT-connect algorithm [KJL00]. Our linear motion planner implements Randomized Gradient Descent [YG07, Sti10], with Trac-IK as its gradient-based IK solver [BA15]. The three extra DOF of our robots is handled by first sampling the gantry position within a sphere near the target end-effector pose, and then using IKFast [DK08] (an analytical IK solver) to obtain all the configurations for the 6-DOF robot arm. Both MPs comply with the standard interfaces described in the compas_fab [RCP+18] API, the robotic fabrication package extending the compas framework.

To compare the effect of linear and nonlinear solving, we perform 40 random trials on solving a sequence of movements for a particular timber element (5^{th} in the sequence, the same construction step as shown in Fig. 5-6). For the linear sequential solver, we include forward (plan from the first movement to the end) and backward order (plan from the last movement to the start). The results shows that when fixed robot configuration constraints are included (for tool-change), the linear sequential solvers cannot solve the problem at all (Fig. 5-10.a.1). In contrast, the nonlinear planner can solve the problem with a 32.4% success rate. Fig. 5-10.a.2 shows the average time of successful and failed trials. In addition to the rollout experiment above, we perform 10 random trials of running the planner until success or timeout (1800 seconds) with automatic random restarts. Fig. 5-10.a.3 shows the planner's average solving time. The dots represents the individual data from the random trials.

To further compare the solvers, we remove the robot configuration constraints and perform the same experiments as above. The result (Fig. 5-10.b.1) shows that the nonlinear planner still have higher success rate (27.0%) than the linear sequential solvers (16.2%, 10.8%) . The nonlinear solver also has a lower averaged planning time until a solution is found (Fig. 5-10.b.3). Finally, comparing Fig. 5-10.a.3 and Fig. 5-10.b.3, it is interesting to see that the nonlinear planner solves the problem faster when robot configuration constraints are included. The fixed configurations provide useful hints for the solver because these configurations are conjured manually to be in a 'non-stretched' position and tested to be collision free.



(a) Fixed robot configuration constraints (used in tool changing) are included.



(b) Fixed robot configuration constraints (used in tool changing) are ignored.

Figure 5-10: Comparison between linear and nonlinear solver for solving 45 robotic movement for the 5th timber element.

Fig. 5-11 shows the planning time for all the elements along the construction sequence using the nonlinear solver. The variation in planning time roughly corresponds to the difficulty of planning in different construction steps. A timelapse video of the real-world construction experiments can be found in the supplementary material.

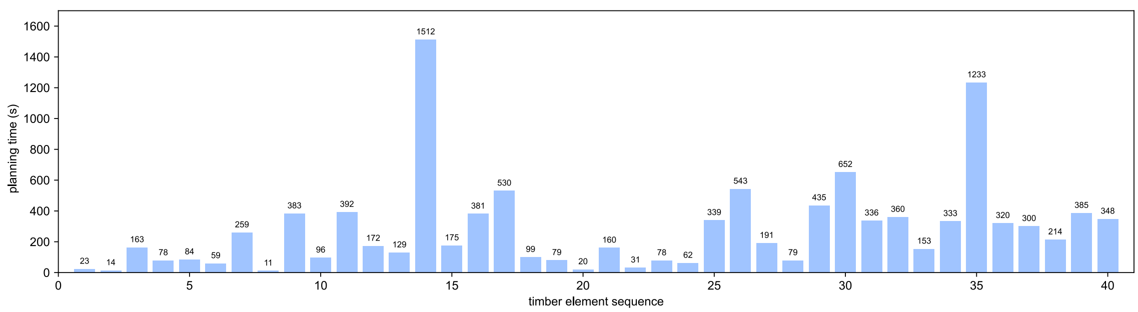


Figure 5-11: Computing time for nonlinear solver to solve movements for each of the 40 timber elements used in our case study. Element index indicates its place in the construction sequence.

5.6 Extending our formulation for practical issues in construction

The following sections aim to show the compatibility of our process formulation method in coping with various practical situations that arise in the presented case study. Beyond the case study, our formulation method can be further applied to other architectural assembly scenarios. More examples can be found in Section 5.7.

5.6.1 Inclusion of temporary scaffolding

Certain assembly processes require the addition of scaffolding in the middle of the process to temporarily support the structure. Using our formulation, both robotic and manual manipulation of scaffolding can be easily incorporated into the Action Flowchart (Fig. 5-12). It can also be used to remove workpieces. This allows all of the collision objects in the scene to be correctly modeled and avoids unexpected collisions during execution.

During the construction of our case study, we used manual carpentry clamps and aluminum profiles for temporary structural support. We initially assumed that their geometries were small enough to not cause collisions and therefore did not model them. However, collisions happened multiple times, causing substantial disruption to the process and proving that temporary scaffold modeling is necessary.

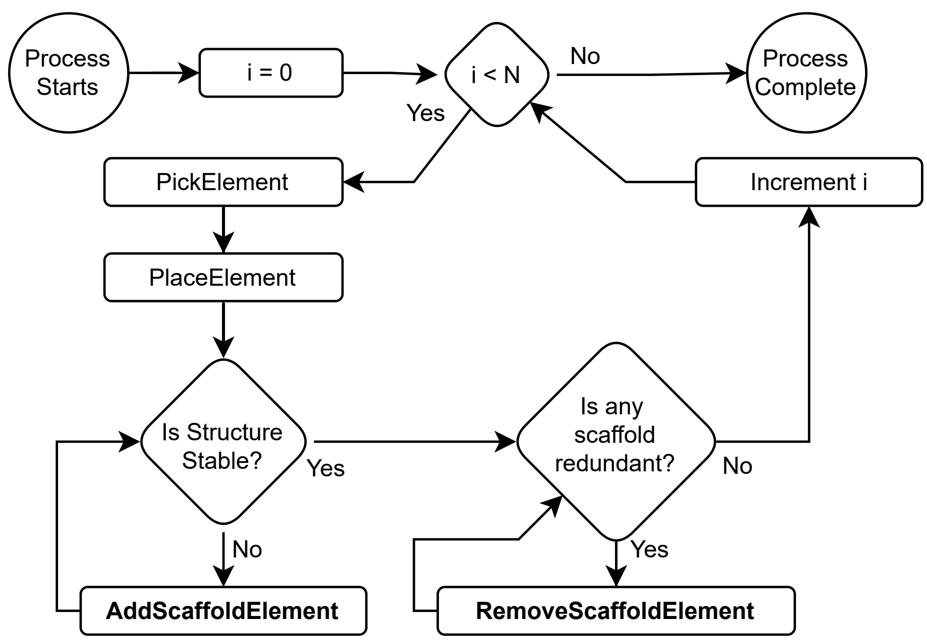


Figure 5-12: Addition of AddScaffoldElement and RemoveScaffoldElement (highlighted in bold) to a generic pick and place flowchart. This is compatible with our case study flowchart in Fig. 5-4.

5.6.2 Executing paths with controlled collision

Many different industrial robotic arms now support compliant movement modes that assist alignment or provide contact forces. Depending on the manufacturer, performing such commands may require parameter values such as SoftDirections and PayloadInformation for each motion segment. Using our flowchart formulation method, we programmed these parameters as optional parameters in the corresponding robotic movements for execution with our ABB controller. The one-to-one relationship between the low-level movement formulation and the execution code allows parameters to be passed seamlessly from the designer to the machine during execution.

5.6.3 Online visual alignment

One of the actions in our case study requires the robotic arm to dock with and detach a clamp from the structure. This is a challenging movement as the clamp may have moved during the clamping process and deviate from the programmed position. By adding a camera at the robot flange next to the tool changer, we are able to detect the misalignment based on a captured image. Because the correction amount is small, we implemented it as a Cartesian offset instead of re-planning the motion. In our PickClampFromStructure action formulation (Fig. 5-13), we have added two special movements: AcquireDockingOffset to direct the camera to acquire the correction image, and apply the offset for subsequent movements and CancelDockingOffset to remove it after the dock and detach procedure is completed.

5.7 Other formulation examples

In this section, we provide two examples of using our flowchart interface to formulate non-repetitive robotic assembly processes published by other researchers. Figure 5-14 describes a multi-robot assembly process where some robots are used as temporary support. It is a generalized description from the two-robot steel tube assembly process by [Par19]. This is different from the scaffolding approach described in Section 5.6.1, because the robots can take turns to support and transfer elements. Figure 5-15 describes a robotic fiber winding process by [DEKW+20] where a robotic arm interfaces with a Cartesian machine.

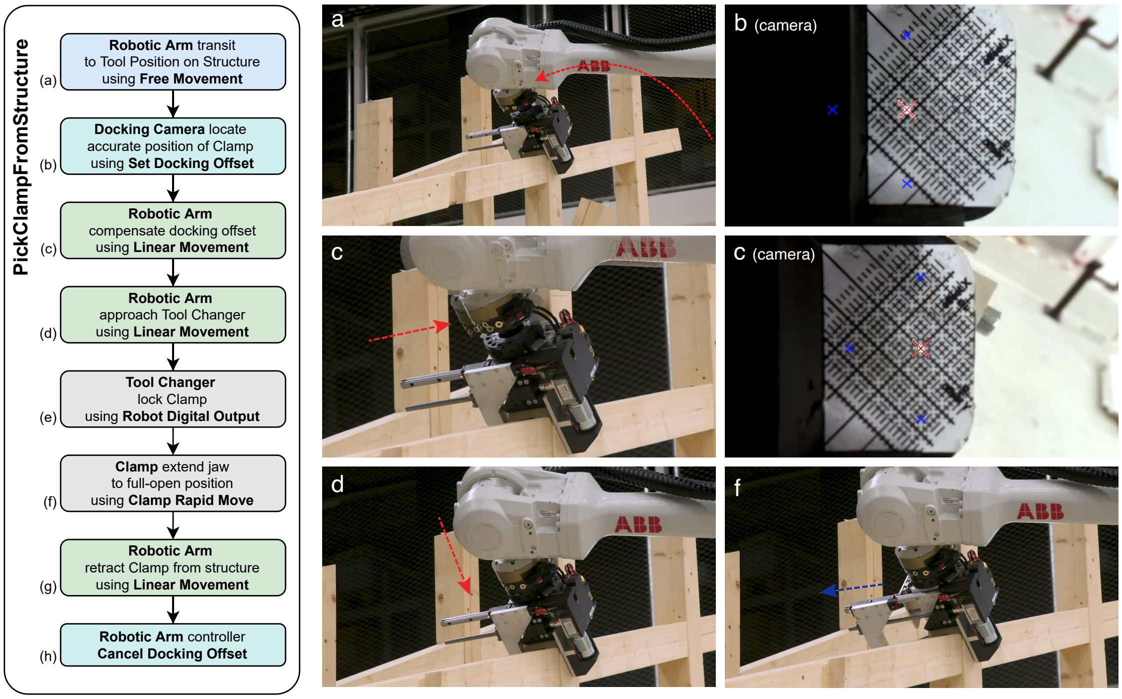


Figure 5-13: Visual alignment and docking procedures implemented by two special movements: AcquireDockingOffset and CancelDockingOffset. Left: movement decomposition of action PickClampFromStructure in our case study. Photos on the right shows the state after corresponding movements (a,b,c,d,f) are executed. Arrows indicate the movement of the robotic arm (red in a,c,d) and the clamp jaw (blue in f)

5.8 Conclusions and future work

This chapter contributes a new protocol for architectural and construction process designers to communicate intent and knowledge to robotic task and motion planners. This protocol allows for complex, realistic construction robotics applications with non-repetitive assemblies to make use of efficient, state-of-the-art planning methods that offload low-level efforts from human designers. The method is demonstrated computationally and physically on a real-world case study. Key contributions and directions for future work are given briefly below:

Interface for designers The computational fabrication field does not yet have a good interface to perform task and motion planning for general-purpose robotic assembly. This chapter has presented a new interface concept that will allow de-

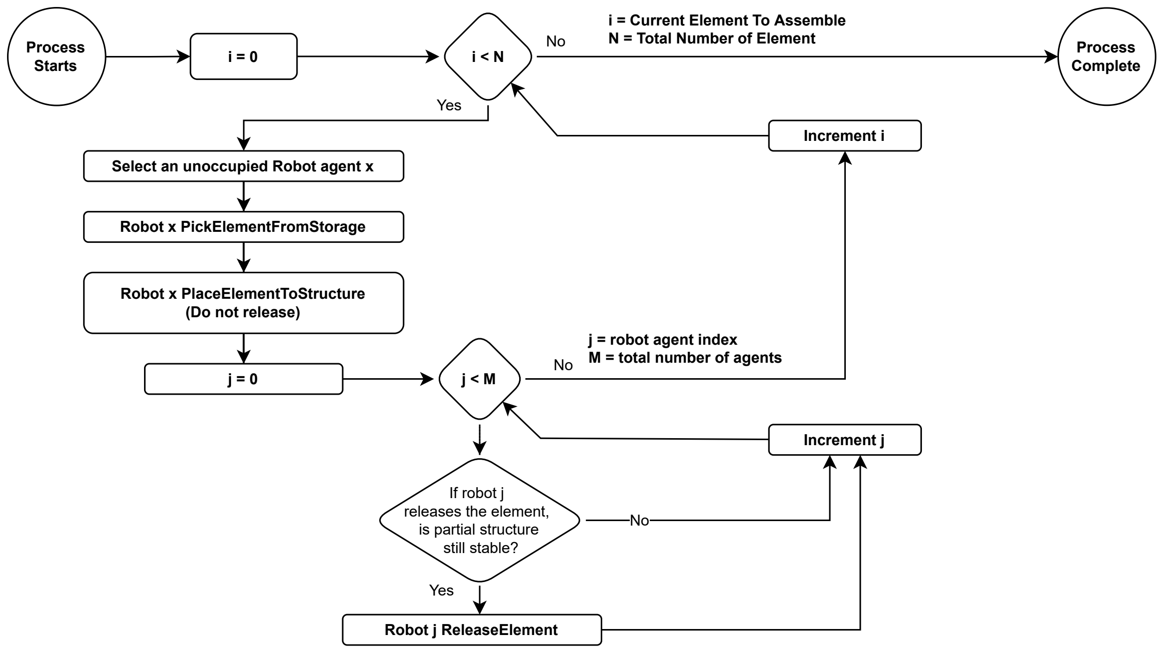


Figure 5-14: Construction process flowchart for multi-robot assembly processes. This is a generalized version of the process presented in [Par19].

signers to formulate and solve planning problems even with non-repetitive action patterns easily. The proposed flowchart-based visualization allows designers to formulate their problems visually using computational logic that understandable both by humans and computers. At the moment, our work did not implement a graphical user interface for manipulating the flowchart. Therefore, designers have to be well versed with Python code to adjust the flowchart logic.

However, we speculate an interactive icon-dragging interface that can be modeled after the visual programming platform Grasshopper [Rob22]. The major input form the architectural designer will be: (1) the geometry of the parts to be assembled (2) the position of the parts after they are assembled. (3) assembly sequence and assembly direction (can potentially be automatic). The process designer should be able to input (1) digital representations of available robots and tools, including their geometry, kinematics, actions and capability. (2) Assembly flowchart. In our demonstration, the architectural designer and the process designer role is filled by two person. However, future work can study whether this separation is useful or necessary. At the moment, the compiler is created specifi-

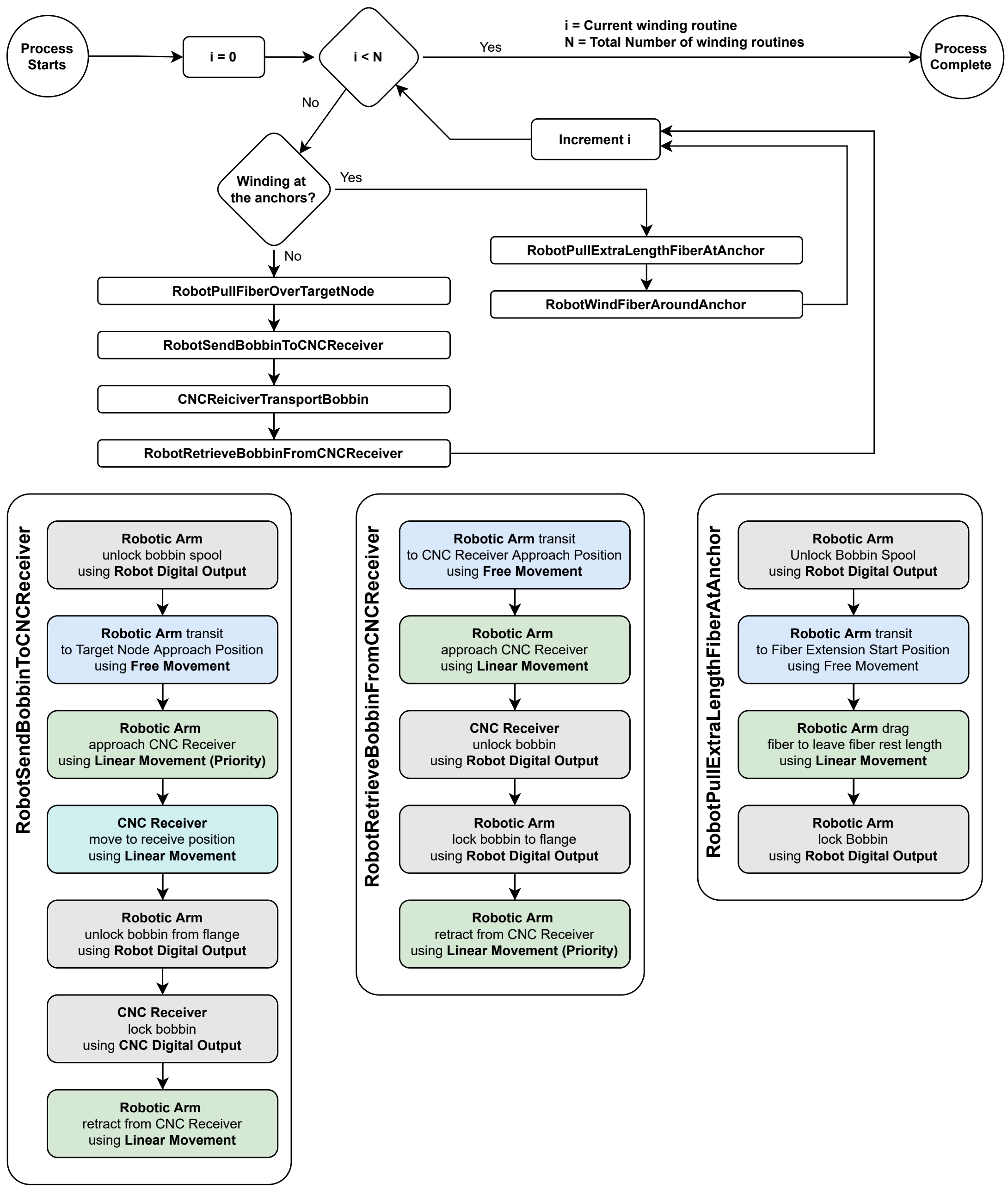


Figure 5-15: Construction process flowchart for spatial fiber winding based on process presented in [DEKW+20].

cally for our choice of tools (clamps and gripper) and parts (wood elements with half-lap joints. Future work should study how this can be generalized such that changing tools or part definition does not require adjustments to the compiler.

Extensibility One key advantage of our formulation is its flexibility. In addition to the timber case study, the examples provided in the appendix demonstrate its versatility on a range of robotic construction problems. However, for repetitive assembly problems, our formulation only provides a generalized description framework but does not offer additional computational benefits (see Sequence and Motion Planning in Section 5.2). In future work, the method can be adapted to work on multi-robot, multi-element planning using the plan skeleton formulation. Our method is compatible with asynchronous multi-robotic movements (multiple robot agents operating at the same time), mobile robotic movements (e.g. robots with a non-holonomic mobile base [DSG+16]), or even non-discrete robotic movements (e.g. 3D printing [MBD20b]), as long as specialized motion planners are provided. For us, the creation of these planners falls onto the domain expert in planning.

Towards full TAMP Another compelling next step is to open up the parameters that are set by the designers when compiling a flowchart into a plan skeleton (Section 5.3.3). In this work, the construction sequence and the grasp pose are provided by the architect, using their intuition. But these values can be filled in by a automatic planning algorithm as well, which needs the planner to solve for both symbolic parameters (e.g. the construction sequence) as well as geometric parameters (e.g. grasp poses, robot trajectories).

Finally, our flowchart-based formulation is designed for offline, pre-planning purposes and currently do not support change of design after the fabrication has started or adaptive online re-planning beyond a complete re-computation. An interesting future direction is to investigate how to dynamically re-plan given incremental changes of design or certain scene observations.

As construction robotics advances from boutique examples to real-world deployment, the thoughtful combination of automation and human expertise becomes increasingly important. The methods presented in this chapter represent a key step towards this future.

Chapter 6

Constructability-driven design

6.1 Introduction

In this chapter, we investigate how automated planning techniques can be used to inform design decisions. Systematically considering performance early in the design process typically requires the development of quantification, calculation, and computer simulation of the desired performance metric. While PDE-based complex physics simulation tools have been used to derive various building performances [Bro19], formulating and computing construction-related performances need a different set of computational algorithms and tools to capture the sequential mechanical behaviors involved. As the first work that attempts to characterize construction performance and use it to inform design, we focus our attention on a simplified yet challenging type of assembly process: monotonic construction of discrete bar structures without scaffolding, where elements are added to the partial construction one-after-one. Following the motivations discussed in Section 2.1.4, We aim to equip designers with computational tools for answering the following questions in the conceptual design stage:

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

By adapting and extending the state-space search algorithm developed in Chapter 3, we devise computational means to (1) find a feasible construction sequence, (2) quantitatively compare different sequences, and (3) find the optimal sequence. In this work, the construction performance considered is entirely based on the structural behavior during construction. Future work could also include additional constraints like a constraint that ensure enough operational space for operators, etc. (see discussions in Section 6.6). The proposed automated search routine allows us to find high-performing construction sequences when the design complexity grows beyond human capacity to figure out the sequence by hand. Furthermore, we can endow a design with a constructability score by summarizing the performance of its high-performing construction sequences. The efficiency of the search algorithm enables us to iterate through a design catalog and choose the one that performs best in terms of construction efficiency.

6.2 Related work

Construction scheduling Research interest in automatically generating and optimizing construction schedules has been around since the early 1960s [NS+72]. This line of research aims to schedule the job tasks and required resources while satisfying the precedence constraints and resource conflicts among tasks and optimizing for overall construction time and cost. Much research efforts have gone into using different algorithms to solve several variants of this problem. Interested readers are referred to [FNRK15] for a review of these efforts. However, most previous research focuses on the scheduling of overall project progress instead of the assembly sequence of individual elements. Until recently, a few works start to investigate optimizing assembly sequences of individual elements [WYS18, FRK14, HPDC22]. The proposed methods use variants of the genetic algorithm to optimize the assembly sequence and use several geometric rules to ensure the partial construction's stability. These methods, due to the stochastic nature of the gradient-free optimization algorithms, do not scale to structures with hundreds of elements and do not have optimality guarantees. Furthermore, the geometric rules can only serve as a proxy for detailed physics simulation and thus the obtained result has no guarantee of the stability of the computed construction sequence. The algorithmic framework proposed in this chapter can be easily adapted to solve problems in these contexts.

Assembly planning Widely used in automated manufacturing and robotics, assembly planning is the process of creating detailed instructions to combine separate parts into a final structure. The goal of assembly planning is to find sequences of operations to assemble the parts (*assembly sequencing* [Jim13]), determine the motions that bring each part into its target pose (*assembly path planning* [GM15]), and propose the usage of additional resources such as formwork, supports, and fixtures to assist the assembly process. Most of the literature in this domain considers only geometric constraints: an assembly plan is defined as valid if there is no collision when assembling each part. Assembly planning problems can be classified into two categories: (1) finding one valid assembly plan for a given design, (2) finding a desired assembly plan to satisfy some objectives of the assembly process such as reducing the usage of fixtures, overall assembly time, etc.

Finding an assembly plan requires identifying movable parts and part groups at each intermediate assembly state, often leading to a combinatorial search problem with very large branching factors. This search is challenging for two reasons: (1) When expanding search nodes, the evaluation of whether the new sub-assembly is assemblable geometrically is not straightforward; (2) the branching factor is large. Wilson [Wil92] invented Directional Blocking Graphs and Non-Directional Blocking Graphs to solve the sub-assembly assemblability evaluation problem. To limit the large branching factor, the idea of *assembly-by-disassembly* has been widely used in the literature, where an assembly plan is obtained by disassembling the completed assembly into parts and then reversing the order and path of disassembly. The advantage of this strategy is that it can drastically reduce the solution space to be searched since parts are constrained the most geometrically in the completed stage [HLW00, GHLPM20]. Due to the challenges outlined above, most of the literature in assembly planning focus on *sequential* and *monotonic* assembly that only allows one part to move at each step and parts are not allowed to be removed once added to the sub-assembly. This work also falls into this category, but we focus on the physical constraint instead of the geometric constraint and explore the connection between the design variation and assembly plans.

Going beyond finding only one valid assembly sequence, assembly planning can be formulated as an optimization to find the desired assembly plan. Typical optimization objectives include minimizing assembly complexity (e.g. the shortest assembly path, simple assembly motion), minimizing the usage of additional resources (e.g. fixtures or supports to maintain the stability of partial assemblies [DPW+14]), and maximizing parts visibility for creating visual assembly instructions [APH+03, HPA+04].

To find an optimal assembly plan, we need to enumerate and evaluate all possible assembly plans and select the best-performing one based on the chosen objective. Although this is possible for assemblies with a few elements, e.g. by using AND/OR tree data structure [DMS90], the complexity grows exponentially with the number of parts involved. Due to this reason, various practical algorithms have been proposed to find suboptimal sequences, e.g. using a greedy algorithm [DPW+14, MSY+15], a heuristic search [APH+03], or an adaptive sampling followed by user editing [KKSS15].

Going beyond the context of mechanical product assembly, some works consider the assembly planning of architectural structures with a structural stability constraint. [BBW15] proposes a backward search algorithm to find stable deconstruction sequences of masonry structures. Divide-and-conquer algorithms have been proposed to first decompose the discrete structure into element groups and then use these groups as a partial ordering to accelerate the construction sequence search in both cable-assisted masonry construction [DPW+14] and robotic spatial extrusion [HZH+16]. Recently, [BAP22] uses graph rigidity theory to identify stable assembly and disassembly sequences for dual robot construction without scaffolding.

Assembly-aware design Works in assembly-aware design attempt to connect the design and the assembly sequence by not just searching assembly plans but also varying parts geometry and/or layout. This is an emerging direction with only a few works in the literature. This design methodology has been explored in the context of designing masonry shell structures that require less formwork during construction [KKS+17] and electromechanical devices where each part can be inserted with multi-step translational motion [DMC18]. [WMW+20] proposes a topology optimization formulation that simultaneously optimizes the structural density layout and the fabrication sequence for additive manufacturing.

6.3 Evaluating construction sequences

In this section, we first detail the assumptions that we make on the considered type of structural systems and construction process (Section 6.3.1). Then, we discuss the structural constraints and objectives for evaluating a given construction sequence, which is the foundation for the proposal of constructability scores in the next section. The overview of the entire design workflow can be found in Figure 6-1.

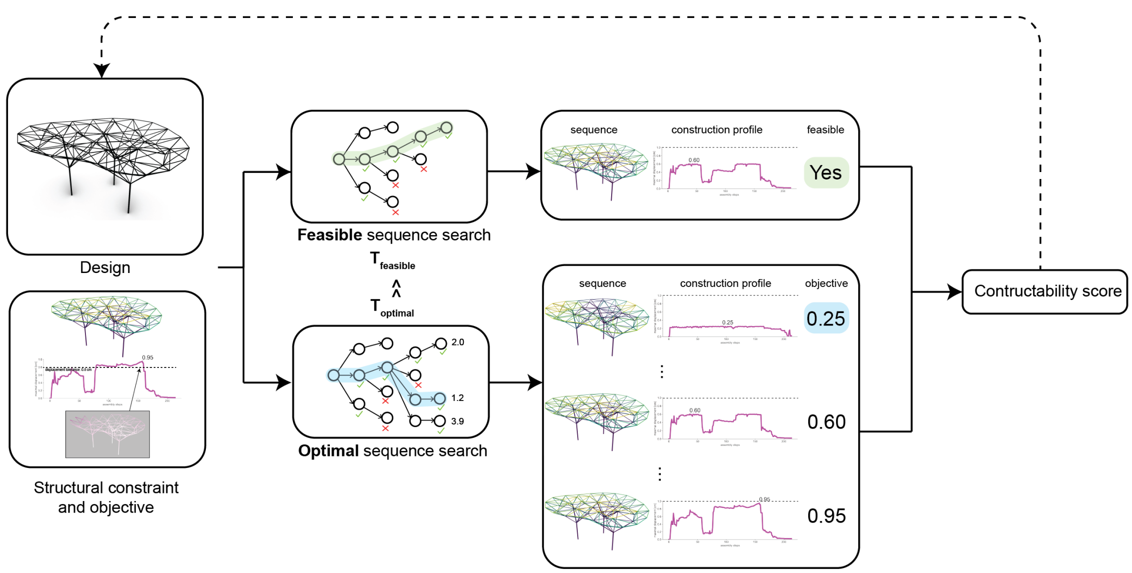


Figure 6-1: Overview of the proposed constructability-driven design workflow.

6.3.1 Assumptions on the structural system and construction process

In this work, we focus on a broad class of building structures, bar structures, which are composed of discrete linear elements and used broadly and in great variety in the built environment. A bar system can be described as a collection of n elements, where each element corresponds to a bar. Each element can be connected to one or more other elements. In this work, we assume that all elements are connected by rigid connections, which can transfer moment and torsional load to allow the partially constructed structure to cantilever, even if the primary structural action in the final structure is dominated by axial forces, as in truss systems. While the structure can be designed and sized according to specific nominated loads in its completed stage, we only consider forces induced by self-weights during the construction.

For a bar system, we define a construction sequence to be an ordering of elements that dictates the sequence in which they are added to the structure. In the scope of this work, elements are not allowed to be removed and later re-introduced once they are assembled in place. Furthermore, no scaffolding element is allowed to be introduced. We call such a construction process a scaffolding-free, monotonic process because the structure only grows with time. The scaffolding-free assumption makes the problem easier to solve mathematically, and also has construction advantages in terms of simplicity of logistics. Future work could integrate scaffolding through further algorithmic development (see Section 6.6 for more discussions).

6.3.2 Structural constraints and objectives on construction sequences

In this work, we consider structural constraints that are related to displacement or stress to check if a certain sequence is feasible. These structural considerations can be quantified by bounding a physical number obtained from physics simulation at each construction step. For bar-based structures with rigid joints, a *stiffness-based* simulation that often involves applying finite element methods is more relevant than the *stability-based* simulation that checks the force equilibrium of an assembly consisting of rigid parts (see Section 2.4.1 for more discussions). Mathematically, the constraint on a construction sequence is can be expressed by the following equation:

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

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

$$f(\psi) : \{\psi\} \to \mathbb{R}$$

A sequence planning algorithm aims to find a feasible sequence under constraints, or find the optimal sequence among a set of found feasible sequences under a prescribed objective metric.

Displacement The displacement constraint requires that a partially constructed structure's maximal nodal displacement is below a given tolerance. Each bar element experiences a self-weight load due to gravity, which causes the structure to bend downwards. Excessive displacement is undesirable since it leads to alignment issues or material failure in the construction process. In this work, the displacement of all nodes is calculated using a first-order linear elastic finite element analysis (FEA) of a 2D or 3D frame structure. The norm of the $x, y(, z)$ translational displacements at each node is compared to the given maximum permitted tolerance.

In Figure 6-2 illustrates different construction sequences for a 2D steel bridge with a span of 20 meters, rigidly supported at the four points on its horizontal boundary. The cross-sections of the steel elements are optimized for the gravity load and the in-service point loads of 10 kN per node, applied at the upper chord of the structure. In this case, a maximal displacement of 10 cm is set (shown as dashed line in Figure 6-2). In the maximum displacement constraint formulation, a feasible construction sequence must have the maximum displacement at each step bounded below this number. We can further plot the maximal nodal displacement against the construction steps. This plot, called *construction profile*, shows us the evolution of the physical attribute that we care about throughout the construction and is an essential tool for us to quantify construction performance. Figure 6-2 demonstrates that the same design, under the same displacement tolerance, can have feasible construction sequences with very different construction profiles. We need to formulate a metric to compare these sequences, so we can find the optimal sequence for a design with respect to the metric.

Depending on one's needs, there are many formulations for evaluating a construction sequence. The simplest of all is optimizing for the worst step during the construction, i.e. using the maximum value of the graph to represent the corresponding sequence. Mathematically, this corresponds to having the objective function equal to the constraint function: $f(\psi) = g(\psi[1:n])$. Such worst-case partial constructions are displayed in the cropped-out boxes for the two sequences in Figure 6-2. There are other formulations for evaluating a sequence based on its construction profile, for example, the Dirichlet energy that measures the smoothness of the displacement evolution and the integration of the graph that measures the total accumulated max displacement. The specific choice of this measure is left to the users. For simplicity, we focus on the maximum formulation in the rest of this work.

Stress Load-induced stress can be another physical aspect that one might concern about during construction. Excessive stress can lead to material failure that is catastrophic to the construction process. Thus, bounding the maximum stress throughout the construction is critical for the safety and the success of the construction process. Similar to the displacement constraint, the maximum stress can be calculated through FEA by taking the maximum of each element's maximal Von-

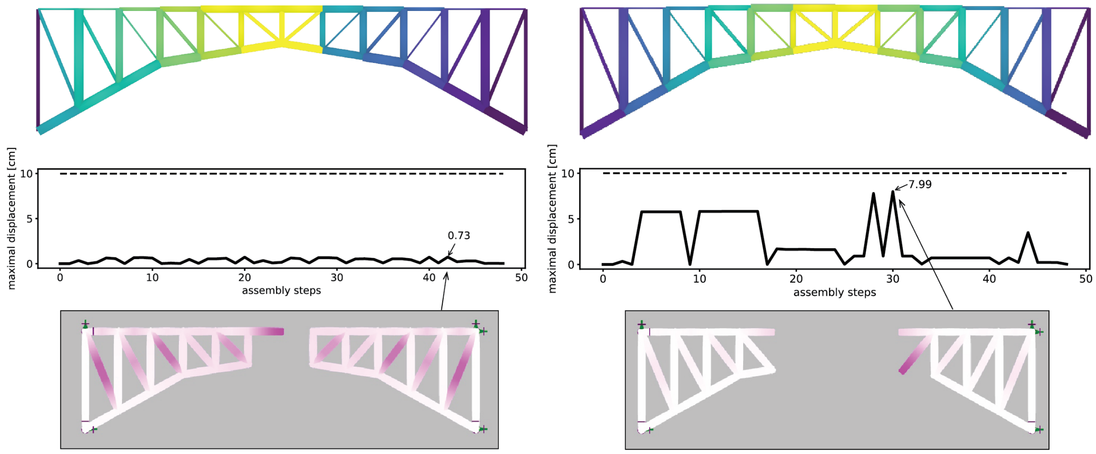


Figure 6-2: Two different construction sequences and their construction profiles (a graph tracking the considered physical attribute) for a 2D bridge design under the displacement constraint. The construction sequence is shown using a color map, with dark purple built first and yellow built last. The dashed line shows the set maximum permitted displacement tolerance (10 cm). Each point on the construction profile graph corresponds to a partial construction and the y-value of the graph is obtained from physics simulation. The cropped-out images show the construction steps that have the maximum displacement, color-coded by the displacement value from white to pink. With the maximum objective formulation, the sequence on the left has the objective value of 0.73 cm and the sequence on the right 7.99 cm.

Mises stress across its length. Figure 6-3 shows two feasible construction sequences under the stress constraint for the same 2D bridge design shown in last section, with a stress tolerance set to be 10 kN/cm^2. We can apply all the sequence objective formulations that we discussed above for the displacement constraint over to the stress constraint.

6.4 Formulating and computing constructability scores

The last section shows that under certain constraints, a design might have many feasible construction sequences. Resting on this framework of construction sequence evaluations, we aim to derive a definition of constructability score for a design to quantitatively answer the construction-related questions we raised for conceptual design in Section 6.1. We propose two types of constructability scores, one focused on *feasibility* and the other focused on *optimality*.

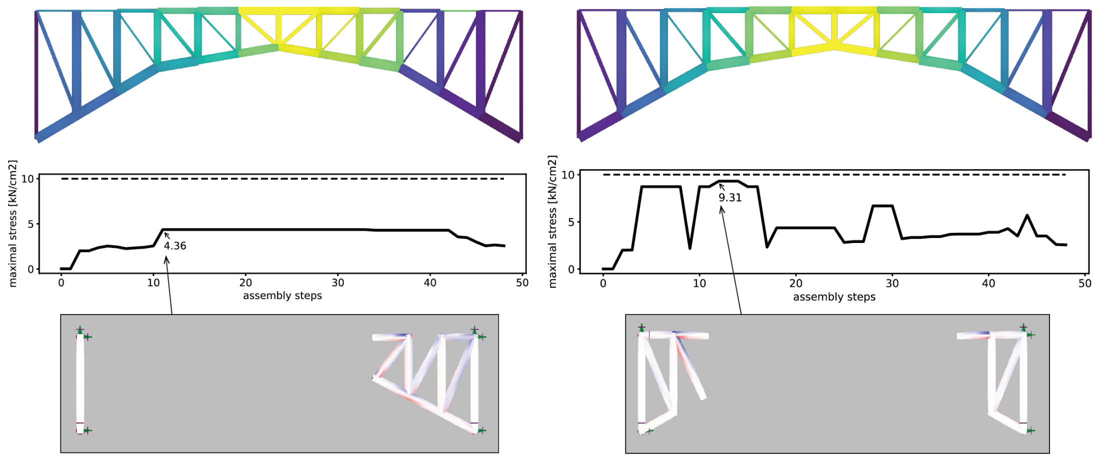


Figure 6-3: Two different construction sequences and their construction profiles for a 2D bridge design under the stress constraint. The construction sequence is shown using a color map, with dark purple built first and yellow built last. The dashed line shows the set maximum permitted stress tolerance (10 kN/cm^2). The cropped-out images show the construction steps that have the maximum stress, color-coded by stress (red compression, blue tension). With the maximum objective formulation, the sequence on the left has the objective value of 4.36 kN/cm^2 and the right 9.31 kN/cm^2.

The constructability score is computed by summarizing its set of feasible construction sequences for the chosen constraint. Due to the vast, combinatorial search space involved in the sequencing problem, algorithmic planning techniques are needed to find the desired construction sequences for a given design. The potential search space contains all the permutations of the elements, which contains the factorial of the total element number n and is far beyond the reach of manual exploration. Even for a simple 2D bridge design like the one shown in Figure 6-2, nuanced details in the construction sequence can have a huge impact on the overall profile. For example, the order of building the upper and lower chord of each cell panel in the bridge can lead to a sequence objective value that is 10 times lower than the other. These nuances in sequencing are very hard for a human to reason about manually, both intuitively and physically.

We adapt the state-space search proposed in Chapter 3 to find sequences. As shown in the previous chapters, forward state space search is more effective than backward search when only considering the structural constraints. In this case, a state includes only the partially constructed structure, and the transition action is simply adding an element. Thus, the set of possible states is large but finite, containing permutations of all the elements. The computation of a performance metric for a given design should satisfy the following requirements:

1. The simulation for computing the performance score should be deterministic, meaning that if we simulate the same design multiple times, the same performance score should be obtained.

2. The simulation should terminate in a reasonable amount of time. Ideally, it should also give users the option to terminate early with an approximated result.

This section details the technical modifications we make to the forward state-space search algorithm to satisfy these requirements. While computing the feasible constructability score requires small adaptions to the original search algorithm, computing the optimal score requires searching for the optimal construction sequence, for which we propose two new algorithms. A flowchart of the algorithms proposed for finding the feasible and optimal scores can be found in Figure 6-4.

6.4.1 The feasible constructability score

The feasibility score is proposed to answer the first question we asked in Section 6.1: "Is this design buildable?". In such cases, we only care about whether a feasible sequence exists or not. The planning algorithm is expected to terminate as soon as it finds a feasible solution without wasting time on any further exploration while being algorithmically complete meaning that it must find a solution if such one exists. Thus, a feasible constructability score can be simply defined by a binary value returned from the planner: 0 when a feasible sequence exists, and 1 when none exists. The only modification that we need to do to the original forward state-space search algorithm is to remove the randomness involved in its tie-breaking process when multiple candidate elements share the same heuristic value (see Section 3.5.3). By

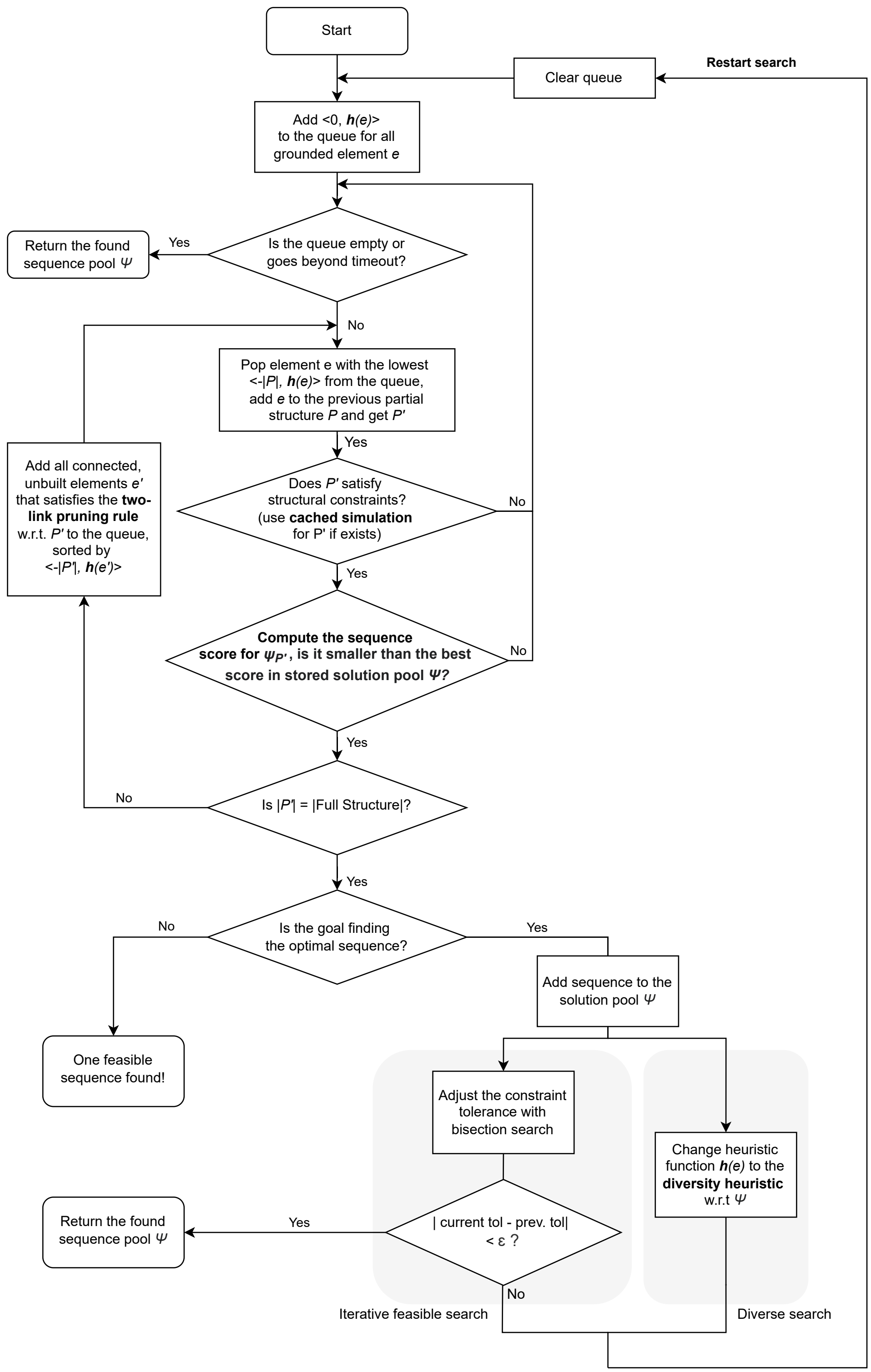


Figure 6-4: A flowchart describing the adapted forward state-space search algorithm, with versions for both feasibility and optimality. The two newly proposed algorithms for finding optimal sequences are highlighted in gray boxes.

doing so, we make the search routine deterministic. In this work, we simply use the lexicographic order of the element indices for tie-breaking.

6.4.2 The optimal constructability score

To answer the other questions we raised in Section 6.1: "what's the best way to build a design?" and "Are there other design options that are easier to build?", we define an optimality score that is based on the best-performing construction sequence concerning the chosen objective that we discussed in Section 6.3.2. Mathematically, the optimal score can be defined as:

$$F_{\text{opt}}(X) = f(\psi_*), \quad \psi_* = \arg\min_{\psi} f(\psi)$$

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

In response to the requirements in Section 6.4, we propose two universal techniques for limiting the branching factors and two new state-space search algorithms for finding the optimal construction sequences. First, we use a branch-and-bound technique to prune worse-performing branches and a cache to reuse FEA simulations. Second, we propose the two-link pruning heuristic to avoid wasting exploration time in search branches that will lead to worse-performing sequences. Then, we propose two search algorithms, one based on finding an optimal sequence by applying the feasible search with iteratively tightened tolerances and the other one based on a diversity-driven search. The iterative feasible search algorithm is guaranteed to find the optimal construction sequence with convergence criteria to terminate itself, but its applications are limited to the case when the sequence constraint and objective are the same functions (e.g. when both f and g compute the maximum displacment). In contrast, the diverse search algorithm works for any sequence constraint and objective formulations, but cannot terminate itself until reaching the user-specified timeout. Finally, we end this section with a remark on the two proposed algorithms.

6.4.2.1 Branch pruning techniques

Branch-and-bound and caching To accelerate the search, we need to prevent the search from exploring branches that we know a prior that will lead to worse-performing sequences. We use the well-known branch-and-bound technique [NKK84] that compares the performance score of the sequence ψ_P from a partial construction P with the best-performing sequence found in the solution pool Ψ. If ψ_P performs better than the best score found so far, we add it to the search queue and continue, and otherwise, the search node is discarded. To save the computational overhead from duplicated FEA simulations on the same partial construction P, we maintain a global cache of the performance score of P for reuse whenever possible.

Two-link pruning rule To further limit the branching factor, we propose a simple graph-based heuristic that uses an ad-hoc connectivity rule to prune less viable successors in the search. Given a partial construction P and the new candidate element e that is connected to P in the final construction, we check if e is connected to P *only* via a node of valence two (Figure 6-5-(3)). If so, e is pruned and will not be added to the search queue as a successor. Otherwise, it will be added as a successor as usual. Intuitively, we are limiting the search from exploring a partial construction that has two successive cantilevering elements, which are known to be not stiff and lead to higher displacement and stress. This can be seen as a type of look-ahead by using a heuristic without involving the extra computational overhead that usually comes with a forward-checking algorithm for early dead-end detection (see Section 3.6.1). This heuristic is also used in [GHB18] in accelerating the planning of a feasible free-form printing sequence. However, while this heuristic is effective empirically and easy to be implemented, using it might cost the planning algorithm to lose its completeness, since it strictly prunes certain worse-performing branches that might still be feasible. For example, for the dendriform structures that only have a few grounded nodes that are later branched out (Section 6.5), the two-link pruning rule is too strict and prevents the planner from finding any feasible result. Thus, specifically for these types of designs, we add an exception that if the candidate element e is connected to the built element e' in the partial construction via a valence-two node, e is still considered viable if the other node of e' is grounded (Figure 6-5-(4)). Because of the easiness of implementing such branch pruning heuristic, the users can use adapt similar domain-specific strategies to help the planner explore the search space more effectively.

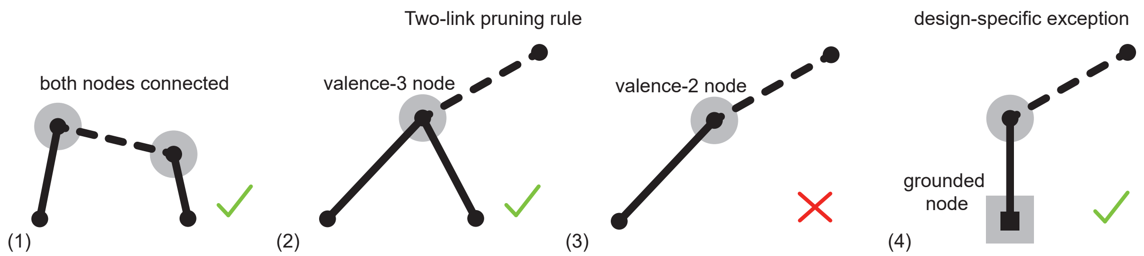


Figure 6-5: Illustrations of the two-link pruning rule. The solid lines represent the partial construction P and the dashed lines represent the candidate element e under consideration. (1-2) e is a valid successor for P since e is connected to P via two nodes or a node with a valence bigger than two; (3) e is not a valid successor for P since e is connected to P via a valence-two node; (4) a design-specific exception for the dendriform structure in Section 6.5: e is connected to P via a valence-two node, but is deemed as a valid successor because its connected element's other node is grounded.

6.4.2.2 Iterative feasible search for finding the optimal sequence

When the sequence constraint and objective function are the same, we are essentially solving an optimization problem of the following form:

$$\begin{aligned} \min_{\psi,t} \quad & t \\ \text{s.t.} \quad & g(\psi[1:k]) \leq t, \quad \forall k \in [1, \cdots n] \end{aligned} \tag{6.1}$$

where ψ is the construction sequence and t is the construction tolerance. A similar formulation is commonly used in the optimization literature for finding a feasible initial guess [NW99]. In this work, we use a bisection search to iteratively tighten the constraint tolerance t, and in the inner loop, we use the feasible search discussed Section 6.4.1 with the two branch-pruning techniques covered in Section 6.4.1. The algorithm is terminated when the lower and upper range considered by the bisection search is smaller than a user-specified tolerance ϵ. This algorithm is guaranteed to find the optimal t in a finite number of iterations, assuming the timeout for the inner-loop feasible sequence search is large enough.

6.4.2.3 Diversity-driven search for finding the optimal sequence

When the sequence objective and the constraint functions are different, we are solving an optimization problem of the following form:

$$\begin{aligned} \min_{\psi} \quad & f(\psi) \\ \text{s.t.} \quad & g(\psi[1:k]) \leq t, \quad \forall k \in [1, \cdots n] \end{aligned} \tag{6.2}$$

where t is a fixed constraint tolerance.

In such cases, the iterative algorithm proposed in the last section cannot be applied, since the feasible search cannot only find one feasible sequence under the tolerance, and it does not use any information of the objective function. We need to change the search algorithm so that it can find multiple, high-performing sequences. However, there are two main challenges: (1) how to prevent the search algorithm converge to previously found solutions and (2) how to bias the search to get out of local minima effectively.

In this work, we propose a diverse search technique to address the first challenge above, while the second challenge remains unsolved. In general, diverse search techniques usually involve using previously found solutions to bias or deflate the search away from existing solutions (see, e.g. the deflation technique in nonconvex optimization [TH22]). Simply letting the search continue by exploring the remaining search queue after the first feasible solution is found will leave the search converging to similar solutions. This is because the search explores the search node ordered in a depth-first manner (see Section 3.5.3 for details in queue ordering), the restarted search will get stuck at local minima, wasting most of the time exploring the permutations of a small set at the rear of the found sequence. This means that the algorithm only explores a limited set of solutions with potentially similar performance scores in the given timeout. To find a more diverse set of solutions, we propose to restart the search by clearing the queue whenever a new solution is found and replace the heuristic functions that we proposed in Section 3.5.3 with a new, diversity-driven heuristic function:

$$h(e, P, \Psi) = -\frac{1}{|\Psi|} \sum_{\psi \in \Psi} \sum_{e_k \in P \bigcup \{e\}} |\psi(e_k) - \psi_P(e_k)| \tag{6.3}$$

where Ψ is the collected construction sequences so far. P is the partial construction of the current search node, and e is the candidate element considered. Here, a construction sequence $\psi = [e_1, \cdots, e_n]$ is viewed as a mapping function $\psi : e \to \mathbb{N}$ that maps an element to its index in the sequence. Intuitively, this heuristic function is computing the weighted average of the accumulated permutation difference between the uncompleted sequence ψ_P and each found sequence $\psi \in \Psi$. This diversity score penalizes new sequence ψ_P from having too many local permutations compared to the found sequences and thus encourages the search to prioritize search nodes that lead to more diverse solutions. The absolute value in the second summation in Equation (6.3) computes the difference between e_k's ordered position in the sequence ψ_P and its ordered position ψ, for each element e_k in the partial construction $P \bigcup \{e\}$. This formulation for computing a distance measure between two permutations is also called the Position distance [SS07]. There are many other distance measures for comparing permutations that are widely used in stochastic local search literature for comparing two solutions [SS07, SRP05, ZSB14]. Experimenting with other permutation distances is left as future work.

6.4.3 Remarks

The practical usability of the optimal score proposed in this section depends on whether the used search algorithm can find the optimal sequence within a certain amount of time. The two optimal search algorithms proposed in the last section are targeted for different types of problems and have their advantages and limitations. In Section 6.5.2, we show empirical results comparing these two algorithms in finding the optimal sequence for a given design.

The iterative search algorithm can only solve the problems that have the same objective and constraint functions. However, while limited to a specific type of problem, the algorithm is guaranteed to find the optimal sequence in finite steps. In Section 6.5, empirical results show that the algorithm can find the optimal solution quite effectively. For example, for a structure with 214 elements, the algorithm can return the optimal solution within 12 minutes.

In contrast, the diverse search algorithm can work for arbitrary combinations of objective and constraint formulation, e.g. minimizing the accumulated stress while ensuring the maximum displacement is under a tolerance. But such generality comes at the cost that the algorithm loses the ability to converge to the optimal solution in finite steps. Like many other heuristic algorithms proposed for complex combinatorial optimization problems, e.g. the Traveling Salesman Problem [ABCC11], the proposed diverse search algorithm for finding an optimal sequence is suboptimal, meaning that it can only find a relatively good solution in the given amount of time, without any guarantee on whether the solution is a local optimum or not. Because of the combinatorial nature of the problem, we do not yet have a convergence criterion for finding local minima for terminating the search early, like the KKT conditions used in many gradient-based optimization algorithms [NW99]. In the proposed algorithm, the tolerance used for the structural constraint, the heuristic function used for finding the first feasible solution, and the timeout are the only three parameters that the users can control to improve the quality of the found sequences. An important consequence of having a suboptimal search algorithm on the optimal constructability score is that the search might get stuck in different local minima for different designs, so the corresponding optimal scores are all higher (worse) than the true values in different degrees. The magnitude of such discrepancies are likely not even for all designs considered, thus the computed performance score can favor certain designs, which we cannot estimate a priori. Thus, improving the search algorithm's optimality or finding a way to compute a lower-bound of optimal sequence score is an important direction for improvement of the currently proposed algorithm (see Section 6.6 for detailed discussions).

6.5 Results

We demonstrate our proposed strategies on a parametrically designed, high-performing, small-scale double-layer roof structure (Figure 6-6). This structure consists of 214 elements and 62 nodes, with a physical dimension of 167 cm $\times$ 151 cm $\times$ 77 cm (length, width, height). Note that this roof structure is scaled down to the size of a dining table or desk. The design loads considered for the completed structure are point loads of 4 N per node, applied at the nodes on the rooftop. The cross-sections are optimized for these designed loads by using the cross-section optimizer of Karamba3D [Pre13]. The structure is conceived to be built with cellulose tubes, the material properties of which can be found in Table C.1 in the appendix. After the cross-section optimization, the completed structure is expected to have a total mass of 2.16 kg and a service displacement of 0.28 cm. For future benchmarking purposes, the detailed design data is presented in Appendix C.

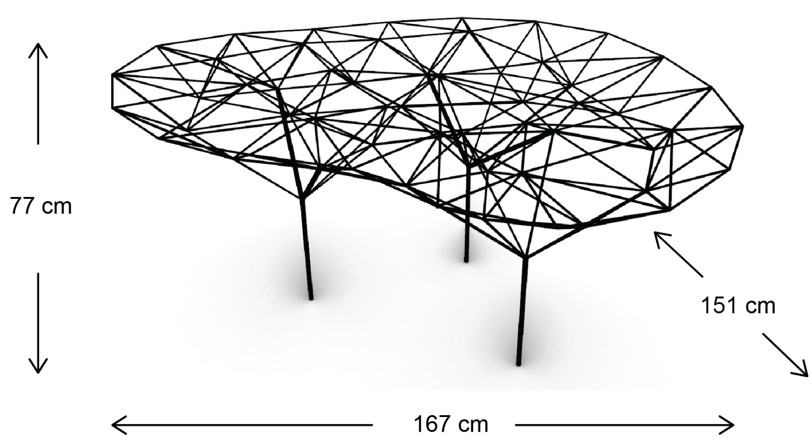


Figure 6-6: Overview of the double-layer roof design.

While the structure's service mechanical behavior in its completed stage can be predicted using standard FEA tools, the in-construction behavior is much harder to estimate without the physically constrained automated planning techniques proposed in this chapter. The design is complex enough that it is hard for a designer to have an intuition on whether this design can be built without scaffolding or not. In this section, we demonstrate two use cases: (1) comparing construction sequences found by various configurations of the search algorithm (Section 6.5.1) and (2) comparing design options using the optimal constructability score (Section 6.5.3). In addition, we provide an empirical comparison for the two search algorithms for finding optimal sequences in Section 6.5.2. For the structural constraint and objective, we focus on constraining and minimizing the maximum displacement in all experiments. All the experiments presented in this section are performed by an IronPython implementation inside the GHPython environment in Grasshopper3D [Rob22]. The structural analysis used for checking the structural constraint and objective is based on Karamba3D [Pre13]. All experiments were performed on a consumer-grade laptop, without any parallel or GPU acceleration.

6.5.1 Comparing construction sequences

In this section, assuming we want to bound the maximum displacement by 0.8 cm, we show three sequences found by the sequence search algorithm under various configurations. Key search statistics are summarized in Table 6.1.

Table 6.1: Statistics of three construction sequences found by three search configurations. The first two rows are run with the feasible search using the EuclideanDist heuristic (see Section 3.5.3.2 for details about EuclideanDist). The optimal sequence is found by the iterative feasible search algorithm with a starting tolerance of 1.7 cm, full search statistics of which are given in Section 6.5.2.

Sequence	Score (cm)	Displacement tolerance (cm)	Timeout (s)	Num. state evaluation	Num. structural constraint failure	Runtime (s)
an infeasible sequence	infeasible	N/A	180	214	0	0.19
a good sequence	0.6	0.6	180	615	401	15.2
the optimal sequence	0.24	1.7	N/A	9354	8029	147.5

An infeasible sequence As a baseline, we start with an infeasible sequence found by a forward state-space search without checking the structural constraint (Figure 6-7). The obtained sequence seems like a viable solution at the first glance since it is computed using the EuclideanDist heuristic, i.e. ordering the elements by the z coordinate of the midpoint of each element, which is a widely used, intuitive heuristic used in many construction practices. However, the displacement construction profile reveals that the sequence is not feasible, and has about 40 steps in the sequence with maximum displacements above the specified tolerance. Thus, this proves that the quantitative analysis of construction sequences using the construction profile is important for revealing the detailed structural behavior during construction. In practice, such construction profiles have been used for examining and predicting the process of bridge constructions with non-standard scaffolding systems [FSC+12].

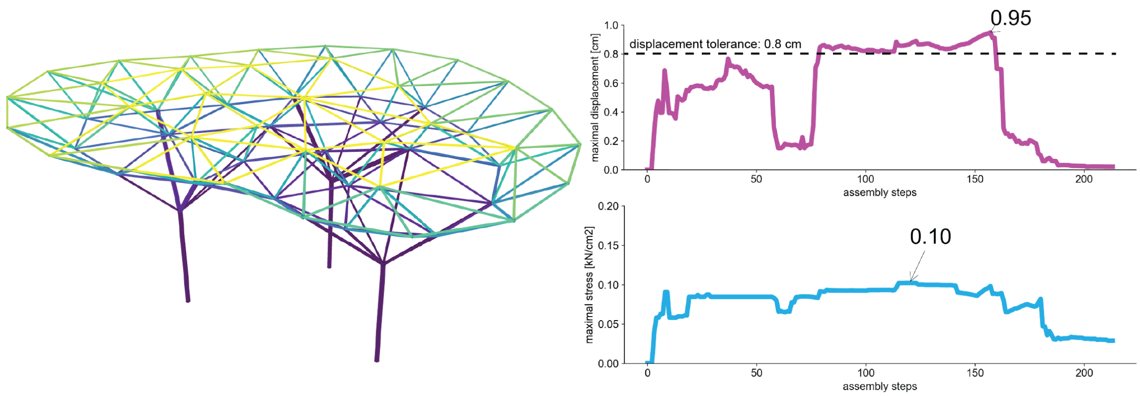


Figure 6-7: An infeasible sequence found by the search algorithm without considering the structural constraint. The construction sequence is color-coded with purple built first and yellow built last. The construction profiles on the right (top displacement, bottom stress) show that the maximum displacement of the sequence is above the given tolerance of 0.8 cm.

A good sequence Next, we look at a good sequence found by the same search algorithm as before, with the only difference that the structural constraint checking with a displacement tolerance of 0.6 cm is included. The displacement profile illustrated in Figure 6-8 shows that the maximum displacement is successfully suppressed under the given tolerance, although with the compromise of increased maximum stress during construction.

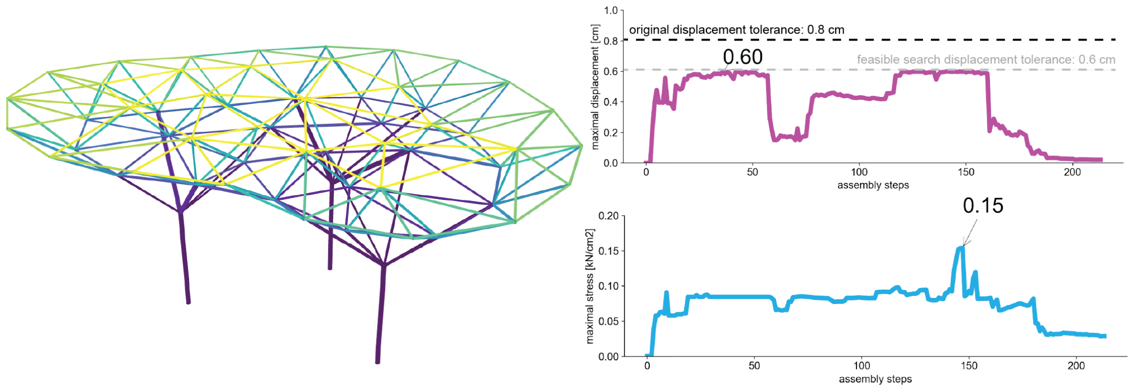


Figure 6-8: A good sequence found by the search algorithm with a maximum displacement tolerance of 0.6 cm. The displacement profile (right-top) shows that the obtained sequence remains under the given tolerance throughout the construction. The stress profile (right-bottom) shows an increase in the maximum stress, compared to the sequence shown in Figure 6-7.

The optimal sequence Finally, we look at the optimal construction sequence that minimizes the maximum displacement (Figure 6-15), computed by the iterative feasible search algorithm in Section 6.4.2.2. A step-by-step illustration of the sequence is shown in Figure 6-15. The construction profiles suggest that the maximum displacement of the sequence is 75% below the infeasible sequence while maintaining the same maximum stress value. This result demonstrates the power of the proposed automated search technique - it is capable of finding the best-performing construction sequence that is otherwise impossible to obtain. Detailed statistics of the iterative search algorithm for finding this sequence are discussed in the next section.

6.5.2 Comparing the two algorithms for computing the optimal sequence

In this section, we compare the two optimal search algorithms by having them compute the optimal sequence to minimize the maximum displacement for the double-layer roof design.

First, the iterative feasible search performs 7 search iterations to find the optimal sequence in 147.5 seconds. The search statistics per iteration are given in Table 6.2.

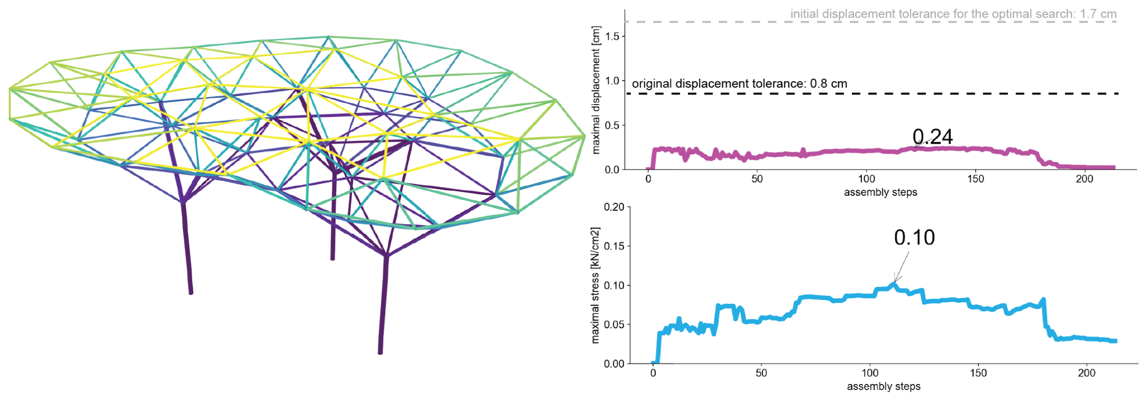


Figure 6-9: The optimal sequence found by the iterative feasible search algorithm. A maximum displacement tolerance of 1.7 cm is used as the initial tolerance for the optimal algorithm. The construction profile shows a 70% reduction of maximum displacement over the given tolerance. The sequence is illustrated step-by-step in Figure 6-15.

We can observe that when the search terminates fast when the tolerance is either too high or too low. When the tolerance is high, the search finds a feasible solution without encountering many constraint failures. When the tolerance is low, the search terminates early because no search branch can satisfy the stringent constraint. When the tolerance is approaching the optimum, the search time increases because the search encounters more constraint violations along the way. Notice that iteration 7 has the most structural constraint failures while having a relatively small runtime, thanks to the FEA caching techniques. A histogram of the obtained sequences is shown in Figure 6-10 and the color-coded sequences with their displacement profiles are shown in Figure 6-11. Figure 6-11 shows that the obtained sequences are visually similar especially viewed from a color-coded perspective, but their performances are very different. This further confirms the strength of the search algorithm in effectively exploring nuanced differences in the sequences to improve the performance.

The diverse search algorithm is given a one-hour budget for computing the best-performing sequences that it can find. The search statistics are presented in Table 6.3. Since the algorithm does not have convergence criteria to terminate itself, it used the entire hour, resulting in 20 sequences. The score distribution of these

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

Table 6.2: Detailed search statistics of the iterative feasible search algorithm, with a starting displacement tolerance of 1.7 cm. At each iteration, the feasible search algorithm is given a 180-second timeout.

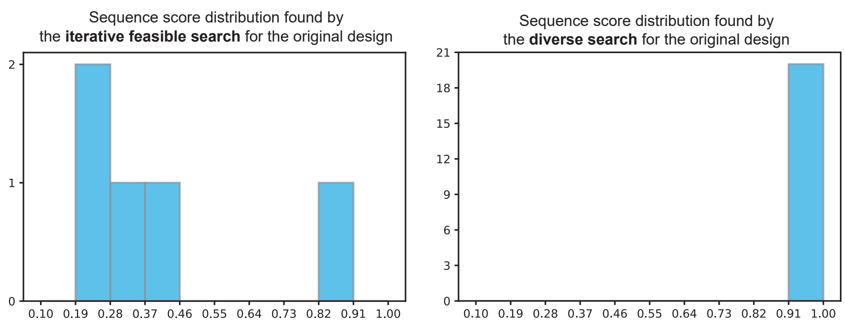


Figure 6-10: Score histograms for the found sequences of the two optimal search algorithms.

sequences is shown in Figure 6-10. The histogram shows that the found sequences all have similar performances around 0.94. Compared to the true optimum value of 0.24 found by the iterative feasible search algorithm, this shows that the diverse search failed to make much progress in minimizing the maximum displacement. Compared to the score value of 0.95 from the baseline sequence obtained by searching without checking the structural constraint (Figure 6-7), the best sequence found is scored 0.94, only improved by 1%. However, the eight top-performing sequences shown in Figure 6-12 reveal the diversity of sequences uncovered by the algorithm. These results show that the diverse search algorithm works well for uncovering a diverse set of sequences but performs poorly at optimizing the score.

Putting the two algorithms together, Table 6.3 shows that the iterative feasible search outperforms the diverse search in both solution quality and total runtime. This echoes with a general observation in optimization: constraint satisfaction alone is much easier than constrained optimization. As a result, if we want to

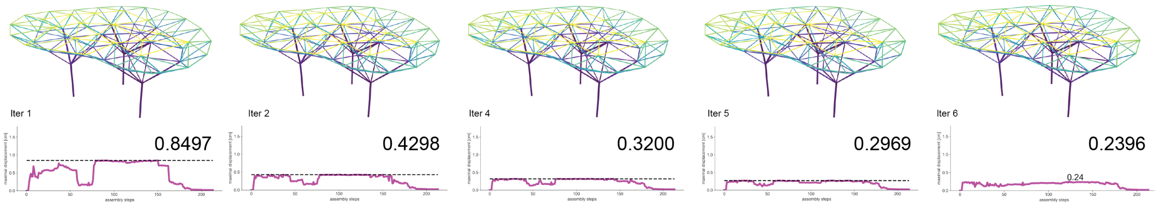


Figure 6-11: The feasible sequences found by the iterative feasible search.

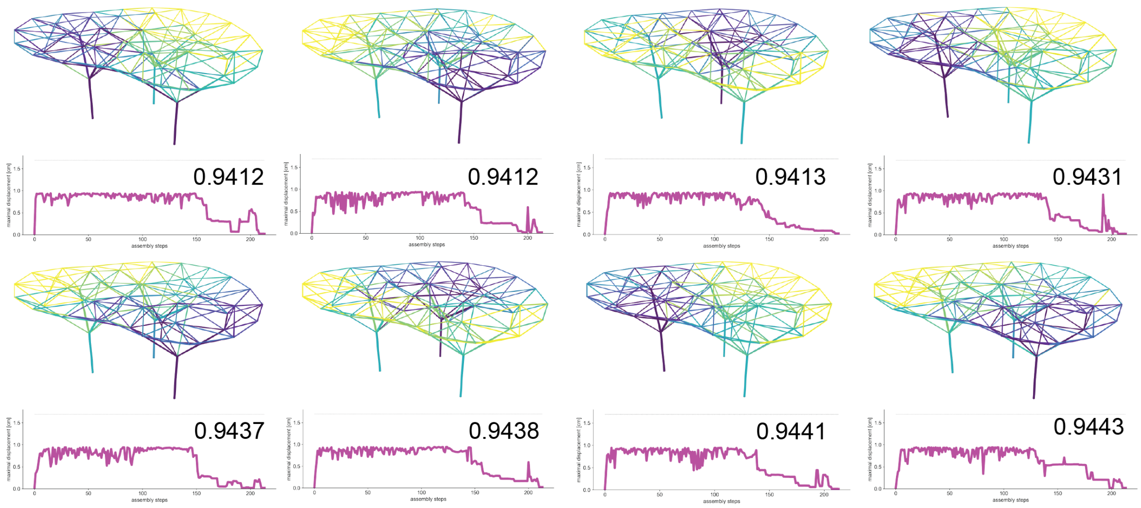


Figure 6-12: Top 8 sequences out of the 20 sequences found by the diverse search.

minimize the maximum displacement or stress, the iterative feasible search algorithm should be used.

6.5.3 Comparing design options

Finally, we use the iterative feasible search to compute the optimality constructability score to compare design options. Figure 6-14 shows the three designs we consider, including the original design as above and two design variations generated using the parametric design model. Table 6.4 shows the statistics for running the optimal search on the three design instances. The step-by-step illustrations of these optimal sequences are shown in Figure 6-15 (original design), Figure C-1 (variation 1), and Figure C-2 (variation 2). Design variation 1 and 2 both have more grounded elements supporting the roofs, and are both heavier than the original design. Design variation 1 dominates the other two in terms of the optimal constructability score, and it also has the lowest maximum stress value. However, this easiness of

Table 6.3: Search statistics comparing the two proposed optimal search algorithms. Both algorithms are run with a displacement tolerance of 1.7 cm, the EuclideanDist heuristic, and a one-hour timeout.

Search algorithm	Num. of state evaluation	Num. of structural constraint failure	Num. of solutions found	Sequence score average	Sequence score std.	Sequence score minimum	Runtime (s)
Iterative feasible search	9354	8029	7	0.42	0.22	0.24	147.5
Diverse search	36429	11929	20	0.95	0.003	0.94	3600

construction comes at the cost of its having more elements and consuming more materials, which is a trade-off to be considered by the designers.

Looking at the histograms of the computed sequences obtained by the optimal search algorithm (Figure 6-13), we see that the original design and design variation 1 have more feasible sequences with lower scores than design variation 2. Although not explored in detail here, these histograms might suggest another way to evaluate the construction robustness of a given design - a more robust design should have more feasible sequences at the lower range in the score spectrum.

Design	Iters	Num. of state evaluation	Num. of structural constraint failure	Num. of solutions found	Sequence score average	Sequence score std.	Sequence score minimum	Runtime (s)
Original design	7	9354	8029	5	0.42	0.22	0.24	147.5
Design variation 1	8	516787	503215	5	0.35	0.21	0.18	701.39
Design variation 2	7	817633	727922	4	0.42	0.26	0.2	561.1

Table 6.4: Search statistics of running the iterative feasible search algorithm on the three design options. All three instances are run with a displacement tolerance of 1.7 cm, the EuclideanDist heuristic, and a one-hour timeout.

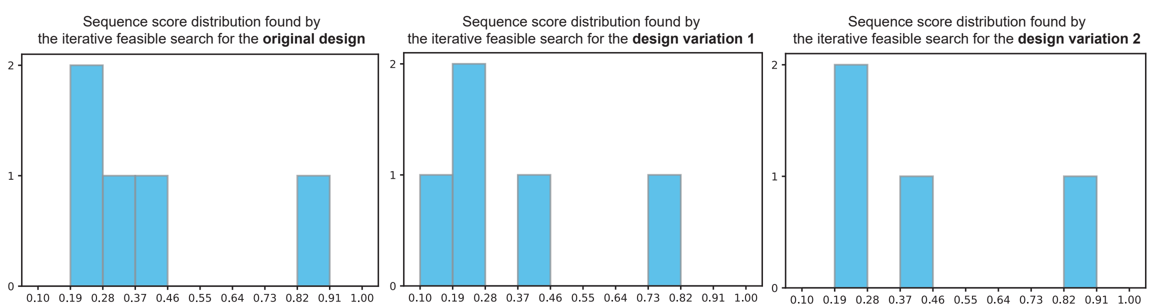


Figure 6-13: The score histograms for the sequence found by the iterative feasible search algorithm on the three design options.

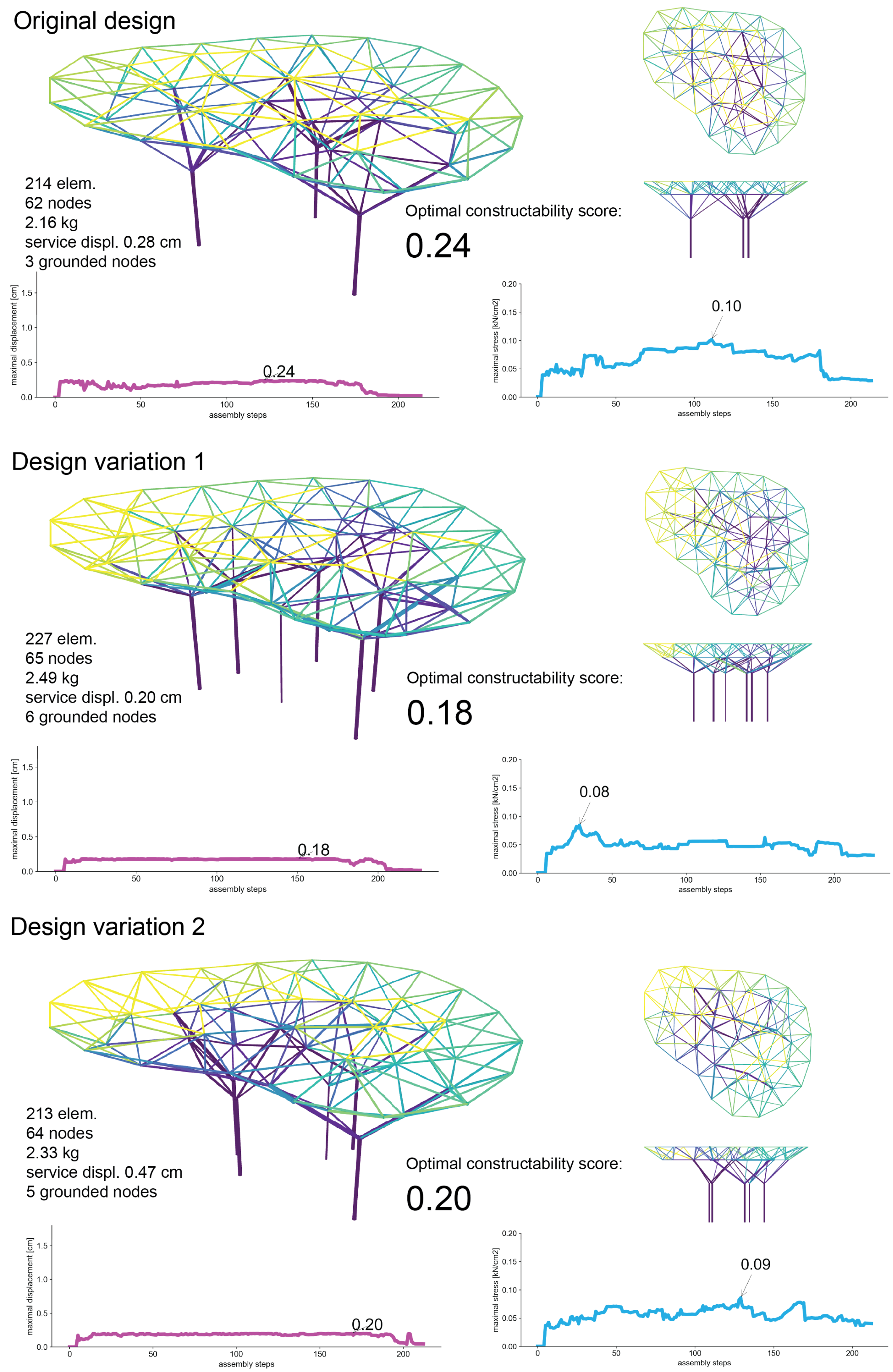


Figure 6-14: Comparing three design options using the optimal constructability score.

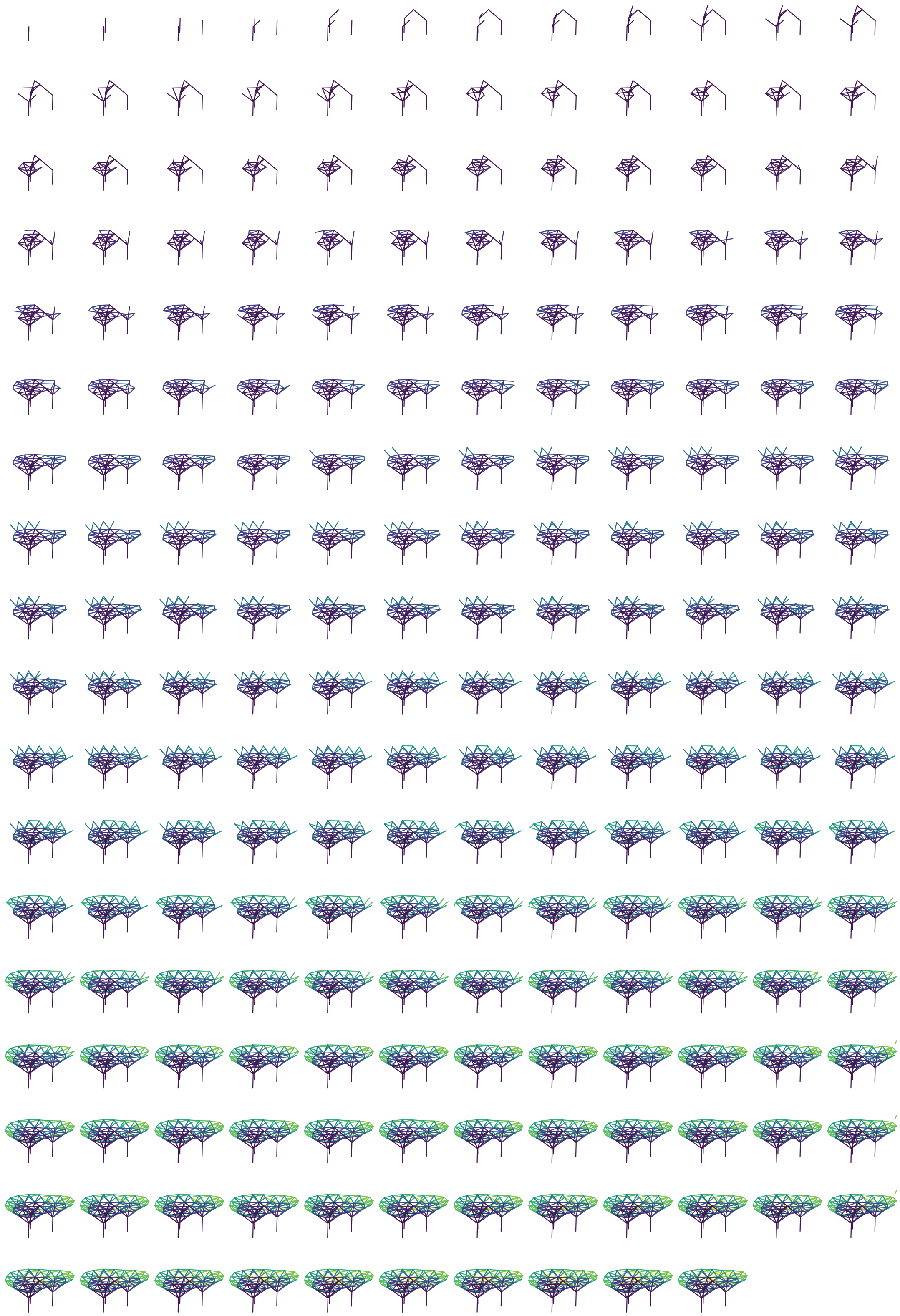


Figure 6-15: The optimal construction sequence for the original design (see Figure 6-9), found by the iterative feasible search algorithm.

6.6 Limitations and remaining challenges

In this chapter, we have demonstrated how to use automated planning techniques to support constructability-driven design. However, many technical challenges remain before we can connect the newly proposed constructability scores with a broader set of design space exploration techniques and go beyond the scaffolding-free construction typology.

First, we have only demonstrated evaluating a discrete design catalog under the optimal constructability score for minimizing maximum displacement. For problem instances that have different functions for structural constraint and objectives, the only proposed algorithm that supports these types, the diverse search, does not perform well in optimizing the objective and takes a long time to terminate. The diversity heuristic used in the diverse search knows nothing about the performance and thus the restarted search relies on the branch-and-bound technique to bias toward a better-performing solution, which can only make slow progress in improving the objective. The unreliability and slowness of this optimal search routine mean that including the optimal constructability score in a formal optimization loop is technically not feasible at this moment. Future directions might include (1) accelerating the search routine while maintaining bounds for optimality and (2) richer feedback from the planner to the outer-loop optimizer to more efficiently guide the optimization. A more detailed discussion on these can be found in Section 7.3.3.

Second, while only structural constraints and objectives are discussed and experimented with in this chapter, the search algorithm is modularized and can be easily extended to include other constraints. These constraints include, for example, operational-space constraints that ensure that there is enough space for either human or robot operators to assemble the element, or collision constraints that ensure the existence of collision-free robotic assembly motions. However, including these additional constraints exacerbates the computational overhead involved in the search. Advances in optimal search algorithms discussed in the last paragraph remain the key to devising search algorithms that can support these additional constraints while maintaining computational tractability.

Finally, while scaffolding-free construction eliminates any waste that might come from the scaffolding system, such a "minimalist" construction process might not be practical for many structures. Introducing a few scaffolding elements might dramatically stabilize the partial construction and thus increase the robustness of the process. However, planning for a minimal set of scaffolding elements poses an extra layer of complexity to the planning algorithm. Planning for an optimal construction sequence with a minimal set of scaffolding elements requires new ways to evaluate the construction plan, as well as new planning algorithms to compute such plans. Achieving this is important for broadening the impact and relevance of the constructability-driven design paradigm proposed in this chapter.

6.7 Conclusions

This chapter has presented a new way to characterize construction performance that can be used to inform design decisions. Through the use of various automated planning techniques, designers can quantifiably compare various construction sequences for a given design and have an automated way to compute optimal construction sequences. This is important because construction sequences are hard to evaluate with intuition and optimal construction sequence is impossible to find manually due to the vast combinatorial search space.

The approaches presented here extend the state-space search algorithm proposed in Chapter 3 to support searching for a feasible and optimal construction sequence for the assembly of scaffolding-free bar structure under certain structural constraints and objectives. The specific contributions developed in this chapter are as follows:

- **Quantifying construction sequence's performance**: We present several formulations for evaluating sequence performance that involves the structure's material and mechanical behavior during construction.

- **Constructability score formulation**: We present two formulations of constructability score, one for feasibility and the other for constructability, to a given design by summarizing the performance of the set of its construction sequences.

- **Computing the optimal sequence**: We present two new optimal search algorithms for finding optimal construction sequences, and provide empirical results to discuss their pros and cons.

- **Constructability-driven design space exploration**: We present several case studies to illustrate how we can use the proposed constructability score to inform design decisions on a discrete design catalog.

Chapter 7

Conclusion

This thesis has presented formal planning techniques to automate the programming process for robotic assembly (and more generally systematic assembly) of building structures. Previous chapters have motivated the problem, reviewed background literature, and presented algorithmic planning strategies for a broad class of assembly problems. In this final chapter, we conclude by summarizing the contributions of this thesis (Section 7.1), highlighting potential impacts (Section 7.2), discussing directions for future work (Section 7.3), and reflecting on the dissertation as a whole (Section 7.4).

7.1 Contributions

The key findings of this thesis are summarized as follows:

- Formulating assembly problems formally can help us devise novel, scalable and efficient search algorithms to fully automate the planning process (Chapter 3 and Chapter 4). The proposed algorithm, for example, can find the extrusion sequence and detailed robot motions for truss models of 300 elements in 10 minutes.

- The planning of complex, multi-tool assembly processes with non-repetitive plan skeletons benefits from a versatile pipeline to convert the high-level design and construction intents to a plan skeleton, upon which an effective solving technique can be applied (Chapter 5). This enables automated planning of robotic motions for a multi-tool robotic timber assembly process that involves 2400 robotic actions within 4 hours of computational time, which has not been shown possible before.

- Using automated planning tools to formulate and compute a new type of construction-related performance measure is demonstrated in a scaffolding-free assembly setting. We show how we could use this constructability measure to evaluate and drive the exploration of design options (Chapter 6).

Detailed contributions from each chapter are summarized as follows. First, we present a comprehensive survey spanning work from six different fields in Chapter 2. We provide a specific in-depth historical presentation of work in construction robotics, with an emphasis on the second, ongoing wave of development that takes a design-centric instead of a productivity-centric approach. After 13 years since the start of modern research in architectural robotics, we present the first attempt in the field to categorize major works according to their fabrication processes. From this taxonomy, we derive insights from existing research trends and identify a universal need for better planning tools. Then, we provide a technical review for related works in adjacent fields of performance-driven structural design, design for manufacturing, computer graphics, and automated planning.

In Chapter 3, we present a rigorous formalization for solving sequence and motion planning in the context of robotic spatial extrusion. We reveal the key planning challenges is, throughout the printing process, satisfying both stiffness constraints that limit the deformation of the structure and geometric constraints that ensure the robot does not collide with the structure. We show that, although these constraints often conflict with each other, a greedy backward state-space search guided by a stiffness-aware heuristic can successfully balance both constraints. The proposed algorithm is the first sequence and motion planning algorithm that is scalable, probabilistically complete, and without the need for human intervention. An empirical benchmark on 40 simulated extrusion problems, along with three real-world verifications, is presented to demonstrate the effectiveness of the proposed algorithm. In Chapter 4, we present a generalization of this extrusion planning algorithm for jointly finding an assembly sequence and robot motion plan for the assembly of bar structures.

In Chapter 5, we tackle assembly processes that have a non-repetitive pattern of robotic primitive behaviors, going beyond the sequence and motion planning problems that we covered in the previous two chapters. Our investigation is motivated by the necessity of robotic modeling and planning for a recently published timber assembly process that utilizes distributed robotic clamps to press together interlocking joints. In addition to pick-and-place operations, the robot needs to move numerous tools within the construction scene. To facilitate an agile process for connecting architectural design, construction process design and task and motion planning, we introduce a flowchart-based specification language that allows various designers to describe their design and construction intent and knowledge. Then, a sequence of compiling and solving techniques are introduced to convert such high-level intentions to a computable plan skeleton and finally solved by calling motion planners. The proposed workflow has been validated by a large-scale, real-world robotic construction experiment. We also showed how other recently published robotic assembly processes can be formulated using our flowcharts to demonstrate their generalizability.

Finally, in Chapter 6, we demonstrate how the proposed algorithmic planning techniques can be incorporated into a design space exploration workflow. We propose a way to measure constructability and show how we can use this measure as a performance score to drive structural design space exploration.

7.2 Potential impacts

In building algorithmic planning techniques, we hope to equip users with a powerful way to program the robots for assembly tasks. This automated planning workflow offloads tedious human programming work that is required by existing planning methods, and can generate programs that are guaranteed to work both in terms of collision and structural constraints in a short amount of time. The proposed planning strategies can be readily deployed to power robotic assembly of bespoke building components in a pre-fabrication factory setting where the uncertainty of the environment is low. In on-site robotic construction settings, assembly plans generated by these long-horizon planning algorithms are also valuable as baseline plans, upon which the adaptive, feedback-driven sensing and control strategies can make local modifications to increase the robustness of the execution.

On the design side, planning algorithms can be used to open up a new type of performance-driven design. When augmented by algorithmic planning, robotic construction provides additional benefits of being a predictable construction method that we can accurately evaluate and predict the feasibility, time and resources. This means that designers can use these measures as guiding principles in the early design stage and make critical design decisions to address potential issues and optimize performance that is related to construction.

7.3 Limitations and future work

Moving beyond the contributions of this thesis, this section summarizes the limitations of current work and highlights promising future directions.

7.3.1 Robust robotic assembly

Robustness of the technology is one of the most important aspects of any technology transfer from academia to industry. In all the real-world robotic verifications presented in this thesis, the planning is performed offline before execution, and the plan is executed in an open-loop fashion. The success of the plan relies on the assumption that the world goes exactly "as planned", and all the physical modeling (geometry or material-wise) is precise (see Section 1.3 for details on the assumptions). However, uncertain factors like fabrication errors and imprecise material modeling are inevitable in the real world and can cause a mismatch between the simulation and reality (for example, as in Section 4.5.1.1). Thus, equipping the robotic system with the ability to self-correct in a closed loop fashion is the key to increasing the robustness of the automated assembly process. Much progress has been made towards short-horizon, robust planning and control that emphasizes system dynamics [Gif18] and sensorial tracking and estimation [San18]. As we have discussed in Section 1.3, we envision that the long-horizon construction planning approach presented can be used as a blueprint-like baseline upon which the close-loop sensing and control strategies can make online adjustments. When the deviation from the baseline plan is too big, a new long-term plan can be recomputed, adjusting to the sensed information. Such an online replanning strategy has been demonstrated in a household preparation setting in the TAMP research [GPL+20]. Looking at the R&D history of now nearly mature robotic solutions for piece picking and palletizing[1], a demonstration of such integration on offline planning and online vision and control system is a crucial step toward a safe and robust industrial deployment of robotic assembly.

The connection detail between elements is a crucial aspect of assembly design that has a huge impact on the success of the automated assembly process. In Chapter 4, the connections are made by human operators. In Chapter 5, we demonstrate using a robotic arm to cooperate with multiple robotic clamps for automated timber assembly with integral joints. An important next step is to demonstrate robotic joining strategies that support a wider variety of materials and corresponding suitable joint types, for example, using robots for welding steel elements and for screwing and connecting plates for timber elements. Existing works provide in-depth investigations in specific robotic joining techniques, e.g. in-place connection details by robotic wire and arc additive manufacturing [Ari22], but their integration with an assembly process has not been demonstrated. Achieving such integration requires support from both advanced vision systems to accurately locate the points

[1]See, for example, the automated solutions provided by companies like XYZ Robotics(https://en.xyzrobotics.ai/) and Mujin (https://mujin-corp.com/).

for operations and adaptations in the planning algorithms to support multi-robot coordination. If the assembly robot and the joining robot move sequentially, the formalization and solving techniques proposed in Chapter 5 can be applied to model and solve such planning problems. An even more optimized process would involve the two robots working simultaneously, but the involved multi-agent path planning problems are much harder to solve, which are subject to active investigations [HOD+21, CLH+22].

7.3.2 Robotic assembly with dynamic scaffoldings

In this thesis, we have mostly investigated assembly without scaffolding elements, with the exception in Chapter 5 where designers manually plan out where and when to add supporting elements. This means that we are relying on the planning algorithm to find a good assembly sequence so that the partial construction can support itself and thus eliminate any waste related to the scaffolding. However, allowing the introduction of a few scaffolding elements might dramatically stabilize the partial construction and thus increase the robustness of the process. The inclusion of scaffolding in the process poses two new planning challenges. First, once added to the structure, scaffolding elements become additional collision objects in the workspace, which further crowds the already tight collision-free space for the robot to maneuver. Thus, a good assembly strategy would be to have a dynamic scaffolding system that is introduced in precise steps in the assembly sequence when needed and removed when the partial construction can stabilize itself. This strategy is similar in spirit to the cable-assisted, self-supporting construction of masonry structures proposed in [DPW+14]. Second, since the number and poses of scaffolding elements are not known a priori, the planning horizon, i.e. the total number of actions to be performed, is unbounded and the branching factor in the search is infinite because the scaffolding elements' poses are continuous variables. As argued throughout this thesis, while existing TAMP systems have displayed abilities to solve small problems with such characteristics, adaptions will be needed to specialize these systems so that they can scale to the complexity involved in construction domains.

7.3.3 Intelligent design feedback and optimization

In Chapter 6, we have demonstrated that our work in automated planning can be incorporated into a design evaluation workflow. However, due to the inherent complexity of finding a near-optimal construction sequence, the optimal constructability score evaluation of a design with 200 elements might take minutes to hours (especially when using different functions for the objective and constraint). The slowness of this evaluation means that including it in a formal optimization loop is technically not feasible at this moment. Future directions might include (1) accelerating the search routine while maintaining bounds for optimality and (2) richer feedback from the planner to the outer-loop optimizer to more efficiently guide the optimization.

First, as the number of elements involved in the structure grows, an element-by-element search strategy will not suffice at some points due to the complexity of the problem. A divide-and-conquer search strategy can be useful, where the structure is first decomposed into groups and then the planner can search in multiple hierarchies. Searching in decomposed groups can be seen as imposing a partial ordering constraint on the resulting sequence. This idea has been explored in [DPW+14, HZH+16], but it remains an open question on how to devise a decomposition algorithm that is complete for both structural and collision constraints. In addition, finding an optimal sequence is crucial for having a reliable optimal constructability measure. Devising machinery to estimate how far a given sequence is away from optimality is an important, yet challenging and unexplored direction. A potential direction is to investigate a continuous relaxed formulation of the combinatorial problem, which can be used to compute a lower bound of the optimum of the original problem. These relaxed formulations might lead to convex problems that can be solved to the optimality using convex optimization machinery, see, for example, SDP relaxation for many classic combinatorial optimization problems [Goe97].

Second, when computing the constructability measure, the planner can return much richer information to the optimizer instead of a single number for the performance score. This extended information includes, for example, the partial structure when the planner first backtracks and the multiple suboptimal sequences the planner has found when searching for the optimal one. Furthermore, due to the potential nonlinearity of the parametric design model and the inherent combinatorial nature of the sequencing problem, the objective landscape of the constructability score is unlikely to be continuous (let alone smooth). Thus, most of the gradient-based optimization algorithms will not be applicable for such problems. Designing specialized optimization algorithms that can take advantage of the extended information from the planner to more efficiently guide the design space exploration is a direction that has great practical relevance. Achieving this means that we arrive at smart planning tools that not only compute construction plans but can also provide design diagnosis and propose design alternatives.

7.4 Concluding remarks

Driven by the need to use resources more responsibly in the built environment, designers today are equipped with increasingly powerful computational means to design and explore high-performing structural solutions. Such design capabilities call for a new construction paradigm that can produce complex geometries precisely, quickly and cheaply. Construction robotics has the potential to deliver such promises, but it is important to have an efficient, automated way to offload the most labor-intensive step in this design-to-fabrication process, the programming of robots to achieve the construction tasks.

This thesis shows that algorithmic planning techniques can eliminate unnecessary human programming labor, generate feasible assembly plans, and stay versatile for various construction tasks. We hope that these planning tools, along with the new principles of construction-driven design, can encourage wider adoption of efficient structures in practice, and thus pave our way toward a more sustainable future for the built environment.

Appendix A

Theoretical results

In this appendix chapter, we state and prove the theoretical claims made in Chapter 3[1].

A.1 Regression polynomial complexity

First, we analyze the complexity of REGRESSION for *geometry-only* extrusion problems (section 3.7.1). Note that it is possible to achieve a better complexity of $\mathcal{O}(|T||E|)$ using an algorithm that caches collisions.

Theorem 1. *REGRESSION will solve any feasible geometry-only extrusion problem in polynomial time.*

Proof. Each colliding pair $\neg\textsc{Safe}(\tau_e, \{e'\})$ induces a partial-ordering constraint that element e' must be extruded after element e in order to safely execute trajectory τ_e. By equation 3, removing element e' weakly decreases the size of the set of partial-order constraints for each trajectory τ_e. Because we assume feasibility, there exists a total ordering ψ of E and a corresponding sequence of trajectories π from T that respect collision constraints. As a result, for every set of unprinted elements $P' \subseteq E$, the element $e' = \psi[i] \in P'$ that has the largest index $i = \max_{\psi[j] \in P'}(j)$ in ψ is guaranteed to have a safe trajectory $\tau_e \in T$. Each of the $|E|$ iterations requires considering

[1] A version of this chapter has been published in [GHLPM20].

at most $|T|$ trajectories and checking collisions with at most $|E|$ elements. As a result, the complexity of REGRESSION is $\mathcal{O}(|T||E|^2)$. □

A.2 Probabilistic completeness

Because TAMP is decidable [DKLP16], extrusion planning is also decidable, meaning that there exists *complete* algorithms that can correctly prove a problem is either feasible or infeasible. However, because we use randomized sampling-based strategies, we instead prove the weaker claim that our algorithms are probabilistically complete. First, we build on our problem formulation in section 3.4.2. by identifying a class of *robustly feasible* [KF11, GLPK18b] extrusion problems, problems that admit a non-degenerate set of solutions making them amenable to sampling-based planning. Define $\chi(\tau, P)$ to be the *clearance* of trajectory τ [KKL98] with respect to printed elements P as the greatest lower bound on the distance from any configuration on τ to the boundary of the currently collision-free configuration space $\partial Q(P)$:

$$\chi(\tau, P) = \text{inf}_{\lambda \in [0,1]} \text{inf}_{q \in \partial Q(P)} ||\tau(\lambda) - q||. \tag{A.1}$$

Let $\mu(X; \mathcal{X})$ be a measure on subsets $X \subseteq \mathcal{X}$ such that $0 < \mu(\mathcal{X}; \mathcal{X}) < \infty$. Let $X \subseteq_{\emptyset} \mathcal{X} \implies [\emptyset \neq X \subseteq \mathcal{X}] \wedge [\mu(X; \mathcal{X}) > 0]$ denote that X is a nonempty subset of $\mathcal{X}$ with positive measure with respect to $\mathcal{X}$.

Definition 4. An extrusion problem $\Pi = \langle N, G, E, \mathcal{Q}, q_0 \rangle$ is *robustly feasible* for a valid extrusion sequence $\vec{\psi} = [\vec{e}_1, \vec{e}_2, ..., \vec{e}_m]$ (definition 1) if there exists sequence of extrusion mode coparameter sets $[\Sigma_{\vec{e}_1}, ..., \Sigma_{\vec{e}_m}]$ s.t.

$$\forall i \in \{1, ..., m\}. \Sigma_{\vec{e}_i} \subseteq_{\emptyset} X_o(\vec{e}_i) \tag{A.2}$$

and $\forall \vec{\sigma} = [\sigma_{\vec{e}_1}, ..., \sigma_{\vec{e}_m}] \in \bigotimes_{i=1}^{m} \Sigma_{\vec{e}_i}$. exists:

- a sequence *start* and *end* extrusion configuration sets $[T_{\sigma_{\vec{e}_1}}, ..., T_{\sigma_{\vec{e}_m}}]$ and $[T'_{\sigma_{\vec{e}_1}}, ..., T'_{\sigma_{\vec{e}_m}}]$

s.t.

$$\forall i \in \{1, ..., m\}. T_{\sigma_{\vec{e}_i}} \subseteq_{\emptyset} \mathcal{T}(\alpha, \sigma_{\vec{e}_i}) \tag{A.3}$$

$$T'_{\sigma_{\vec{e}_i}} \subseteq_{\emptyset} \mathcal{T}(\sigma_{\vec{e}_i}, \alpha) \tag{A.4}$$

and $\forall [q_{\sigma_{\vec{e}_1}}, ..., q_{\sigma_{\vec{e}_m}}] \in \bigotimes_{i=1}^{m} T_{\sigma_{\vec{e}_i}}$
and $\forall [q'_{\sigma_{\vec{e}_1}}, ..., q'_{\sigma_{\vec{e}_m}}] \in \bigotimes_{i=1}^{m} T'_{\sigma_{\vec{e}_i}}$. exists:

- a *solution* (definition 3) comprised of $2m+1$ trajectories $\pi = [\tau_{t_1}, \tau_{\vec{e}_1}, ..., \tau_{t_{m+1}}]$ s.t.

$$\forall i \in \{1, ...,m\}\ .\tau_{\vec{e}_i}(0) = q_{\sigma_{\vec{e}_i}}, \tau_{\vec{e}_i}(1) = q'_{\sigma_{\vec{e}_i}} \tag{A.5}$$

$$\chi(\tau_{t_i}, \psi_{1:i-1}), \chi(\tau_{\vec{e}_i}, \psi_{1:i-1}) > 0 \tag{A.6}$$

and $\chi(\tau_{t_{m+1}}, E) > 0$.

Breaking down the definition, equation A.2 requires the mode set $\Sigma_{\vec{e}_i}$ for each extrusion to have positive measure with respect to the mode space for $\vec{e}_i$. Equations A.3 and A.4 states that for each mode $\sigma_{\vec{e}_i} \in \Sigma_{\vec{e}_i}$, the set of transition configurations $T_{\sigma_{\vec{e}_i}}$ from $\alpha \rightarrow \sigma_{\vec{e}_i}$ and the set of transition configurations $T'_{\sigma_{\vec{e}_i}}$ from $\sigma_{\vec{e}_i} \rightarrow \alpha$ both have positive measure relative to their respective spaces. Finally, equation A.6 states that there exists solutions π where the transit trajectory τ_{t_i} between the pair of transition configurations $q'_{\sigma_{\vec{e}_{i-1}}}, q_{\sigma_{\vec{e}_i}}$ for transit mode α has positive clearance and the extrusion trajectory $\tau_{\vec{e}}$ between each pair of transition configurations $q_{\sigma_{\vec{e}_i}}, q'_{\sigma_{\vec{e}_i}}$ for extrusion mode $\sigma_{\vec{e}_i}$ has positive clearance. As a result, the motion planning problem $q'_{\sigma_{\vec{e}_{i-1}}} \rightarrow q_{\sigma_{\vec{e}_i}}$ and the constrained motion planning problem $q_{\sigma_{\vec{e}_i}} \rightarrow q'_{\sigma_{\vec{e}_i}}$ subject to manifold $\mathcal{M}(\sigma_{\vec{e}_i})$ are both robustly feasible.

We assume that PlanMotion is a probabilistically complete motion planner and PlanConstrained is a problematically complete constrained motion planner. Assume that SampleOrientation($\vec{e}$) randomly samples $X_o(\vec{e}_i)$ independently with probability density bounded away from zero and SampleIK(p, x_o) randomly samples the $(d-5)$-dimensional space of kinematic solutions independently with probability density also bounded away from zero. As a result, SampleIK can be used to sample both $\mathcal{T}(\alpha, \sigma_{\vec{e}_i})$ and $\mathcal{T}(\sigma_{\vec{e}_i}, \alpha)$ when $x_o = \sigma_{\vec{e}_i}$.

Theorem 2. *Progression is probabilistically complete for robustly-feasible extrusion problems.*

Proof. We consider a sequence of m events where each event involves both SampleExtrusion and PlanMotion succeeding given the set of solutions described in definition 4. Because Progression is *persistent* (section 3.5.4), each search node will be revisited in a finite amount of time. As a result, we can ignore the computation in between each revisit. For the ith event in the sequence, SampleOrientation has positive probability of sampling a mode coparameter $\sigma_{\vec{e}_i} \in \Sigma_{\vec{e}_i}$. Likewise, SampleIK has positive probability of sampling transition configurations $q_{\sigma_{\vec{e}_i}} \in T_{\sigma_{\vec{e}_i}}$ and $q'_{\sigma_{\vec{e}_i}} \in T'_{\sigma_{\vec{e}_i}}$. Because PlanConstrained and PlanMotion are probabilistically complete, for i sufficiently large the probability that they identify a solution is positive. As a result, for i sufficiently large, the probability that both SampleExtrusion and PlanMotion succeed on a given attempt, satisfying the ith event, is also positive. Thus, the ith event will succeed in a finite number of reattempts with probability one, and all m events will succeed in a finite amount of time with probability one. □

Theorem 3. *Regression is probabilistically complete for robustly-feasible extrusion problems.*

Proof. We trivially apply the argument in theorem 2 but in the reverse direction from $i \in \{m, ..., 1\}$. □

Appendix B

Simulated extrusion benchmarks

In this appendix chapter, we present the benchmarking results for extrusion planning in Chapter 3[1].

Figures B-1, B-2, and B-3 display the extrusion problems that we considered. For each problem, we ran one trial of REGRESSION+*StiffPlan* and recorded the extrusion sequence it produced. For successful trials, elements are colored by their index in a extrusion sequence, where purple elements are printed first and red elements are printed last. All elements in the structure are black an unsuccessful trial. Some problems are the result of a linear transformation, such as a rotation or scaling, applied to the same original frame structure. Other problems are discretized version of the same object but with varying degrees of topological complexity. All the tested problem instances are available at `https://github.com/yijiangh/assembly_instances`.

[1]A version of this chapter has been published in [GHLPM20].

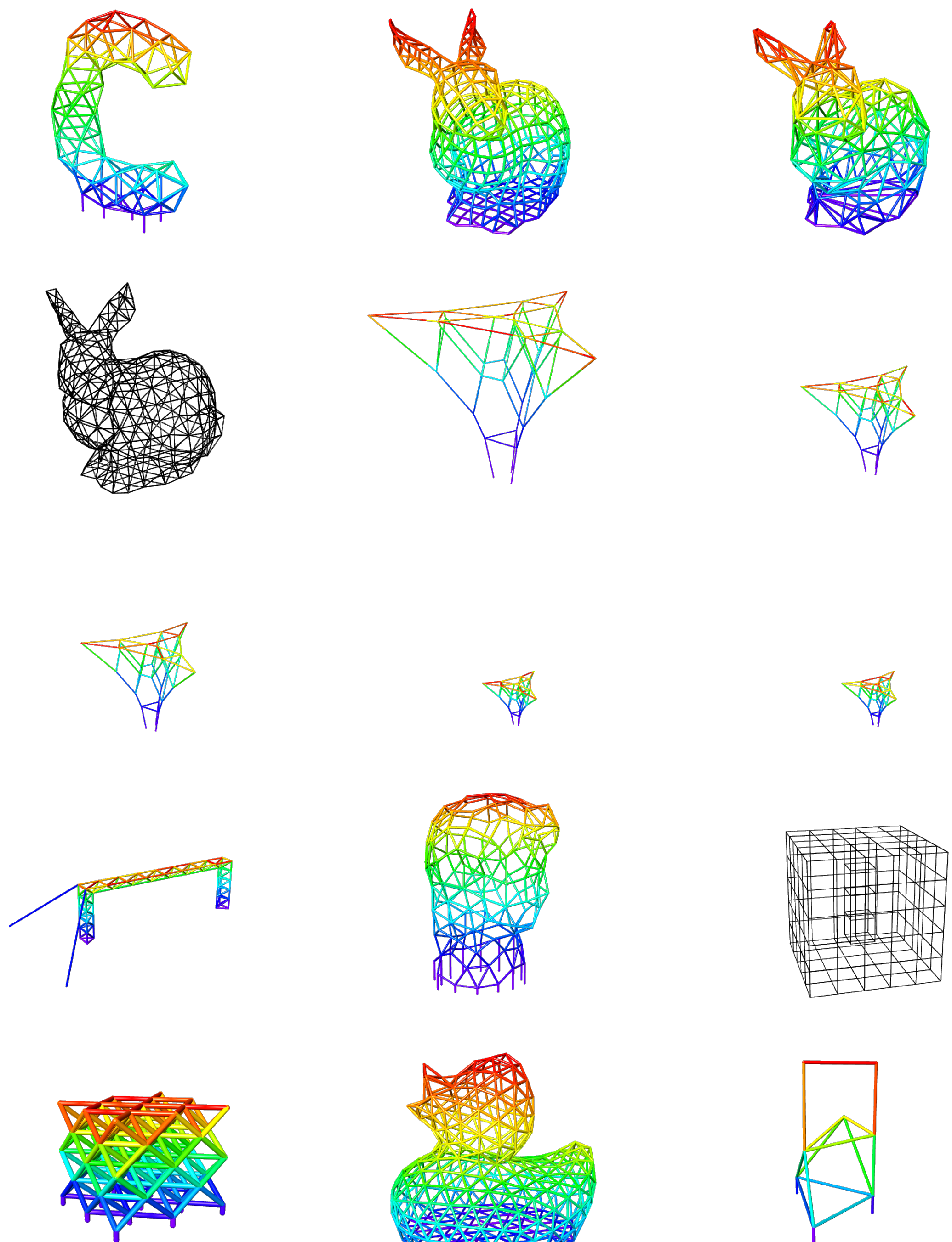


Figure B-1: Extrusion Problems - 1. For successful trials, elements are colored by their index in a extrusion sequence, where purple elements are printed first and red elements are printed last. All elements in the structure are black an unsuccessful trial.

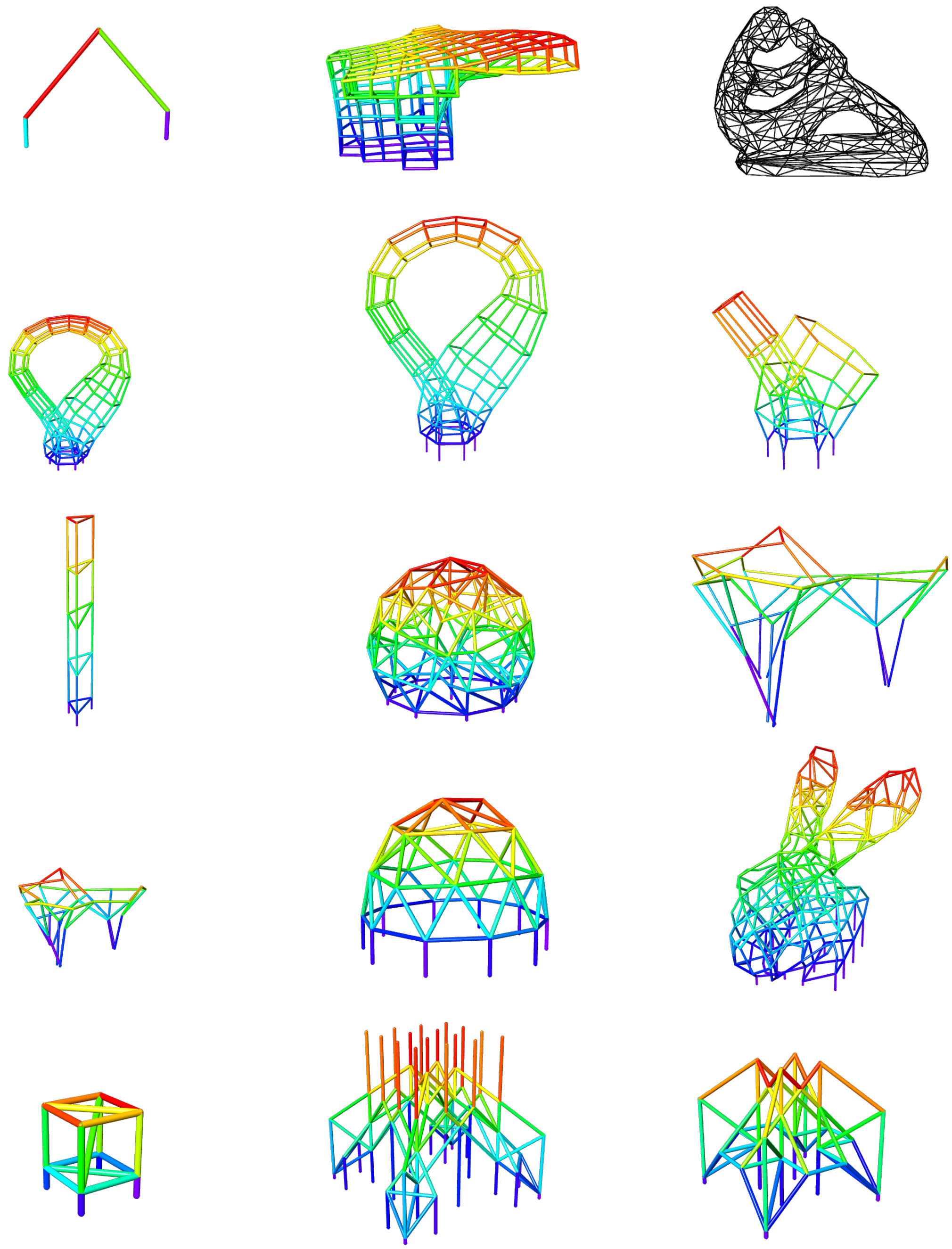


Figure B-2: Extrusion Problems - 2. For successful trials, elements are colored by their index in a extrusion sequence, where purple elements are printed first and red elements are printed last. All elements in the structure are black an unsuccessful trial.

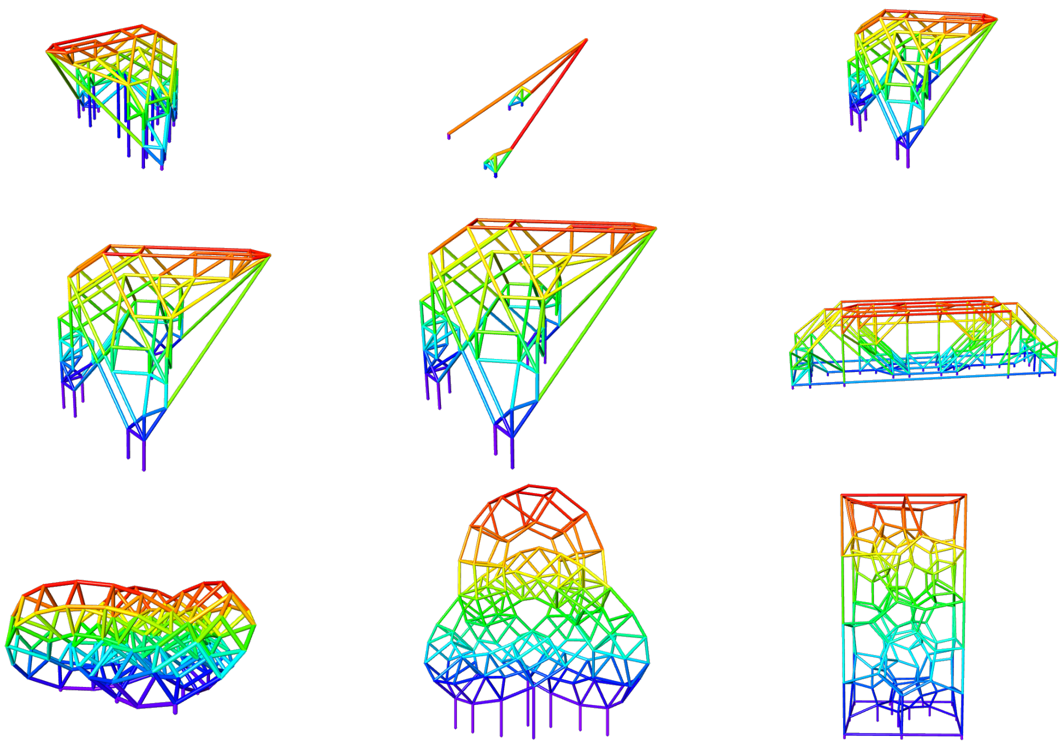


Figure B-3: Extrusion Problems- 3. For successful trials, elements are colored by their index in a extrusion sequence, where purple elements are printed first and red elements are printed last. All elements in the structure are black an unsuccessful trial.

Appendix C

Optimal construction sequencing benchmarks

In this appendix chapter, we provide the detailed data for the double-layer roof design used in Section 6.5 for the convenience of future benchmarking purposes. The material property of the cellulose used in the design is given in Table C.1. The node geometry is given in Table C.2 and the element cross-sections, node connectivity, and the optimal sequence (Figure 6-15) is given in Table C.3. In addition, Figure 6-15, Figure C-1 and Figure C-2 present detailed, step-by-step illustration for the optimal sequences computed for the original design, variation 1, and variation 2 in Section 6.5.3.

Table C.1: Material property of cellulose.

Young's modulus E (kN/cm2)	In-plane shear modulus G12 (kN/cm2)	Transverse shear modulus G3 (kN/cm2)	weight (kN/m3)	Tensile strength (kN/cm2)	Compressive strength (kN/cm2)
207	82.8	82.8	12.7	4.34	4.34

Table C.2: The node position data of the double-layer roof design in Figure 6-6. Node #59, 60, 61 are grounded.

Node indices	X (m)	Y (m)	Z (m)
0	0.253	-0.259	-0.133
1	0.446	-0.119	-0.381
2	0.472	-0.372	-0.133
3	0.705	-0.267	-0.133
4	0.643	0.141	-0.133
5	0.125	-0.035	-0.133
6	0.380	0.212	-0.133
7	0.267	0.467	-0.133
8	0.497	0.448	-0.381
9	0.681	0.625	-0.133
10	0.436	0.702	-0.133
11	0.863	0.446	-0.133
12	-0.285	0.272	-0.133
13	-0.081	0.413	-0.381
14	0.107	0.229	-0.133
15	-0.069	0.133	-0.133
16	-0.391	0.503	-0.133
17	-0.079	0.738	-0.133
18	0.092	-0.376	0.000
19	0.299	-0.543	0.000
20	0.564	-0.557	0.000
21	0.806	-0.440	0.000
22	0.859	-0.065	-0.133
23	0.992	-0.248	0.000

Continued on next page

Table C.2 – continued from previous page

Node indices	X (m)	Y (m)	Z (m)
24	1.078	0.006	0.000
25	0.893	0.191	-0.133
26	1.090	0.275	0.000
27	1.039	0.538	0.000
28	0.863	0.740	0.000
29	0.627	0.867	0.000
30	0.361	0.906	0.000
31	0.178	0.712	-0.133
32	0.091	0.916	0.000
33	-0.176	0.955	0.000
34	-0.331	0.747	-0.133
35	-0.437	0.917	0.000
36	-0.577	0.693	0.000
37	-0.579	0.425	0.000
38	-0.468	0.181	0.000
39	-0.262	0.012	0.000
40	-0.043	-0.142	0.000
41	0.654	0.031	0.000
42	-0.080	0.516	-0.133
43	0.057	0.510	0.000
44	0.595	0.716	0.000
45	0.225	-0.021	0.000
46	0.358	-0.032	-0.133
47	-0.064	0.174	0.000
48	0.649	0.372	-0.133

Continued on next page

Table C.2 – continued from previous page

Node indices	X (m)	Y (m)	Z (m)
49	0.703	0.360	0.000
50	-0.265	0.729	0.000
51	0.357	0.361	0.000
52	0.332	-0.293	0.000
53	0.802	-0.207	0.000
54	-0.203	0.477	0.000
55	0.276	0.736	0.000
56	0.026	0.783	0.000
57	0.144	0.192	0.000
58	0.459	-0.045	0.000
59	0.446	-0.119	-0.769
60	0.497	0.448	-0.769
61	-0.081	0.413	-0.769

Table C.3: The element cross-section, length, and node connectivity data of the double-layer roof design in Figure 6-6. All tubes have a thickness of 0.16 cm. The node id of each element maps in the list of nodes listed in Table C.2. The optimal sequence shown in Figure 6-15 is listed in the last column, where each entry corresponds to the element's order in the sequence. For example, in the first row, element #0 is the 3rd element to be built in the optimal sequence.

Element indices	diameter (cm)	length (m)	node 1 id	node 2 id	optimal sequence
0	0.8	0.344	0	1	3
1	0.6	0.355	1	2	20
2	1.0	0.388	1	3	9
3	0.8	0.410	1	4	13

Continued on next page

Table C.3 – continued from previous page

Element indices	diameter (cm)	length (m)	node 1 id	node 2 id	optimal sequence
4	1.0	0.415	1	5	22
5	0.6	0.419	1	6	7
6	0.8	0.339	7	8	4
7	0.6	0.355	8	9	37
8	1.6	0.360	8	10	8
9	0.8	0.363	6	8	6
10	1.6	0.422	4	8	16
11	1.0	0.442	8	11	41
12	0.6	0.351	12	13	46
13	0.6	0.361	13	14	32
14	1.6	0.375	13	15	34
15	0.6	0.408	13	16	63
16	1.6	0.409	13	17	35
17	0.6	0.431	7	13	5
18	0.6	0.239	0	18	69
19	1.0	0.367	5	18	70
20	0.6	0.276	2	19	72
21	0.6	0.317	0	19	71
22	0.6	0.245	2	20	73
23	0.8	0.348	3	20	74
24	0.6	0.240	3	21	75
25	0.6	0.262	22	23	155
26	0.6	0.316	3	23	158
27	0.8	0.266	22	24	152
28	0.6	0.293	24	25	151

Continued on next page

Table C.3 – continued from previous page

Element indices	diameter (cm)	length (m)	node 1 id	node 2 id	optimal sequence
29	0.6	0.252	25	26	140
30	0.6	0.314	11	26	139
31	0.6	0.239	11	27	76
32	0.6	0.254	9	28	77
33	0.6	0.323	11	28	78
34	0.6	0.281	9	29	79
35	0.8	0.284	10	29	80
36	0.6	0.255	10	30	81
37	0.6	0.298	30	31	120
38	0.6	0.259	31	32	123
39	0.6	0.280	17	32	109
40	0.8	0.272	17	33	110
41	0.6	0.292	33	34	111
42	0.6	0.240	34	35	147
43	0.8	0.422	17	35	166
44	0.6	0.285	34	36	153
45	0.8	0.297	16	36	148
46	0.6	0.243	16	37	112
47	0.8	0.357	12	37	113
48	0.6	0.244	12	38	82
49	0.8	0.264	15	39	84
50	0.6	0.293	12	39	83
51	0.6	0.239	5	40	85
52	0.8	0.306	15	40	86
53	0.6	0.173	4	41	87

Continued on next page

Table C.3 – continued from previous page

Element indices	diameter (cm)	length (m)	node 1 id	node 2 id	optimal sequence
54	0.6	0.262	22	41	114
55	0.6	0.191	42	43	89
56	0.8	0.252	7	43	88
57	0.6	0.182	9	44	90
58	0.6	0.207	10	44	91
59	0.6	0.167	5	45	92
60	0.6	0.188	45	46	93
61	0.6	0.139	15	47	95
62	0.6	0.223	14	47	94
63	0.6	0.144	48	49	97
64	0.6	0.225	11	49	96
65	0.6	0.149	34	50	116
66	0.6	0.229	17	50	115
67	0.6	0.191	7	51	99
68	0.6	0.201	6	51	98
69	0.6	0.158	0	52	100
70	0.6	0.208	2	52	101
71	0.6	0.175	3	53	102
72	0.6	0.203	22	53	117
73	0.6	0.185	42	54	103
74	0.6	0.232	16	54	118
75	0.6	0.167	31	55	121
76	0.6	0.211	10	55	104
77	0.6	0.175	17	56	119
78	0.6	0.214	31	56	126

Continued on next page

Table C.3 – continued from previous page

Element indices	diameter (cm)	length (m)	node 1 id	node 2 id	optimal sequence
79	0.6	0.143	14	57	105
80	0.6	0.257	15	57	106
81	0.6	0.167	46	58	108
82	0.6	0.293	4	58	107
83	0.6	0.267	18	19	127
84	0.6	0.269	18	40	122
85	0.6	0.265	19	20	129
86	0.6	0.268	20	21	131
87	0.6	0.268	21	23	159
88	0.6	0.268	23	24	157
89	0.6	0.269	24	26	150
90	0.6	0.268	26	27	138
91	0.6	0.268	27	28	137
92	0.6	0.268	28	29	143
93	0.6	0.269	29	30	125
94	0.6	0.270	30	32	133
95	0.6	0.270	32	33	146
96	0.6	0.263	33	35	165
97	0.6	0.264	35	36	169
98	0.6	0.267	36	37	171
99	0.6	0.268	37	38	174
100	0.6	0.268	38	39	176
101	0.6	0.267	39	40	179
102	0.6	0.379	18	45	124
103	0.6	0.254	18	52	128

Continued on next page

Table C.3 – continued from previous page

Element indices	diameter (cm)	length (m)	node 1 id	node 2 id	optimal sequence
104	0.6	0.251	19	52	130
105	0.6	0.351	20	52	132
106	0.6	0.423	20	53	154
107	0.6	0.233	21	53	134
108	0.6	0.194	23	53	160
109	0.6	0.425	24	41	156
110	0.6	0.349	24	53	161
111	0.6	0.500	26	41	141
112	0.6	0.397	26	49	142
113	0.6	0.381	27	49	135
114	0.6	0.270	28	44	144
115	0.6	0.413	28	49	149
116	0.6	0.154	29	44	136
117	0.6	0.302	30	44	145
118	0.6	0.190	30	55	162
119	0.6	0.258	32	55	163
120	0.6	0.148	32	56	164
121	0.6	0.243	33	50	168
122	0.6	0.265	33	56	167
123	0.6	0.255	35	50	170
124	0.6	0.315	36	50	172
125	0.6	0.432	36	54	173
126	0.6	0.380	37	54	175
127	0.6	0.404	38	47	177
128	0.6	0.397	38	54	178

Continued on next page

Table C.3 – continued from previous page

Element indices	diameter (cm)	length (m)	node 1 id	node 2 id	optimal sequence
129	0.6	0.256	39	47	180
130	0.6	0.294	40	45	181
131	0.6	0.317	40	47	182
132	0.6	0.333	41	49	183
133	0.6	0.445	41	51	184
134	0.6	0.280	41	53	185
135	0.6	0.209	41	58	186
136	0.6	0.357	43	47	187
137	0.6	0.334	43	51	188
138	0.6	0.262	43	54	189
139	0.6	0.315	43	55	190
140	0.6	0.275	43	56	191
141	0.6	0.329	43	57	192
142	0.6	0.372	44	49	193
143	0.6	0.427	44	51	194
144	0.6	0.319	44	55	195
145	0.6	0.349	45	47	196
146	0.6	0.404	45	51	197
147	0.6	0.293	45	52	198
148	0.6	0.228	45	57	199
149	0.6	0.235	45	58	200
150	0.6	0.333	47	54	201
151	0.6	0.209	47	57	202
152	0.6	0.346	49	51	203
153	0.6	0.260	50	54	204

Continued on next page

Table C.3 – continued from previous page

Element indices	diameter (cm)	length (m)	node 1 id	node 2 id	optimal sequence
154	0.6	0.296	50	56	205
155	0.6	0.384	51	55	206
156	0.6	0.272	51	57	207
157	0.6	0.420	51	58	208
158	0.6	0.478	52	53	209
159	0.6	0.279	52	58	210
160	0.6	0.379	53	58	211
161	0.6	0.383	54	56	212
162	0.6	0.255	55	56	213
163	0.6	0.246	0	2	19
164	0.6	0.258	0	5	23
165	0.6	0.256	2	3	21
166	0.8	0.254	3	22	51
167	0.6	0.258	22	25	54
168	0.6	0.257	11	25	49
169	0.6	0.255	9	11	40
170	0.6	0.256	9	10	36
171	0.8	0.258	10	31	52
172	0.6	0.258	17	31	56
173	0.6	0.253	17	34	65
174	0.6	0.250	16	34	67
175	0.6	0.255	12	16	62
176	0.8	0.257	12	15	45
177	1.0	0.256	5	15	44
178	0.6	0.250	0	46	15

Continued on next page

Table C.3 – continued from previous page

Element indices	diameter (cm)	length (m)	node 1 id	node 2 id	optimal sequence
179	0.6	0.359	2	46	24
180	0.6	0.413	3	4	18
181	0.6	0.419	3	46	17
182	1.0	0.297	4	22	53
183	0.6	0.255	4	25	48
184	0.6	0.304	25	48	50
185	0.6	0.226	11	48	42
186	0.6	0.255	9	48	38
187	0.6	0.289	7	10	11
188	0.6	0.392	10	48	39
189	0.6	0.261	7	31	55
190	0.8	0.324	31	42	66
191	0.6	0.222	17	42	57
192	0.6	0.342	34	42	68
193	0.6	0.312	16	42	64
194	0.6	0.319	12	42	59
195	0.6	0.383	15	42	60
196	0.6	0.200	14	15	47
197	0.6	0.233	5	46	28
198	0.6	0.265	5	14	33
199	0.6	0.355	5	6	27
200	0.6	0.333	4	46	25
201	0.8	0.273	4	6	12
202	0.6	0.232	4	48	26
203	0.6	0.350	7	42	58

Continued on next page

Table C.3 – continued from previous page

Element indices	diameter (cm)	length (m)	node 1 id	node 2 id	optimal sequence
204	0.6	0.287	7	14	31
205	0.6	0.279	6	7	10
206	1.0	0.393	7	48	30
207	0.6	0.245	6	46	14
208	0.6	0.341	14	42	61
209	0.8	0.274	6	14	43
210	0.6	0.313	6	48	29
211	1.6	0.388	1	59	0
212	1.6	0.388	8	60	1
213	1.6	0.388	13	61	2

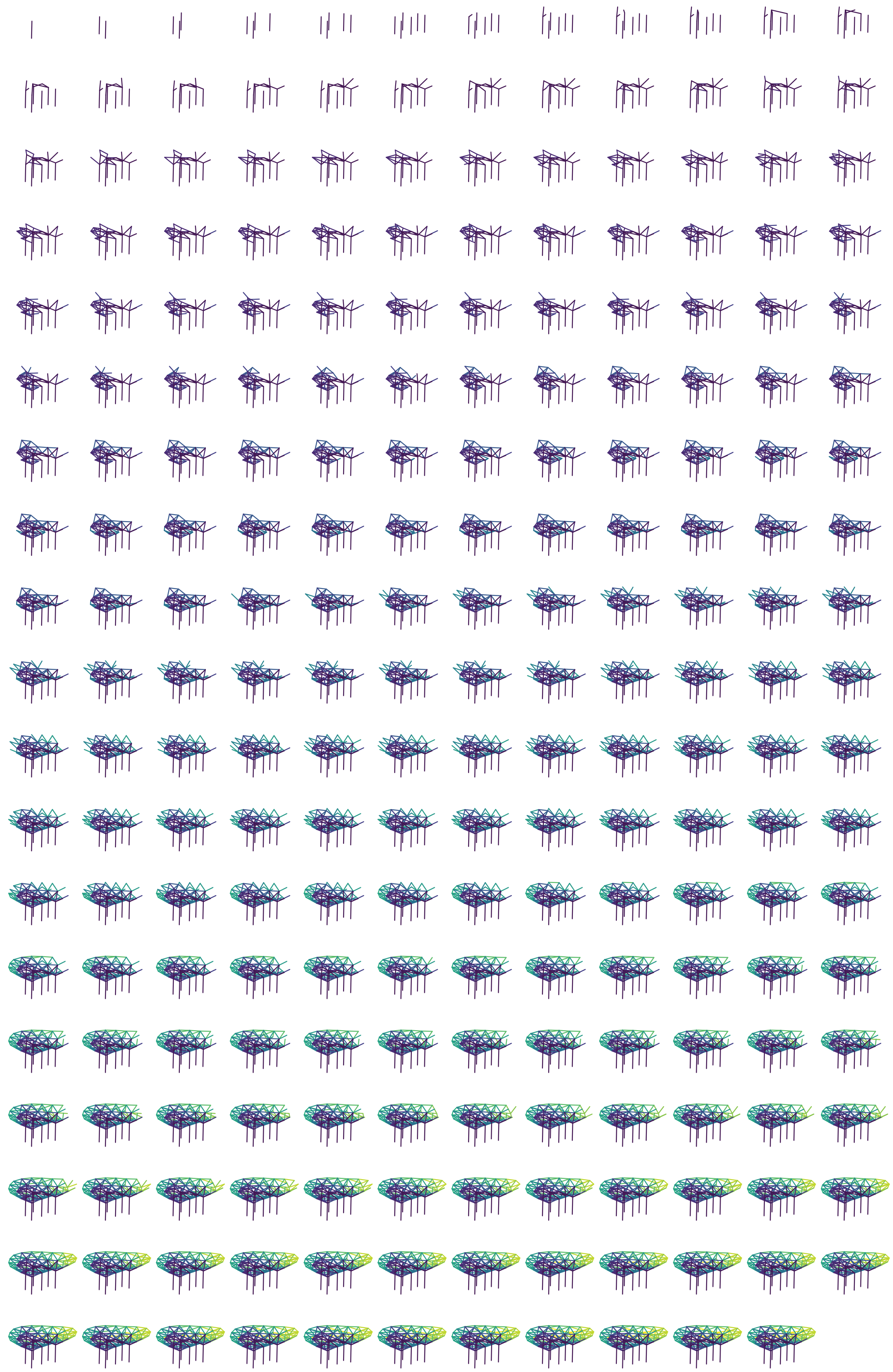


Figure C-1: Optimal sequence for design variation 1 (see Figure 6-14), found by the iterative feasible search algorithm.

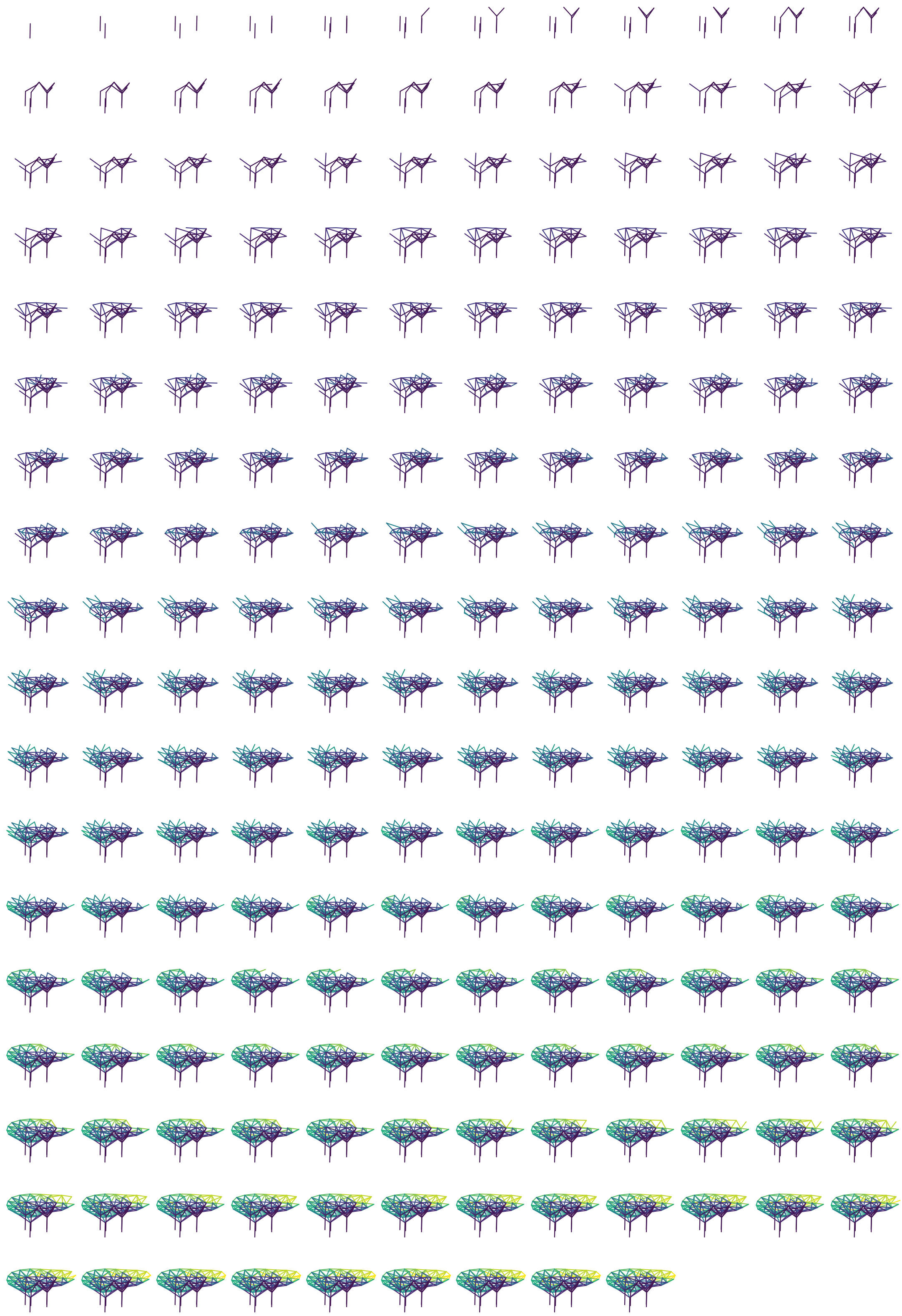


Figure C-2: Optimal sequence for design variation 2 (see Figure 6-14), found by the iterative feasible search algorithm.

Bibliography

[AB01] S. M. L Adriaenssens and M. R Barnes. Tensegrity spline beam and grid shell structures. *Engineering Structures*, 23(1):29–36, January 2001.

[AB17] Stefano Andreani and Martin Bechthold. [r]evolving brick: geometry and performance innovation in ceramic building systems through design robotics. In *Fabricate 2014*, Negotiating Design & Making, page 182–191. UCL Press, dgo-digital original edition, 2017.

[ABCC11] David L Applegate, Robert E Bixby, Vašek Chvátal, and William J Cook. The traveling salesman problem. In *The Traveling Salesman Problem*. Princeton university press, 2011.

[ABVW14] Sigrid Adriaenssens, Philippe Block, Diederik Veenendaal, and Chris Williams. *Shell structures for architecture: form finding and optimization*. Routledge, 2014.

[ABY+20] Ana Anton, Patrick Bedarf, Angela Yoo, Benjamin Dillenburger, Lex Reiter, Timothy Wangler, and Robert J. Flatt. Concrete choreography: prefabrication of 3d-printed columns. In *Fabricate 2020*, Making Resilient Architecture, page 286–293. UCL Press, 2020.

[ACP+14] Giuseppe Alemanno, Paolo Cignoni, Nico Pietroni, Federico Ponchio, and Roberto Scopigno. *Interlocking Pieces for Printing Tangible Cultural Heritage Replicas*. The Eurographics Association, 2014.

[AHM+20] Felix Amtsberg, Yijiang Huang, Daniel J.M. Marshall, Kevin Moreno Gata, and Caitlin Mueller. Structural upcycling: Matching digital and natural geometry. In *Proceedings of Advances in Architectural Geometry*, 2020.

[AJL+19] Rahul Arora, Alec Jacobson, Timothy R. Langlois, Yijiang Huang, Caitlin Mueller, Wojciech Matusik, Ariel Shamir, Karan Singh, and David I. W. Levin. Volumetric Michell trusses for parametric design & fabrication. In *Proceedings of the ACM Symposium on Computational Fabrication - SCF '19*, pages 1–13, Pittsburgh, Pennsylvania, 2019. ACM Press.

[AJNP16] David Andreen, Petra Jenning, Nils Napp, and Kirstin Petersen. Emergent structures assembled by large swarms of simple robots. In *Proceedings of the 36th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, 2016.

[AKGK16] Aleksandra Anna Apolinarska, Michael Knauss, Fabio Gramazio, and Matthias Kohler. The sequential roof. In *Advancing wood architecture*, page 45–59. Routledge, 2016.

[AL21] Johan Julius Petrus Aejmelaeus-Lindström. *Rock Printing. Robotic fabrication of jammed architectural structures from bulk materials*. Doctoral thesis, ETH Zurich, 2021. Accepted: 2021-07-26T08:48:03Z.

[ALS94] R Alami, J.-P. Laumond, and T Siméon. Two manipulation planning algorithms. In *Workshop on Algorithmic Foundations of Robotics (WAFR)*, 1994.

[AM83] Robert U. Ayres and Steven M Miller. *Robotics, applications and social implications*. Ballinger Pub. Co., Cambridge, Mass, 1983.

[AMG+18] Inés Ariza, Ammar Mirjan, Augusto Gandia, Gonzalo Casas, Samuel Cros, Fabio Gramazio, and Matthias Kohler. In place detailing: Combining 3d printing and robotic assembly. In *Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture*, page 312–321. Association for Computer Aided Design in Architecture (ACADIA), Oct 2018.

[AMS+20] Lidia Atanasova, Daniela Mitterberger, Timothy Sandy, Fabio Gramazio, Matthias Kohler, and Kathrin Dörfler. Design tool for open-ended collaborative assembly processes. In *Proceedings of the 40th Annual Conference of the Association of Computer Aided Design in Architecture (ACADIA)*, page 10, 2020.

[APH+03] Maneesh Agrawala, Doantam Phan, Julie Heiser, John Haymaker, Jeff Klingner, Pat Hanrahan, and Barbara Tversky. Designing effective step-by-step assembly instructions. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 828–837. ACM, 2003.

[APTP16] Felix Amtsberg, Gernot Parmann, Andreas Trummer, and Stefan Peters. From analysis to production and back attempts and results of reusable adaptive freeform production strategies for double curved concrete construction elements. In Dagmar Reinhardt, Rob Saunders, and Jane Burry, editors, *Robotic Fabrication in Architecture, Art and Design 2016*, page 304–315. Springer International Publishing, Cham, 2016.

[Ari22] Inés Ariza. *Adaptive Detailing: Design and Fabrication Methods for In Place Wire and Arc Additive Manufacturing Connection Details*. PhD thesis, ETH Zurich, Zurich, 2022.

[ASD+17] Inés Ariza, T. Shan Sutherland, James B. Durham, Caitlin T. Mueller, Wes Mcgee, and Brandon Clifford. Robotic fabrication of stone assembly details. In *Fabricate 2017*, page 106–113. UCL Press, 2017.

[ASL90] Rachid Alami, Thierry Siméon, and Jean-Paul Laumond. A geometrical approach to planning manipulation tasks. The case of discrete placements and grasps. In *International Symposium of Robotic Research (ISRR)*, 1990.

[Att15] Marco Attene. Shapes in a box: Disassembling 3D objects for efficient packing and fabrication. In *Computer Graphics Forum*, volume 34, pages 64–76. Wiley Online Library, 2015.

[AYB+19] Ana Anton, Angela Yoo, Patrick Bedarf, Lex Reiter, Timothy Wangler, and Benjamin Dillenburger. Vertical modulations: Computational design for concrete 3d printed columns. In *Proceedings of the 39st Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, October 2019.

[BA08] Carlos Balaguer and Mohamed Abderrahim. Trends in robotics and automation in construction. *Robotics and Automation in Construction*, Oct 2008.

[BA15] Patrick Beeson and Barrett Ames. Trac-ik: An open-source library for improved solving of generic inverse kinematics. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 928–935. IEEE, 2015.

[BA20] Hayley Bouza and Serdar Aşut. Advancing reed-based architecture through circular digital fabrication. In *Proceedings of eCAADe*, page 10, 2020.

[BAP22] Edvard P.G. Bruun, Sigrid Adriaenssens, and Stefana Parascho. Structural rigidity theory applied to the scaffold-free (dis)assembly of space frames using cooperative robotics. *Automation in Construction*, 141:104405, Sep 2022.

[Bar99] Michael R. Barnes. Form Finding and Analysis of Tension Structures by Dynamic Relaxation. *International Journal of Space Structures*, 14(2):89–104, June 1999.

[BB16] M V A Raju Bahubalendruni and Bibhuti Bhusan Biswal. A review on assembly sequence generation and its automation. *Proceedings* *of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 230(5):824–838, 2016.

[BBC11] Johannes Braumann and Sigrid Brell-Cokcan. Parametric robot control: integrated CAD/CAM for architectural design. In *Proceedings of the 31st Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, 2011.

[BBC12a] Johannes Braumann and Sigrid Brell-Cokcan. Digital and physical computing for industrial robots in architecture: Interfacing arduino with industrial robots. In *Proceedings of CAADRIA*, page 317–326, Chennai, India, 2012.

[BBC12b] Johannes Braumann and Sigrid Brell-Cokcan. Real-time robot simulation and control for architectural design. In *Proceedings of the 30th eCAADe Conference*, page 8, 2012.

[BBCK22] Richard Buswell, Ana Blanco, Sergio Cavalaro, and Peter Kinnell, editors. *Third RILEM International Conference on Concrete and Digital Fabrication: Digital Concrete 2022*, volume 37 of *RILEM Bookseries*. Springer International Publishing, Cham, 2022.

[BBH13] Tobias Bonwetsch, Ralph Bärtschi, and Matthias Helmreich. Brickdesign: A software for planning robotically controlled nonstandard brick assemblies. In Sigrid Brell-Çokcan and Johannes Braumann, editors, *Rob | Arch 2012*, page 102–109, Vienna, 2013. Springer.

[BBVM16] Giulio Brugnaro, Ehsan Baharlou, Lauren Vasey, and Achim Menges. Robotic softness: An adaptive robotic fabrication process for woven structures. In *Proceedings of ACADIA*, page 154–163, Ann Arbor (Michigan), USA, 2016.

[BBW15] Lukas Beyeler, Jean-Charles Bazin, and Emily Whiting. A graph-based approach for discovery of stable deconstruction sequences. In Philippe Block, Jan Knippers, Niloy J. Mitra, and Wenping Wang, editors, *Advances in Architectural Geometry 2014*, page 145–157. Springer International Publishing, Cham, 2015.

[BBWSH17] Moritz Bächer, Bernd Bickel, Emily Whiting, and Olga Sorkine-Hornung. Spin-It: Optimizing moment of inertia for spinnable objects. *Commun. ACM*, 60(8):92–99, July 2017.

[BCWZ18] Joshua Bard, Dana Cupkova, Newell Washburn, and Garth Zeglin. Robotic concrete surface finishing: a moldless approach to creating thermally tuned surface geometry for architectural building components using profile-3d-printing. *Construction Robotics*, 2(1):53–65, Dec 2018.

[BDK10] Geoffrey Boothroyd, Peter Dewhurst, and Winston A Knight. *Product design for manufacture and assembly*. CRC press, 2010.

[BDZD21] Patrick Bedarf, Alessandro Dutto, Michele Zanini, and Benjamin Dillenburger. Foam 3d printing for construction: A review of applications, materials, and processes. *Automation in Construction*, 130:103861, Oct 2021.

[Bec10] Martin Bechthold. The Return of the Future: A Second Go at Robotic Construction. *Architectural Design*, 80(4):116–121, 2010.

[BG01] Blai Bonet and Héctor Geffner. Planning as heuristic search. *Artificial Intelligence*, 129(1):5–33, 2001.

[BGMG15] Daniel Bryce, Sicun Gao, David Musliner, and Robert Goldman. Smt-based nonlinear pddl+ planning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[BH14] Serguei Bagrianski and Allison B. Halpern. Form-finding of compressive structures using Prescriptive Dynamic Relaxation. *Computers & Structures*, 132:65–74, February 2014.

[BH19] Giulio Brugnaro and Sean Hanna. Adaptive robotic carving. In Jan Willmann, Philippe Block, Marco Hutter, Kendra Byrne, and Tim Schork, editors, *Robotic Fabrication in Architecture, Art and Design 2018*, page 336–348, Cham, 2019. Springer International Publishing.

[BK07] Tobias Bonwetsch and Matthias Kohler. Digitally fabricating non-standardised brick walls. In *ManuBuild Conference Proceedings*, page 6, 2007.

[BK13] Martin Bechthold and Nathan King. Design robotics. In Sigrid Brell-Çokcan and Johannes Braumann, editors, *Rob | Arch 2012*, page 118–130, Vienna, 2013. Springer.

[BKGK06] Tobias Bonswetch, Daniel Kobel, Fabio Gramazio, and Matthias Kohler. The informed wall: applying additive digital fabrication techniques on architecture. page 489–495, Louisville (Kentucky), USA, 2006.

[BKK13] Jan Brüninghaus, Carsten Krewet, and Bernd Kuhlenkötter. Robot assisted asymmetric incremental sheet forming. In Sigrid Brell-Çokcan and Johannes Braumann, editors, *Rob | Arch 2012*, page 155–160, Vienna, 2013. Springer.

[BKLP13] Jennifer Barry, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. A hierarchical approach to manipulation with diverse actions. In

Robotics and Automation (ICRA), 2013 IEEE International Conference on, pages 1799–1806. IEEE, 2013.

[BL14] Thomas Bock and Silke Langenberg. Changing building sites: Industrialisation and automation of the building process. *Architectural Design*, 84(3):88–99, 2014.

[BLFS+20] Joris Burger, Ena Lloret-Fritschi, Fabio Scotto, Thibault Demoulin, Lukas Gebhard, Jaime Mata-Falcón, Fabio Gramazio, Matthias Kohler, and Robert J. Flatt. Eggshell: Ultra-thin three-dimensional printed formwork for concrete structures. *3D Printing and Additive Manufacturing*, 7(2):48–59, Apr 2020.

[BLPL14] Michael Budig, Willi Viktor Lauer, Raffael Petrovic, and Jason Lim. Design of robotic fabricated high rises. In Wes McGee and Monica Ponce de Leon, editors, *Robotic Fabrication in Architecture, Art and Design 2014*, page 111–129. Springer International Publishing, Cham, 2014.

[BLT+15] Gaurav Bharaj, David I. W. Levin, James Tompkin, Yun Fei, Hanspeter Pfister, Wojciech Matusik, and Changxi Zheng. Computational design of metallophone contact sounds. *ACM Transactions on Graphics*, 34(6):1–13, November 2015.

[BML+14] Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. Projective dynamics: Fusing constraint projections for fast simulation. *ACM transactions on graphics (TOG)*, 33(4):1–11, 2014.

[BMS13] Joshua Bard, Steven Mankouche, and Matthew Schulte. Morphfaux: Recovering architectural plaster by developing custom robotic tools. In Sigrid Brell-Çokcan and Johannes Braumann, editors, *Rob | Arch 2012*, page 139–142. Springer Vienna, Vienna, 2013.

[BN95] Christer Bäckström and Bernhard Nebel. Complexity results for sas+ planning. *Computational Intelligence*, 11(4):625–655, 1995.

[BO07] Philippe Block and John Ochsendorf. Thrust Network Analysis: A New Methodology for Three-Dimensional Equilibrium. *Journal of the International Association for Shell and Spatial Structures*, 48(3):167–173, December 2007.

[Bra22] BranchTech. Branch Technology. ttps://branchtechnology.com/, 2022.

[Bro07] Torgny Brogårdh. Present and future robot control development—an industrial perspective. *Annual Reviews in Control*, 31(1):69–79, 2007.

[Bro16] Blaine Brownell. A History of Wood and Craft in Japanese Design. https://www.architectmagazine.com/technology/a-history-of-wood-and-craft-in-japanese-design_o, June 2016.

[Bro19] Nathan C. (Nathan Collin) Brown. *Early building design using multi-objective data approaches*. PhD thesis, Massachusetts Institute of Technology, 2019.

[BSK+09] Jur van den Berg, Mike Stilman, James Kuffner, Ming Lin, and Dinesh Manocha. Path Planning among Movable Obstacles: A Probabilistically Complete Approach. In *Algorithmic Foundation of Robotics VIII*, Springer Tracts in Advanced Robotics, pages 599–614. Springer, Berlin, Heidelberg, 2009.

[BSK11] Dmitry Berenson, Siddhartha Srinivasa, and James Kuffner. Task space regions: A framework for pose-constrained manipulation planning. *The International Journal of Robotics Research*, 30(12):1435–1460, 2011.

[BSK+19] Patrick Baudisch, Arthur Silber, Yannis Kommana, Milan Gruner, Ludwig Wall, Kevin Reuss, Lukas Heilman, Robert Kovacs, Daniel Rechlitz, and Thijs Roumen. Kyub: A 3D Editor for Modeling Sturdy Laser-Cut Objects. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, Glasgow Scotland Uk, May 2019. ACM.

[BSZD22] Patrick Bedarf, Anna Szabo, Michele Zanini, and Benjamin Dillenburger. Robotic 3d printing of mineral foam for a lightweight composite concrete slab. In *Proceedings of 27th International Conference of the Association for ComputerAided Architectural Design Research in Asia (CAADRIA)*, volume 2, page 61–70, 2022.

[BTAP20] Edvard P. G. Bruun, Ian Ting, Sigrid Adriaenssens, and Stefana Parascho. Human–robot collaboration: a fabrication framework for the sequential design and construction of unplanned spatial structures. *Digital Creativity*, 31(4):320–336, Oct 2020.

[BVMB18] Shajay Bhooshan, Tom Van Mele, and Philippe Block. Equilibrium-aware shape design for concrete printing. In Klaas De Rycke, Christoph Gengnagel, Olivier Baverel, Jane Burry, Caitlin Mueller, Minh Man Nguyen, Philippe Rahm, and Mette Ramsgaard Thomsen, editors, *Humanizing Digital Reality: Design Modelling Symposium Paris 2017*, page 493–508. Springer, Singapore, 2018.

[BVMB20] Shajay Bhooshan, Tom Van Mele, and Philippe Block. Morph & slerp: Shape description for 3d printing of concrete. In *Symposium* *on Computational Fabrication*, page 1–10, Virtual Event USA, Nov 2020. ACM.

[CAG09] Stephane Cambon, Rachid Alami, and Fabien Gravot. A Hybrid Approach to Intricate Motion, Manipulation and Task Planning. *International Journal of Robotics Research (IJRR)*, 28, 2009.

[Can88] John Canny. *The complexity of robot motion planning*. MIT press, 1988.

[Car20] Josephine V. Carstensen. Topology optimization with nozzle size restrictions for material extrusion-type additive manufacturing. *Structural and Multidisciplinary Optimization*, 62(5):2481–2497, November 2020.

[CAS] CASTCONNEX | UMass Amherst John W. Olver Design Building. https://www.castconnex.com/projects/umass-amherst-john-w-olver-design-building.

[CB16] Erwin Coumans and Yunfei Bai. PyBullet, a Python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016.

[CBVS20] Stephanie Chaltiel, Maite Bravo, Diederik Veenendaal, and Gavin Sayers. Drone Spraying on Light Formwork for Mud Shells. In Bob Sheil, Mette Ramsgaard Thomsen, Martin Tamke, and Sean Hanna, editors, *Design Transactions*, Rethinking Information Modelling for a New Material Age, pages 150–157. UCL Press, 2020.

[CCC+18] Kristof Crolla, Paul Hung Hon Cheng, Ding Yuen Shan Chan, Arthur Ngo Foon Chan, and Darwin Lau. Inflatable architecture production with cable-driven robots. In *Proceedings of CAADRIA*, page 10, 2018.

[CDPR20] Kunaljit Chadha, Alexandre Dubor, Laura PuigpInes, and Irene Rafols. Space filling curves for optimising single point incremental sheet forming using supervised learning algorithms. In *Proeedings of eCAADe*, page 555–562, Berlin, Germany, 2020.

[CELM14] Brandon Clifford, Nazareth Ekmekjian, Patrick Little, and Andrew Manto. Variable carving volume casting. In Wes McGee and Monica Ponce de Leon, editors, *Robotic Fabrication in Architecture, Art and Design 2014*, page 3–15. Springer International Publishing, Cham, 2014.

[CFK+22] Aidan Curtis, Xiaolin Fang, Leslie Pack Kaelbling, Tomás Lozano-Pérez, and Caelan Reed Garrett. Long-horizon manipulation of unknown objects via task and motion planning with estimated affordances. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1940–1946. IEEE, 2022.

[CFL13] Amanda Coles, M Fox, and D Long. A hybrid lp-rpg heuristic for modelling numeric resource flows in planning. *Journal of Artificial Intelligence Research*, 46:343–412, 2013.

[CFLM16] Michael Cashmore, Maria Fox, Derek Long, and Daniele Magazzeni. A compilation of the full pddl+ language into smt. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[CKPT17] Henrik I. Christensen, Arindam Khan, Sebastian Pokutta, and Prasad Tetali. Approximation and online algorithms for multidimensional bin packing: A survey. *Computer Science Review*, 24:63–79, May 2017.

[CKS16] Ron Culver, Julia Koerner, and Joseph Sarafian. Fabric forms: The robotic positioning of fabric formwork. In Dagmar Reinhardt, Rob Saunders, and Jane Burry, editors, *Robotic Fabrication in Architecture, Art and Design 2016*, page 106–121. Springer International Publishing, Cham, 2016.

[Cla20] Mollie Claypool. Towards discrete automation. In *Fabricate 2020*, Making Resilient Architecture, page 272–279. UCL Press, 2020.

[CLF+18] Xuelin Chen, Honghua Li, Chi-Wing Fu, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. 3D fabrication with universal building blocks and pyramidal shells. *ACM Transactions on Graphics*, 37(6):1–15, December 2018.

[CLH+22] Jingkai Chen, Jiaoyang Li, Yijiang Huang, Caelan Garrett, Dawei Sun, Chuchu Fan, Andreas Hofmann, Caitlin Mueller, Sven Koenig, and Brian C. Williams. Cooperative Task and Motion Planning for Multi-Arm Assembly Systems. *arXiv preprint arXiv:2203.02475*, March 2022.

[Clo18] Hanne Cloesen. *Structurally Informed Exploration of a Grammar-Based Design Space: An Application of Evolutionary Algorithms to Graphic-Statics Based Shape Grammars*. Master thesis, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland, 2018.

[CML+17] Weikai Chen, Yuexin Ma, Sylvain Lefebvre, Shiqing Xin, Jonàs Martínez, and wenping wang. Fabricable tile decors. *ACM Transactions on Graphics*, 36(6):1–15, November 2017.

[Com22] Robot Components. Robotcomponents. https://github.com/RobotComponents/RobotComponents, May 2022.

[COSS13] Jens Cortsen, Silvan Oesterle, Dorthe Sølvason, and Hanno Stehling. From digital design to automated production. In Sigrid Brell-Çokcan and Johannes Braumann, editors, *Rob | Arch 2012*, page 149–154. Springer Vienna, Vienna, 2013.

[Cou15] Erwin Coumans. Bullet physics simulation. In *ACM SIGGRAPH 2015 Courses*, page 7. ACM, 2015.

[CPMS14] Paolo Cignoni, Nico Pietroni, Luigi Malomo, and Roberto Scopigno. Field-aligned mesh joinery. *ACM Transactions on Graphics*, 33(1):1–12, January 2014.

[CS16] Lujie Chen and Lawrence Sass. Fresh Press Modeler: A generative system for physically based low fidelity prototyping. *Computers & Graphics*, 54:157–165, February 2016.

[CSLM13] Desai Chen, Pitchaya Sitthi-amorn, Justin T. Lan, and Wojciech Matusik. Computing and Fabricating Multiplanar Models. *Computer Graphics Forum*, 32(2pt3):305–315, 2013.

[CTN+13] Stelian Coros, Bernhard Thomaszewski, Gioacchino Noris, Shinjiro Sueda, Moira Forberg, Robert W. Sumner, Wojciech Matusik, and Bernd Bickel. Computational design of mechanical characters. *ACM Transactions on Graphics*, 32(4):1–12, July 2013.

[Cuv20] Pierre Cuvilliers. *The Constrained Geometry of Structures: Optimization Methods for Inverse Form-Finding Design*. PhD thesis, Massachusetts Institute of Technology, Cambridge, 2020.

[CYJ21] Fang-Che Cheng, Chia-Ching Yen, and Tay-Sheng Jeng. Object recognition and user interface design for vision-based autonomous robotic grasping point determination. In *proceedings of CAADRIA*, page 633–642, Hong Kong, 2021.

[CyL19] Jose Luis García del Castillo y López. Robot ex machina: A framework for real-time robot programming and control. In *Proceedings of ACADIA*, 2019.

[CZL+15] Xuelin Chen, Hao Zhang, Jinjie Lin, Ruizhen Hu, Lin Lu, Qixing Huang, Bedrich Benes, Daniel Cohen-Or, and Baoquan Chen. Dapper: Decompose-and-pack for 3D printing. *ACM Transactions on Graphics*, 34(6):1–12, November 2015.

[CZS+19] Subramanian Chidambaram, Yunbo Zhang, Venkatraghavan Sundararajan, Niklas Elmqvist, and Karthik Ramani. Shape Structuralizer: Design, Fabrication, and User-driven Iterative Refinement of 3D Mesh Models. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, Glasgow Scotland Uk, May 2019. ACM.

[CZXZ14] Xiang Chen, Changxi Zheng, Weiwei Xu, and Kun Zhou. An Asymptotic Numerical Method for Inverse Elastic Shape Design. *ACM Trans. Graph.*, 33(4):95:1–95:11, July 2014.

[CZY19] Zhewen Chen, Liming Zhang, and Philip F Yuan. Innovative design approach to optimized performance on large-scale robotic 3d-printed spatial structure. In *Proceedings of the 24th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2019*, volume 2, page 451–460, 2019.

[Dan20] Renaud Aleis Pierre Emile Danhaive. *Structural design synthesis using machine learning*. PhD thesis, Massachusetts Institute of Technology, 2020.

[dAVS18] Pedro de Azambuja Varela and Sousa. Reinforced, reusable, reconfigurable molds for cast voussoirs. In *Kepczynska-Walczak, A, Bialkowski, S (eds.), Computing for a better tomorrow - Proceedings of the 36th eCAADe Conference - Volume 1, Lodz University of Technology, Lodz, Poland, 19-21 September 2018, pp. 771-780*. CUMINCAD, 2018.

[DCEF+17] Tim Detert, Sami Charaf Eddine, Jean-Christophe Fauroux, Tobias Haschke, Francesco Becchi, Burkhard Corves, Roberto Guzman, Fabian Herb, Bruno Linéatte, and Daniel Martin. Bots2rec: introducing mobile robotic units on construction sites for asbestos rehabilitation. *Construction Robotics*, 1(1):29–37, Dec 2017.

[DD13] Alexandre Dubor and Gabriel-Bello Diaz. Magnetic architecture: Generative design through sensoric robots. In Sigrid Brell-Çokcan and Johannes Braumann, editors, *Rob | Arch 2012*, page 206–213, Vienna, 2013. Springer.

[DDB21] Rushi Dai, Orwa Diraneyya, and Sigrid Brell-Çokcan. Improving data communication on construction sites via LoRaWAN. *Construction Robotics*, 5(2):87–100, June 2021.

[DDL+22] Kathrin Dörfler, Gido Dielemans, Lukas Lachmayer, Tobias Recker, Annika Raatz, Dirk Lowke, and Markus Gerke. Additive manufacturing using mobile robots: Opportunities and challenges for building construction. *Cement and Concrete Research*, 158:106772, 2022.

[Dec03] Rina Dechter. *Constraint processing*. Morgan Kaufmann, 2003.

[DEK+09] Christian Dornhege, Patrick Eyerich, Thomas Keller, Sebastian Trüg, Michael Brenner, and Bernhard Nebel. Semantic Attachments for Domain-Independent Planning Systems. In *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 114–121. AAAI Press, 2009.

[DEKW+20] Rebeca Duque Estrada, Fabian Kannenberg, Hans Jakob Wagner, Maria Yablonina, and Achim Menges. Spatial winding: cooperative heterogeneous multi-robot system for fibrous structures. *Construction Robotics*, 4(3):205–215, Dec 2020.

[DFW87] Thomas De Fazio and Daniel Whitney. Simplified generation of all mechanical assembly sequences. *IEEE Journal on Robotics and Automation*, 3(6):640–658, 1987.

[DH19] Mohamed Dawod and Sean Hanna. Bim-assisted object recognition for the on-site autonomous robotic assembly of discrete structures. *Construction Robotics*, 3(1):69–81, Dec 2019.

[DHM14] Rachel Dickey, Jili Huang, and Saurabh Mhatre. Objects of rotation. In Wes McGee and Monica Ponce de Leon, editors, *Robotic Fabrication in Architecture, Art and Design 2014*, page 233–247. Springer International Publishing, Cham, 2014.

[DHS+19] Pradeep Devadass, Tobias Heimig, Sven Stumm, Ethan Kerber, and Sigrid Brell-Cokcan. Robotic constraints informed design process. In *Proceedings of the 39th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, page 10, 2019.

[DHV+18] Stylianos Dritsas, Samuel E. P. Halim, Yadunund Vijay, Naresh G. Sanandiya, and Javier G. Fernandez. Digital fabrication with natural composites. *Construction Robotics*, 2(1):41–51, Dec 2018.

[Dia10] Rosen Diankov. *Automated construction of robotic manipulation programs*. PhD thesis, Robotics Institute, Carnegie Mellon University, 2010.

[DIC+19] Alexandre Dubor, Jean-Baptiste Izard, Edouard Cabay, Aldo Sollazzo, Areti Markopoulou, and Mariola Rodriguez. On-site robotics for sustainable construction. In Jan Willmann, Philippe Block, Marco Hutter, Kendra Byrne, and Tim Schork, editors, *Robotic Fabrication in Architecture, Art and Design 2018*, page 390–401, Cham, 2019. Springer International Publishing.

[DK08] Rosen Diankov and James Kuffner. OpenRAVE: A Planning Architecture for Autonomous Robotics. Technical Report CMU-RI-TR-08-34, Robotics Institute, Carnegie Mellon University, 2008.

[DKLP16] Ashwin Deshpande, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Decidability of Semi-Holonomic Prehensile Task and Motion Planning. *Workshop on Algorithmic Foundations of Robotics (WAFR)*, 2016.

[DKR+20] Rushi Dai, Ethan Kerber, Felix Reuter, Sven Stumm, and Sigrid Brell-Cokcan. The digitization of the automated steel construction through the application of microcontrollers and mqtt. *Construction Robotics*, 4(3):251–259, Dec 2020.

[DMC18] Ruta Desai, James McCann, and Stelian Coros. Assembly-aware design of printable electromechanical devices. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*, page 457–472, Berlin, Germany, 2018. ACM Press.

[DMS90] L S Homem De Mello and Arthur C Sanderson. AND/OR graph representation of assembly plans. *IEEE Transactions on robotics and automation*, 6(2):188–199, 1990.

[Dor15] Christian Dornhege. *Task planning for high-level robot control*. PhD thesis, Verlag nicht ermittelbar, 2015.

[DPW+14] Mario Deuss, Daniele Panozzo, Emily Whiting, Yang Liu, Philippe Block, Olga Sorkine-Hornung, and Mark Pauly. Assembling self-supporting structures. *ACM Transactions on Graphics (TOG)*, 33(6):214, 2014.

[DRR13] Kathrin Dörfler, Florian Rist, and Romana Rust. Interlacing: An experimental approach to integrating digital and physical design methods. In Sigrid Brell-Çokcan and Johannes Braumann, editors, *Rob | Arch 2012*, page 82–91, Vienna, 2013. Springer.

[DS18] Stylianos Dritsas and Gim Song Soh. Building robotics design for construction: design considerations and principles for mobile systems. *Constr Robot*, 2018.

[DSBR15] Mehmet Dogar, Andrew Spielberg, Stuart Baker, and Daniela Rus. Multi-robot grasp planning for sequential assembly operations. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 193–200. IEEE, 2015.

[DSG+16] Kathrin Dörfler, Timothy Sandy, Markus Giftthaler, Fabio Gramazio, Matthias Kohler, and Jonas Buchli. Mobile robotic brickwork. In *Robotic Fabrication in Architecture, Art and Design 2016*, pages 204–217. Springer, 2016.

[DSM13] Karola Dierichs, Tobias Schwinn, and Achim Menges. Robotic pouring of aggregate structures. In Sigrid Brell-Çokcan and Johannes Braumann, editors, *Rob | Arch 2012*, page 196–205, Vienna, 2013. Springer.

[dSPGA13] Lavidra de Silva, Amit Kumar Pandey, Mamoun Gharbi, and Rachd Alami. Towards Combining HTN Planning and Geometric

Task Planning. In *RSS Workshop on Combined Robot Motion Planning and AI Planning for Practical Applications*, 2013.

[DWW+18] Chengkai Dai, Charlie C L Wang, Chenming Wu, Sylvain Lefebvre, Guoxin Fang, and Yongjin Liu. Support-free volume printing by multi-axis motion. *ACM Transactions on Graphics (TOG)*, 2018.

[DZB+19] Niccolò Dambrosio, Christoph Zechmeister, Serban Bodea, Valentin Koslowski, Marta Gil-Pérez, Bas Rongen, Jan Knippers, and Achim Menges. Buga fibre pavilion: Towards an architectural application of novel fiber composite building systems. In *Proceedings of ACADIA*, page 10, 2019.

[Dö18] Kathrin Dörfler. *Strategies for Robotic in Situ Fabrication*. Doctoral thesis, ETH Zurich, Oct 2018.

[EG14] Khaled Elashry and Ruairi Glynn. An approach to automated construction using adaptive programing. In Wes McGee and Monica Ponce de Leon, editors, *Robotic Fabrication in Architecture, Art and Design 2014*, page 51–66. Springer International Publishing, Cham, 2014.

[EGK17] Philipp Eversmann, Fabio Gramazio, and Matthias Kohler. Robotic prefabrication of timber structures: towards automated large-scale spatial assembly. *Construction Robotics*, 1(1-4):49–60, 2017.

[EHN94] Kutluhan Erol, James Hendler, and Dana S Nau. HTN planning: Complexity and expressivity. In *AAAI*, volume 94, pages 1123–1128, 1994.

[EHP+11] Esra Erdem, Kadir Haspalamutgil, Can Palaz, Volkan Patoglu, and Tansel Uras. Combining high-level causal reasoning with low-level geometric reasoning and motion planning for robotic manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[EJLFGK20] Selen Ercan Jenny, Ena Lloret-Fritschi, Fabio Gramazio, and Matthias Kohler. Crafting plaster through continuous mobile robotic fabrication on-site. *Construction Robotics*, 4(3):261–271, Dec 2020.

[EKWM20] Rebeca Duque Estrada, Fabian Kannenberg, Hans Jakob Wagner, and Achim Menges. Integrative design methods for spatial winding. In *Proceedings of Advances in Architectural Geometry*, page 20, 2020.

[Eri04] Christer Ericson. *Real-time collision detection*. CRC Press, 2004.

[ERP+19] Jimmy Etienne, Nicolas Ray, Daniele Panozzo, Samuel Hornus, Charlie C. L. Wang, Jonàs Martínez, Sara McMains, Marc Alexa, Brian Wyvill, and Sylvain Lefebvre. CurviSlicer: Slightly curved slicing for 3-axis printers. *ACM Transactions on Graphics*, 38(4):1–11, August 2019.

[FAG+20] I. Filoscia, T. Alderighi, D. Giorgi, L. Malomo, M. Callieri, and P. Cignoni. Optimizing Object Decomposition to Reduce Visual Artifacts in 3D Printing. *Computer Graphics Forum*, 39(2):423–434, May 2020.

[Fai13] William Fairham. *Woodwork Joints: How to Make and Where to Use Them*. Skyhorse, New York, NY, 1st edition edition, Mar 2013.

[FCM+18] Filippo A. Fanni, Gianmarco Cherchi, Alessandro Muntoni, Alessandro Tola, and Riccardo Scateni. Fabrication oriented shape decomposition using polycube mapping. *Computers & Graphics*, 77:183–193, December 2018.

[FGS17] FABLAB, RUAIRI GLYNN, and BOB SHEIL. Matter & making. In *Fabricate 2011*, Making Digital Architecture, page 73–84. UCL Press, dgo-digital original edition, 2017.

[FGWK18] Enrique Fernandez-Gonzalez, Brian Williams, and Erez Karpas. Scottyactivity: Mixed discrete-continuous planning with convex optimization. *Journal of Artificial Intelligence Research*, 62:579–664, 2018.

[FHL14] Jared Friedman, Ahmed Hosny, and Amanda Lee. Robotic bead rolling. In Wes McGee and Monica Ponce de Leon, editors, *Robotic Fabrication in Architecture, Art and Design 2014*, page 83–98. Springer International Publishing, Cham, 2014.

[Fit89] John Fitchen. *Building construction before mechanization*. MIT Press, 1989.

[FKM14] Jared Friedman, Heamin Kim, and Olga Mesa. Experiments in additive clay depositions. In Wes McGee and Monica Ponce de Leon, editors, *Robotic Fabrication in Architecture, Art and Design 2014*, page 261–272. Springer International Publishing, Cham, 2014.

[FLE+20] Thilo Feucht, Jörg Lange, Maren Erven, Christopher Borg Costanzi, Ulrich Knaack, and Benedikt Waldschmitt. Additive manufacturing by means of parametric robot programming. *Construction Robotics*, 4(1):31–48, Jun 2020.

[FM01] Sándor P. Fekete and Joseph S. B. Mitchell. Terrain decomposition and layered manufacturing. *International Journal of Computational Geometry & Applications*, 11(06):647–668, December 2001.

[FMB15] Ursula Frick, Tom Van Mele, and Philippe Block. Decomposing Three-Dimensional Shapes into Self-supporting, Discrete-Element Assemblies. In *Modelling Behaviour*, pages 187–201. Springer, Cham, 2015.

[FN71] Richard E Fikes and Nils J Nilsson. {STRIPS}: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence*, 2:189–208, 1971.

[FNRK15] Vahid Faghihi, Ali Nejat, Kenneth F. Reinschmidt, and Julian H. Kang. Automation in construction scheduling: a review of the literature. *The International Journal of Advanced Manufacturing Technology*, 81(9–12):1845–1856, Dec 2015.

[Fou17] Renzo Piano Foundation. IBM Padiglione itinerante. https://www.fondazionerenzopiano.org/en/project/ibm-padiglione-itinerante/, May 2017.

[FRK14] Vahid Faghihi, Kenneth F. Reinschmidt, and Julian H. Kang. Construction scheduling using genetic algorithm based on building information model. *Expert Systems with Applications*, 41(16):7565–7578, Nov 2014.

[FSC+12] Jim Fraser, David Seel, Jonathan Chadwick, Ketan Valambhia, and Alister Offiler. Cannon Place, London: Design and construction over a live railway station. *Proceedings of the Institution of Civil Engineers - Civil Engineering*, 165(2):74–81, May 2012.

[FSK08] Alexander I. J. Forrester, András Sóbester, and Andy J. Keane. *Engineering Design via Surrogate Modelling: A Practical Guide*. Wiley, first edition, July 2008.

[FSY+15] Chi-Wing Fu, Peng Song, Xiaoqi Yan, Lee Wei Yang, Pradeep Kumar Jayaraman, and Daniel Cohen-Or. Computational interlocking furniture assembly. *ACM Transactions on Graphics (TOG)*, 34(4):91, 2015.

[FT14] Pavlos Fereos and Marios Tsiliakos. Isoprototyping - rapid robotic aided fabrication for double curvature surfaces. In *Proceedings of eCAADe*, page 433–443, Newcastle upon Tyne, UK, 2014.

[Gan20] Augusto Gandia. *Robotic Fabrication Simulation: A Computational Method for the Design of Fabrication-aware Spatial Structures*. PhD thesis, ETH Zurich, 2020.

[GBG00] E Gambao, C Balaguer, and F Gebhart. Robot assembly system for computer-integrated construction. *Automation in Construction*, 9(5):479–487, Sep 2000.

[GCH+21] Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomas Lozano-Pérez. Integrated Task and Motion Planning. *Annual review of control, robotics, and autonomous systems*, 4, 2021.

[GD22] Rena Giesecke and Benjamin Dillenburger. Large-scale robotic fabrication of polychromatic relief glass. *International Journal of Architectural Computing*, 20(1):18–30, Mar 2022.

[GDB+19] Nadja Gaudillière, Romain Duballet, Charles Bouyssou, Alban Mallet, Philippe Roux, Mahriz Zakeri, and Justin Dirrenberger. Large-scale additive manufacturing of ultra-high-performance concrete of integrated formwork for truss-shaped pillars. In *Robotic Fabrication in Architecture, Art and Design 2018*, page 459–472. Springer, Cham, 2019.

[GG16] Andrew T Gaynor and James K Guest. Topology optimization considering overhang constraints: Eliminating sacrificial support material in additive manufacturing through design. *Structural and Multidisciplinary Optimization*, 54(5):1157–1172, 2016.

[GHB18] Matthew K Gelber, Greg Hurst, and Rohit Bhargava. Freeform assembly planning. *IEEE Transactions on Automation Science and Engineering*, 16(3):1315–1329, 2018.

[GHLPM20] Caelan Reed Garrett, Yijiang Huang, Tomás Lozano-Pérez, and Caitlin Tobin Mueller. Scalable and probabilistically complete planning for robotic spatial extrusion. In *Robotics: Science and Systems (RSS)*, 2020.

[GHR15] Salomé Galjaard, Sander Hofman, and Shibo Ren. New Opportunities to Optimize Structural Designs in Metal by Using Additive Manufacturing. In Philippe Block, Jan Knippers, Niloy J. Mitra, and Wenping Wang, editors, *Advances in Architectural Geometry 2014*, pages 79–93, Cham, 2015. Springer International Publishing.

[GHRV18] Andrei Gheorghe, Philipp Hornung, Sigurd Reiss, and Robert Vierlinger. Architecture challenge 16 - robotic contouring. In *Proceedings of eCAADe 2018*, page 8, 2018.

[Gif18] Markus Giftthaler. *Towards a Unified Framework of Efficient Algorithms for Numerical Optimal Robot Control*. Doctoral thesis, ETH Zurich, 2018. Accepted: 2020-04-01T08:29:24Z.

[GLFB12] Peter Gregory, Derek Long, Maria Fox, and J Christopher Beck. Planning modulo theories: Extending the planning paradigm. In *Twenty-Second International Conference on Automated Planning and Scheduling*, 2012.

[GLPK15] Caelan Reed Garrett, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Backward-Forward Search for Manipulation Planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2015-Decem, 2015.

[GLPK18a] Caelan Reed Garrett, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. FFRob: Leveraging symbolic planning for efficient task and motion planning. *International Journal of Robotics Research*, 37(1):104–136, 2018.

[GLPK18b] Caelan Reed C.R. Garrett, Tomás Lozano-Pérez, and L.P. Leslie Pack Kaelbling. Sampling-based methods for factored task and motion planning. *The International Journal of Robotics Research*, 37(13-14), 2018.

[GLPK20] Caelan R. Garrett, Tomás Lozano-Pérez, and Leslie P. Kaelbling. PDDLStream: Integrating Symbolic Planners and Blackbox Samplers. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 2020.

[GM15] Somayé Ghandi and Ellips Masehian. Review and taxonomies of assembly and disassembly path planning problems and approaches. *Computer-Aided Design*, 67:58–86, 2015.

[GMW14] Fabio Gramazio, Kohler Matthias, and Jan Willmann. *The robotic touch*. Park Books, 2014.

[GNT04] Malik Ghallab, Dana S Nau, and Paolo Traverso. *Automated Planning: Theory and Practice*. Elsevier, 2004.

[Goe97] Michel X. Goemans. Semidefinite programming in combinatorial optimization. *Mathematical Programming*, 79(1):143–161, October 1997.

[GPL+20] Caelan Reed Garrett, Chris Paxton, Tomás Lozano-Pérez, Leslie Pack Kaelbling, and Dieter Fox. Online Replanning in Belief Space for Partially Observable Task and Motion Problems. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5678–5684, May 2020.

[GPR+18] Augusto Gandia, Stefana Parascho, Romana Rust, Gonzalo Casas, Fabio Gramazio, and Matthias Kohler. Towards Automatic Path Planning for Robotically Assembled Spatial Structures. In *Robotic Fabrication in Architecture, Art and Design*, pages 59–73. Springer, 2018.

[GPZ20] Francisca Gil-Ureta, Nico Pietroni, and Denis Zorin. Reinforcement of General Shell Structures. *ACM Transactions on Graphics*, 39(5):1–19, June 2020.

[Gra15] Gramazio & Kohler Research, ETH Zurich. Extruded Structures, elective course at eth zurich. https://gramaziokohler.arch.ethz.ch/web/e/lehre/284.html, 2015.

[GRS15] Ian Gibson, David Rosen, and Brent Stucker. *Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*. Springer, 2015.

[GSD+14] Akash Garg, Andrew O. Sageman-Furnas, Bailin Deng, Yonghao Yue, Eitan Grinspun, Mark Pauly, and Max Wardetzky. Wire mesh design. *ACM Transactions on Graphics*, 33(4):1–12, July 2014.

[GSD+17] Markus Giftthaler, Timothy Sandy, Kathrin Dörfler, Ian Brooks, Mark Buckingham, Gonzalo Rey, Matthias Kohler, Fabio Gramazio, and Jonas Buchli. Mobile robotic fabrication at 1: 1 scale: the In situ Fabricator. *Construction Robotics*, 1(1-4):3–14, 2017.

[GT97] J M Gere and S P Timoshenko. Mechanics of materials. *PWS-KENT Publishing Company*, 534:4, 1997.

[GWYN19] Yisong Gao, Lifang Wu, Dong-Ming Yan, and Liangliang Nan. Near support-free multi-directional 3D printing via global-optimal decomposition. *Graphical Models*, 104:101034, July 2019.

[GZN+15] Wei Gao, Yunbo Zhang, Diogo C. Nazzetta, Karthik Ramani, and Raymond J. Cipra. RevoMaker: Enabling Multi-directional and Functionally-embedded 3D printing using a Rotational Cuboidal Platform. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 437–446, Charlotte NC USA, November 2015. ACM.

[HADWM21] Yijiang Huang, Latifa Alkhayat, Catherine De Wolf, and Caitlin Tobin Mueller. Algorithmic circular design with reused structural elements: Method and tool. In *Proceedings of International FIB symposium of Conceptual Design of Structures*, 2021.

[Has00] Yukio Hasegawa. A new wave of construction automation and robotics in japan. In *Proceedings of ISARC*, Taipei, Taiwan, Sep 2000.

[Hau12] Kris Hauser. The minimum constraint removal problem with three robotics applications. In *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, volume 33, pages 5–17, 2012.

[Hau13] Kris K Hauser. Minimum Constraint Displacement Motion Planning. In *Robotics: Science and Systems (RSS)*, 6 2013.

[HBA13] Kristian Hildebrand, Bernd Bickel, and Marc Alexa. Orthogonal slicing for additive manufacturing. *Computers & Graphics*, 37(6):669–675, October 2013.

[HCM18] Yijiang Huang, Josephine Carstensen, and Caitlin Mueller. 3D truss topology optimization for automated robotic spatial extrusion. In *Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium 2018*. 2018.

[HCTM18] Yijiang Huang, Josephine Carstensen, Lavender Tessmer, and Caitlin Mueller. Robotic extrusion of architectural structures with nonstandard topology. In *Robotic Fabrication in Architecture, Art and Design 2018*. Springer, 2018.

[HDB+20] Norman Hack, Inka Dressler, Leon Brohmann, Stefan Gantner, Dirk Lowke, and Harald Kloft. Injection 3D Concrete Printing (I3DCP): Basic Principles and Case Studies. *Materials*, 13(5):1093, January 2020.

[HE80] Robert M Haralick and Gordon L Elliott. Increasing tree search efficiency for constraint satisfaction problems. *Artificial intelligence*, 14(3):263–313, 1980.

[Hel06] Malte Helmert. The Fast Downward Planning System. *Journal of Artificial Intelligence Research (JAIR)*, 26:191–246, 2006.

[HGM18] Yijiang Huang, Caelan R Garrett, and Caitlin T Mueller. Automated sequence and motion planning for robotic spatial extrusion of 3D trusses. *Construction Robotics*, 2(1):15–39, 12 2018.

[HGT+21] Yijiang Huang, Caelan Reed Garrett, Ian Ting, Stefana Parascho, and Caitlin Tobin Mueller. Robotic additive construction of bar structures: Unified sequence and motion planning. *Construction Robotics*, pages 115–130, 2021.

[HKK+17] Volker Helm, Michael Knauss, Thomas Kohlhammer, Fabio Gramazio, and Matthias Kohler. Additive robotic fabrication of complex timber structures. In *Advancing Wood Architecture: A Computational Approach*, pages 29 – 43. Routledge, 2017.

[HL10] Kris Hauser and Jean-Claude Latombe. Multi-modal Motion Planning in Non-expansive Spaces. *International Journal of Robotics Research (IJRR)*, 29:897–915, 2010.

[HL14] Norman Hack and Willi Viktor Lauer. Mesh-Mould: Robotically Fabricated Spatial Meshes as Reinforced Concrete Formwork. *Architectural Design*, 84(3):44–53, 2014.

[HLK19] Norman Hack, Hendrik Lindemann, and Harald Kloft. Adaptive modular spatial structures for shotcrete 3d printing. In *Proceedings of CAADRIA*, page 363–372, Wellington, New Zealand, 2019.

[HLW00] D. Halperin, J.-C. Latombe, and R. H. Wilson. A general framework for assembly planning: The motion space approach. *Algorithmica*, 26(3–4):577–601, Apr 2000.

[HLZC14] Ruizhen Hu, Honghua Li, Hao Zhang, and Daniel Cohen-Or. Approximate pyramidal shape decomposition. *ACM Transactions on Graphics*, 33(6):1–12, November 2014.

[HMA15] Philipp Herholz, Wojciech Matusik, and Marc Alexa. Approximating Free-form Geometry with Height Fields for Manufacturing. *Computer Graphics Forum*, 34(2):239–251, 2015.

[HNTH11] Kris Hauser and Victor Ng-Thow-Hing. Randomized multi-modal motion planning for a humanoid robot manipulation task. *International Journal of Robotics Research (IJRR)*, 30(6):676–698, 2011.

[HOD+20] Valentin N. Hartmann, Ozgur S. Oguz, Danny Driess, Marc Toussaint, and Achim Menges. Robust task and motion planning for long-horizon architectural construction planning. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[HOD+21] Valentin N. Hartmann, Andreas Orthey, Danny Driess, Ozgur S. Oguz, and Marc Toussaint. Long-horizon multi-robot rearrangement planning for construction assembly. *Preprint arxiv:2106.02489*, 2021.

[Hof03] Jörg Hoffmann. The metric-ff planning system: Translating“ignoring delete lists”to numeric state variables. *Journal of artificial intelligence research*, 20:291–341, 2003.

[HOL21] Ryan Hughes, Toni Osterlund, and Niels Martin Larsen. Integrated design-for-manufacturing and ar-aided-assembly workflows for lightweight reciprocal frame timber structures. *Construction Robotics*, 5(2):147–157, Jun 2021.

[HPA+04] Julie Heiser, Doantam Phan, Maneesh Agrawala, Barbara Tversky, and Pat Hanrahan. Identification and validation of cognitive design principles for automated generation of assembly instructions. In *Proceedings of the working conference on Advanced visual interfaces*, AVI ’04, page 311–319, New York, NY, USA, May 2004. Association for Computing Machinery.

[HPDC22] Lihui Huang, Roshan Pradhan, Souravik Dutta, and Yiyu Cai. Bim4d-based scheduling for assembling and lifting in precast-enabled construction. *Automation in Construction*, 133:103999, Jan 2022.

[HSG+17] Kathleen Hajash, Bjorn Sparrman, Christophe Guberan, Jared Laucks, and Skylar Tibbits. Large-Scale Rapid Liquid Printing. *3D Printing and Additive Manufacturing*, 4(3):123–132, September 2017.

[HSK+18] Shuai D Han, Nicholas M Stiffler, Athanasios Krontiris, Kostas E Bekris, and Jingjin Yu. Complexity results and fast methods for optimal tabletop rearrangement with overhand grasps. *The International Journal of Robotics Research*, 37(13-14):1775–1795, 2018.

[Hua18] Yijiang Huang. *Automated motion planning for robotic assembly of discrete architectural structures*. Master thesis, Massachusetts Institute of Technology, 2018.

[HVG+21] Yijiang Huang, Leung Pok Yin Victor, Caelan Garrett, Fabio Gramazio, Matthias Kohler, and Caitlin Mueller. The new analog: A protocol for linking design and construction intent with algorithmic planning for robotic assembly of complex structures. In *Symposium on Computational Fabrication*, SCF '21. Association for Computing Machinery, 2021.

[HWT+15] Volker Helm, Jan Willmann, Andreas Thoma, Luka Piškorec, Norman Hack, Fabio Gramazio, and Matthias Kohler. Iridescence print: Robotically printed lightweight mesh structures. *3D Printing and Additive Manufacturing*, 2(3):117–122, 2015.

[HZH+16] Yijiang Huang, Juyong Zhang, Xin Hu, Guoxian Song, Zhongyuan Liu, Lei Yu, and Ligang Liu. Framefab: Robotic fabrication of frame shapes. *ACM Transactions on Graphics (TOG)*, 35(6):224, 2016.

[IDH+17] Jean-Baptiste Izard, Alexandre Dubor, Pierre-Elie Hervé, Edouard Cabay, David Culla, Mariola Rodriguez, and Mikel Barrado. Large-scale 3d printing with cable-driven parallel robots. *Construction Robotics*, 1(1):69–76, Dec 2017.

[IH06] Yuichi Ikeda and Tsunenori Harada. Application of the automated building construction system using the conventional construction method together. In *Proceedings of ISARC*, Tokyo, Japan, Oct 2006.

[IRHS20] Alexandra Ion, Michael Rabinovich, Philipp Herholz, and Olga Sorkine-Hornung. Shape approximation by developable wrapping. *ACM Transactions on Graphics*, 39(6):1–12, December 2020.

[JA18] Ryan Luke Johns and Jeffrey Anderson. Interfaces for adaptive assembly. In *Proceedings of ACADIA*, page 126–135, Mexico City, Mexico, 2018.

[Jac17] Alec Jacobson. Generalized Matryoshka: Computational Design of Nesting Objects. *Computer Graphics Forum*, 36(5):27–35, August 2017.

[Jac19] Alec Jacobson. RodSteward: A Design-to-Assembly System for Fabrication using 3D-Printed Joints and Precision-Cut Rods. *Computer Graphics Forum*, 38(7):765–774, October 2019.

[JARCG19] Benjamin Jenett, Amira Abdel-Rahman, Kenneth Cheung, and Neil Gershenfeld. Material–robot system for assembly of discrete cellular structures. *IEEE Robotics and Automation Letters*, 4(4):4019–4026, Oct 2019.

[JB17] Krystyna Januszkiewicz and Marta Banachowicz. Nonlinear Shaping Architecture Designed with Using Evolutionary Structural Optimization Tools. *IOP Conference Series: Materials Science and Engineering*, 245:082042, October 2017.

[JBD+17] Andrei Jipa, Mathias Bernhard, Benjamin Dillenburger, Nicolas Ruffray, Timothy Wangler, and Robert Flatt. skelethon formwork 3d printed plastic formwork for load-bearing concrete structures. In *Blucher Design Proceedings*, page 345–352, Concepción, Chile, Nov 2017. Editora Blucher.

[Jef16] Michael Jeffers. Autonomous robotic assembly with variable material properties. In Dagmar Reinhardt, Rob Saunders, and Jane Burry, editors, *Robotic Fabrication in Architecture, Art and Design 2016*, page 48–61. Springer International Publishing, Cham, 2016.

[Jim13] Pablo Jiménez. Survey on assembly sequencing: a combinatorial and geometrical perspective. *Journal of Intelligent Manufacturing*, 24(2):235–250, 2013.

[JP20] Gabriella Joyce and Antony Pelosi. Robotic connections for clt panels. In *e 25th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 20*, page 10, 2020.

[JR19] Nathaniel L. Jones and Christoph F. Reinhart. Effects of real-time simulation feedback on design for visual comfort. *Journal of Building Performance Simulation*, 12(3):343–361, May 2019.

[JTSW17] Caigui Jiang, Chengcheng Tang, Hans-Peter Seidel, and Peter Wonka. Design and volume optimization of space structures. *ACM Transactions on Graphics*, 36(4):1–14, July 2017.

[JWM+20] Ryan Luke Johns, Martin Wermelinger, Ruben Mascaro, Dominic Jud, Fabio Gramazio, Matthias Kohler, Margarita Chli, and Marco Hutter. Autonomous dry stone: On-site planning and assembly of stone walls with a robotic excavator. *Construction Robotics*, 4(3):127–140, Dec 2020.

[KAGK17] Thomas Kohlhammer, Aleksandra Anna Apolinarska, Fabio Gramazio, and Matthias Kohler. Design and structural analysis of complex timber structures with glued t-joint connections for robotic assembly:. *International Journal of Space Structures*, Dec 2017.

[Kas22] Alexandre Kaspar. *Garment Design Workflows for On-Demand Machine Knitting*. PhD thesis, Massachusetts Institute of Technology, February 2022.

[KB15] A Krontiris and K E Bekris. Dealing with Difficult Instances of Object Rearrangement. In *Robotics: Science and Systems (RSS)*, Rome, Italy, 9 2015.

[KB16] A Krontiris and K E Bekris. Efficiently Solving General Rearrangement Tasks: A Fast Extension Primitive for an Incremental Sampling-based Planner. In *International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, 2016.

[KCB+19] Markus Kayser, Levi Cai, Christoph Bader, Sara Falcone, Nassia Inglessis, Barrak Darweesh, João Costa, and Neri Oxman. Fiberbots: Design and digital fabrication of tubular structures using robot swarms. In Jan Willmann, Philippe Block, Marco Hutter, Kendra Byrne, and Tim Schork, editors, *Robotic Fabrication in Architecture, Art and Design 2018*, page 285–296, Cham, 2019. Springer International Publishing.

[KCKH12] Ulrich Knaack, Sharon Chung-Klatte, and Reinhard Hasselbach. *Prefabricated systems: Principles of Construction*. Birkhäuser, 2012.

[KCT+11] M Kalakrishnan, S Chitta, E Theodorou, P Pastor, and S Schaal. STOMP: Stochastic trajectory optimization for motion planning. In *2011 IEEE International Conference on Robotics and Automation*, pages 4569–4574, 5 2011.

[KE81] H Koning and J Eizenberg. The Language of the Prairie: Frank Lloyd Wright's Prairie Houses. *Environment and Planning B: Planning and Design*, 8(3):295–323, September 1981.

[KEKD19] Hyunchul Kwon, Martin Eichenhofer, Thodoris Kyttas, and Benjamin Dillenburger. Digital composites: Robotic 3d printing of continuous carbon fiber-reinforced plastics for functionally-graded building components. In Jan Willmann, Philippe Block, Marco Hutter, Kendra Byrne, and Tim Schork, editors, *Robotic Fabrication in Architecture, Art and Design 2018*, page 363–376, Cham, 2019. Springer International Publishing.

[KEN+13] Thomas Kiser, Michael Eigensatz, Minh Man Nguyen, Philippe Bompas, and Mark Pauly. Architectural Caustics — Controlling

Light with Geometry. In Lars Hesselgren, Shrikant Sharma, Johannes Wallner, Niccolo Baldassini, Philippe Bompas, and Jacques Raynaud, editors, *Advances in Architectural Geometry 2012*, pages 91–106, Vienna, 2013. Springer.

[KF11] Sertac Karaman and Emilio Frazzoli. Sampling-based Algorithms for Optimal Motion Planning. *International Journal of Robotics Research (IJRR)*, 30(7):846–894, 2011.

[KFI88] Kimio Kikuchi, Shuzo Furuta, and Takayoshi Imai. Development and the result of practical works of concrete floor finishing robot. In *Proceedings of ISARC*, Tokyo, Japan, Jun 1988.

[KFW19] E. Karasik, R. Fattal, and M. Werman. Object Partitioning for Support-Free 3D-Printing. *Computer Graphics Forum*, 38(2):305–316, May 2019.

[KG13] Nathan King and Jonathan Grinham. Automating eclipsis. In Sigrid Brell-Çokcan and Johannes Braumann, editors, *Rob | Arch 2012*, page 214–221, Vienna, 2013. Springer.

[KHLM17] Bongjin Koo, Jean Hergel, Sylvain Lefebvre, and Niloy J. Mitra. Towards Zero-Waste Furniture Design. *IEEE Transactions on Visualization and Computer Graphics*, 23(12):2627–2640, December 2017.

[Kho04] Behrokh Khoshnevis. Automated construction by contour crafting—related robotics and information technologies. *Automation in Construction*, 13(1):5–19, January 2004.

[KIS+12] Daniel Kopta, Thiago Ize, Josef Spjut, Erik Brunvand, Al Davis, and Andrew Kensler. Fast, effective BVH updates for animated scenes. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 197–204. ACM, 2012.

[KIT+22] Gene Ting-Chun Kao, Antonino Iannuzzo, Bernhard Thomaszewski, Stelian Coros, Tom Van Mele, and Philippe Block. Coupled rigid-block analysis: Stability-aware design of complex discrete-element assemblies. *Computer-Aided Design*, 146:103216, May 2022.

[KJL00] James J Kuffner Jr. and Steven M LaValle. {RRT-Connect}: An efficient approach to single-query path planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2000.

[KKL98] Lydia E Kavraki, Mihail N Kolountzakis, and J-C Latombe. Analysis of probabilistic roadmaps for path planning. *Robotics and Automation, IEEE Transactions on*, 14(1):166–171, 1998.

[KKL14] Jae Woo Kim, Kyung Kyu Kang, and Ji Hyoung Lee. Survey on Automated LEGO Assembly Construction. In *Proceedings of WSCG 2014 Conference on Computer Graphics, Visualization and Computer Vision*. 2014.

[KKS+17] Gene TC Kao, Axel Körner, Daniel Sonntag, Long Nguyen, Achim Menges, and Jan Knippers. Assembly-aware design of masonry shell structures: a computational approach. In *Proceedings of IASS Annual Symposia*, volume 2017, page 1–10. International Association for Shell and Spatial Structures (IASS), 2017.

[KKSS15] Bernhard Kerbl, Denis Kalkofen, Markus Steinberger, and Dieter Schmalstieg. Interactive disassembly planning for complex objects. *Computer Graphics Forum*, 34(2):287–297, 2015.

[KŁH+22] Nicolas Kubail Kalousdian, Grzegorz Łochnicki, Valentin N. Hartmann, Samuel Leder, Ozgur S. Oguz, Achim Menges, and Marc Toussaint. Learning Robotic Manipulation of Natural Materials With Variable Properties for Construction Tasks. *IEEE Robotics and Automation Letters*, 7(2):5749–5756, April 2022.

[KLP11] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Hierarchical task and motion planning in the now. In *ICRA*, 2011.

[KM20] Erez Karpas and Daniele Magazzeni. Automated planning for robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:417–439, 2020.

[KMK18] Zachary Kingston, Mark Moll, and Lydia E Kavraki. Sampling-based methods for motion planning with constraints. *Annual review of control, robotics, and autonomous systems*, 1:159–185, 2018.

[KMK19a] Zachary Kingston, Mark Moll, and Lydia E. Kavraki. Exploring implicit spaces for constrained sampling-based planning. *International Journal of Robotics Research*, 2019.

[KMK19b] Zachary Kingston, Mark Moll, and Lydia E. Kavraki. Exploring implicit spaces for constrained sampling-based planning. *International Journal of Robotics Research*, 38(10-11):1151–1178, 2019.

[KN14] Ammar Kalo and Michael Jake Newsum. An investigation of robotic incremental sheet metal forming as a method for prototyping parametric architectural skins. In Wes McGee and Monica Ponce de Leon, editors, *Robotic Fabrication in Architecture, Art and Design 2014*, page 33–49. Springer International Publishing, Cham, 2014.

[KO05] Axel Kilian and John Ochsendorf. Particle-spring systems for structural form finding. *Journal Of The International Association For Shell And Spatial Structures*, 46(147):8, 2005.

[KO13] Steven Keating and Neri Oxman. Compound fabrication: A multi-functional robotic platform for digital design and fabrication. *Robotics and Computer-Integrated Manufacturing*, 29(6):439–448, 2013.

[KPB+14] Mehrnoush Latifi Khorasgani, Daniel Prohasky, Jane Burry, Akbar Akbarzadeh, and Nicholas Williams. Robothermodon: An artificial sun study lab with a robot arm and advanced model platform. In *Proceedings of ACADIA*, page 6, 2014.

[KS13] Lukáš Kurilla and Ladislav Svoboda. Geometry optimization: Realization of a fluid-form structure composed of spherical components, fabricated by means of computer software and robotic arms. In Sigrid Brell-Çokcan and Johannes Braumann, editors, *Rob | Arch 2012*, page 184–195, Vienna, 2013. Springer.

[KSJP08] Danny M. Kaufman, Shinjiro Sueda, Doug L. James, and Dinesh K. Pai. Staggered projections for frictional contact in multibody systems. *ACM Transactions on Graphics*, 27(5):1–11, December 2008.

[KSL+19] M Kleinbort, K Solovey, Z Littlefield, K E Bekris, and D Halperin. Probabilistic Completeness of RRT for Geometric and Kinodynamic Planning With Forward Propagation. *IEEE Robotics and Automation Letters*, 4(2):x–xvi, 4 2019.

[KSLO96] L E Kavraki, P Svestka, J.-C. Latombe, and M H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[KSM+15] Oliver David Krieg, Tobias Schwinn, Achim Menges, Jian-Min Li, Jan Knippers, Annette Schmitt, and Volker Schwieger. Biomimetic lightweight timber plate shells: Computational integration of robotic fabrication, architectural geometry and structural design. In Philippe Block, Jan Knippers, Niloy J. Mitra, and Wenping Wang, editors, *Advances in Architectural Geometry 2014*, page 109–125, Cham, 2015. Springer International Publishing.

[KSW+17] Robert Kovacs, Anna Seufert, Ludwig Wall, Hsiang-Ting Chen, Florian Meinel, Willi Müller, Sijing You, Maximilian Brehm, Jonathan Striebel, Yannis Kommana, Alexander Popiak, Thomas Bläsius, and Patrick Baudisch. TrussFab: Fabricating Sturdy Large-Scale Structures on Desktop 3D Printers. In *Proceedings of the 2017*

CHI Conference on Human Factors in Computing Systems, pages 2606–2616, Denver Colorado USA, May 2017. ACM.

[KTGW14] Nathan King, Kadri Tamre, Georg Grasser, and Allison Weiler. Phase change: Approaching research methodologies through design robotics. In Wes McGee and Monica Ponce de Leon, editors, *Robotic Fabrication in Architecture, Art and Design 2014*, page 347–356. Springer International Publishing, Cham, 2014.

[KTT20] Ammar Kalo, Kenneth Tracy, and Mark Tam. Robotic sand carving: Machining techniques derived from a traditional balinese craft. In *25th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2020*, volume 2, page 443–452, Hong Kong, 2020.

[KY89] Roozbeh Kangari and Tetsuji Yoshida. Prototype robotics in construction industry. *Journal of Construction Engineering and Management*, 115(2):284–301, Jun 1989.

[KZN+20] ANDREW KUDLESS, JOSHUA ZABEL, CHUCK NAEVE, TENNA FLORIAN, JANE BURRY, JENNY SABIN, BOB SHEIL, and MARILENA SKAVARA. The design and fabrication of confluence park. In *Fabricate 2020*, Making Resilient Architecture, page 28–35. UCL Press, 2020.

[LA16] Fabien Lagriffoul and Benjamin Andres. Combining task and motion planning: A culprit detection problem. *The International Journal of Robotics Research*, 35(8):890–927, 2016.

[LA20] Niels Martin Larsen and Anders Kruse Aagaard. Robotic processing of crooked sawlogs for use in architectural construction. *Construction Robotics*, 4(1):75–83, Jun 2020.

[Lag16] Fabien Lagriffoul. On Benchmarks for Combined Task and Motion Planning. In *Robotics: Science and Systems (RSS) 2016 Workshop on Task and Motion Planning*, 2016.

[LAT+21] Pok Yin Victor Leung, Aleksandra Anna Apolinarska, Davide Tanadini, Fabio Gramazio, and Matthias Kohler. Automatic assembly of jointed timber structure using distributed robotic clamps. In *PROJECTIONS - Proceedings of the 26th CAADRIA Conference*, page 10, 2021.

[LaV98] Steven M LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998.

[LaV06] Steven M LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[LBC21] Hyung Joo Lee and Sigrid Brell-Cokcan. Cartesian coordinate control for teleoperated construction machines. *Construction Robotics*, 5(1):1–11, Mar 2021.

[LBDA21] Mackenzie Leake, Gilbert Bernstein, Abe Davis, and Maneesh Agrawala. A mathematical foundation for foundation paper pieceable quilts. *ACM Transactions on Graphics*, 40(4):1–14, August 2021.

[LBRM12] Linjie Luo, Ilya Baran, Szymon Rusinkiewicz, and Wojciech Matusik. Chopper: Partitioning models into 3D-printable parts. *ACM Transactions on Graphics*, 31(6):1–9, November 2012.

[LDB+14] Fabien Lagriffoul, Dimitar Dimitrov, Julien Bidot, Alessandro Saffiotti, and Lars Karlsson. Efficiently combining task and motion planning using geometric constraints. *International Journal of Robotics Research (IJRR)*, 2014.

[LDG+18] F. Lagriffoul, N.T. Dantam, C. Garrett, A. Akbari, S. Srivastava, and L.E. Kavraki. Platform-Independent Benchmarks for Task and Motion Planning. *IEEE Robotics and Automation Letters*, 3(4), 2018.

[Lee18] Juney Lee. *Computational Design Framework for 3D Graphic Statics*. PhD thesis, ETH Zurich, Department of Architecture, Zurich, 2018.

[LEM+17] Marco Livesu, Stefano Ellero, Jonàs Martínez, Sylvain Lefebvre, and Marco Attene. From 3D models to 3D prints: An overview of the processing pipeline. *Computer Graphics Forum*, 36(2):537–564, May 2017.

[Leu18] Pok Yin Victor Leung. Prototyping non-developable paper models without gluing tabs. In *Proceedings of the 2nd ACM Symposium on Computational Fabrication*, pages 1–11, Cambridge Massachusetts, June 2018. ACM.

[LF16] Ena Lloret Fritschi. *Smart Dynamic Casting - A digital fabrication method for non-standard concrete structures*. Doctoral thesis, ETH Zurich, 2016.

[LFY+19] Hao-Yu Liu, Xiao-Ming Fu, Chunyang Ye, Shuangming Chai, and Ligang Liu. Atlas refinement with bounded packing efficiency. *ACM Transactions on Graphics*, 38(4):1–13, August 2019.

[LFZ18] Wallace Lira, Chi-Wing Fu, and Hao Zhang. Fabricable eulerian wires for 3D shape abstraction. *ACM Transactions on Graphics*, 37(6):1–13, December 2018.

[LGC+18] Jikai Liu, Andrew T. Gaynor, Shikui Chen, Zhan Kang, Krishnan Suresh, Akihiro Takezawa, Lei Li, Junji Kato, Jinyuan Tang, Charlie

C. L. Wang, Lin Cheng, Xuan Liang, and Albert. C. To. Current and future trends in topology optimization for additive manufacturing. *Structural and Multidisciplinary Optimization*, 57(6):2457–2483, June 2018.

[LGK13] Jason Lim, Fabio Gramazio, and Matthias Kohler. A software environment for designing through robotic fabrication: Developing a graphical programming toolkit for the digital design and scaled robotic fabrication of high rises. In *Proceedings of the 18th International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA 2013)*, page 10, 2013.

[LHVBC17] Elisa Lublasser, Linda Hildebrand, Anya Vollpracht, and Sigrid Brell-Cokcan. Robot assisted deconstruction of multi-layered façade constructions on the example of external thermal insulation composite systems. *Construction Robotics*, 1(1):39–47, Dec 2017.

[Lif08] Vladimir Lifschitz. What is Answer Set Programming? In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, pages 1594–1597. AAAI Press, 2008.

[LKJ01] Steven M LaValle and James J Kuffner Jr. Randomized kinodynamic planning. *The international journal of robotics research*, 20(5):378–400, 2001.

[Lla15] Daniel Cardoso Llach. *Builders of the Vision: Software and the Imagination of Design*. Routledge, 2015.

[LLB+20] PAUL LOH, DAVID LEGGETT, JANE BURRY, JENNY SABIN, BOB SHEIL, and MARILENA SKAVARA. House #05: Incremental construction in digital practice. In *Fabricate 2020*, Making Resilient Architecture, page 280–285. UCL Press, 2020.

[LLG+21] Mackenzie Leake, Frances Lai, Tovi Grossman, Daniel Wigdor, and Ben Lafreniere. PatchProv: Supporting Improvisational Design Practices for Modern Quilting. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–17, Yokohama Japan, May 2021. ACM.

[LLZ+20] Hao-Yu Liu, Zhong-Yuan Liu, Zheng-Yu Zhao, Ligang Liu, and Xiao-Ming Fu. Practical Fabrication of Discrete Chebyshev Nets. *Computer Graphics Forum*, 39(7):13–26, 2020.

[LMF16] Juney Lee, Caitlin Mueller, and Corentin Fivet. Automatic generation of diverse equilibrium structures through shape grammars and graphic statics. *International Journal of Space Structures*, 31(2-4):147–164, June 2016.

[LMM02] Andrea Lodi, Silvano Martello, and Michele Monaci. Two-dimensional packing problems: A survey. *European Journal of Operational Research*, 141(2):241–252, Sep 2002.

[LP81] Tomás Lozano-Pérez. Automatic planning of manipulator transfer movements. *IEEE Transactions on Systems, Man, and Cybernetics*, 11:681–698, 1981.

[LPJM+87] T Lozano-Pérez, J L Jones, E Mazer, P A O'Donnell, W E L Grimson, P Tournassoud, and A Lanusse. HANDEY: A Robot System that Recognizes, Plans, and Manipulates. In *IEEE International Conference on Robotics and Automation (ICRA)*, 1987.

[LPK14] Tomás Lozano-Pérez and Leslie Pack Kaelbling. A Constraint-Based Method for Solving Sequential Manipulation Planning Problems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3684–3691. IEEE, 2014.

[LPR20] Danny Leen, Nadya Peek, and Raf Ramakers. LamiFold: Fabricating Objects with Integrated Mechanisms Using a Laser cutter Lamination Workflow. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 304–316, Virtual Event USA, October 2020. ACM.

[LPW79] Tomás Lozano-Pérez and Michael A Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10):560–570, 1979.

[LSD+16] Timothy Langlois, Ariel Shamir, Daniel Dror, Wojciech Matusik, and David I. W. Levin. Stochastic structural analysis for context-aware design and fabrication. *ACM Transactions on Graphics*, 35(6):1–13, November 2016.

[LTO+20] Aline A. S. Leao, Franklina M. B. Toledo, José Fernando Oliveira, Maria Antónia Carravilla, and Ramón Alvarez-Valdés. Irregular packing problems: A review of mathematical models. *European Journal of Operational Research*, 282(3):803–822, May 2020.

[Luk19] Mina Konaković Luković. *Computational Design of Auxetic Shells*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2019.

[LVK11] Justin Lavallee, Rachel Vroman, and Yair Keshet. Automated folding of sheet metal components with a six-axis industrial robot. In *Proceedings of ACADIA*, page 8, 2011.

[LWS+20] Yi-Heng Lu, Shih-Yuan Wang, Yu-Ting Sheng, Che-Wei Lin, Yu-Hsuan Pang, and Wei-Tse Hung. In *25th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2020*, page 10, 2020.

[LWW+19] Samuel Leder, Ramon Weber, Dylan Wood, Oliver Bucklin, and Achim Menges. Distributed robotic timber construction. In *Proceedings of ACADIA*, page 10, 2019.

[LYH+15] Sheng-Jie Luo, Yonghao Yue, Chun-Kai Huang, Yu-Huan Chung, Sei Imai, Tomoyuki Nishita, and Bing-Yu Chen. Legolization: Optimizing LEGO designs. *ACM Transactions on Graphics*, 34(6):1–12, November 2015.

[LYUI20] Maria Larsson, Hironori Yoshida, Nobuyuki Umetani, and Takeo Igarashi. Tsugite: Interactive design and fabrication of wood joints. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, page 317–327, Virtual Event USA, Oct 2020. ACM.

[LZZ+21a] Zhong-Yuan Liu, Zhan Zhang, Di Zhang, Chunyang Ye, Ligang Liu, and Xiao-Ming Fu. Modeling and fabrication with specified discrete equivalence classes. *ACM Transactions on Graphics*, 40(4):41:1–41:12, July 2021.

[LZZ+21b] Ming Lu, Xinjie Zhou, Yifan Zhou, Liming Zhang, Weiran Zhu, and Philip F. Yuan. Research on realtime communication and control workflow with vision feedback integrated in robotic fabrication. In *Proceedings of eCAADe*, page 145–152, Novi Sad, Serbia, 2021.

[Mae94] J. Maeda. Development and application of the smart system. In *Automation and Robotics in Construction Xi*, page 457–464. Elsevier, 1994.

[Mat22] Mataerial. Mataerial. http://www.mataerial.com/, 2022.

[MB16] Sina Mostafavi and Henriette Bier. Materially informed design to robotic production: A robotic 3d printing system for informed material deposition. In Dagmar Reinhardt, Rob Saunders, and Jane Burry, editors, *Robotic Fabrication in Architecture, Art and Design 2016*, page 338–349. Springer International Publishing, Cham, 2016.

[MBD20a] Ioanna Mitropoulou, Mathias Bernhard, and Benjamin Dillenburger. Print paths key-framing: Design for non-planar layered robotic fdm printing. In *Symposium on Computational Fabrication*, page 1–10, Virtual Event USA, Nov 2020. ACM.

[MBD20b] Ioanna Mitropoulou, Mathias Bernhard, and Benjamin Dillenburger. Print paths key-framing: Design for non-planar layered robotic fdm printing. In *Symposium on Computational Fabrication*, SCF '20, New York, NY, USA, 2020. Association for Computing Machinery.

[McD91] Drew McDermott. Regression planning. *International Journal of Intelligent Systems*, 6(4):357–416, 1991.

[MCHW18] Y. Ma, Z. Chen, W. Hu, and W. Wang. Packing Irregular Objects in 3D Space via Hybrid Optimization. *Computer Graphics Forum*, 37(5):49–59, August 2018.

[MDS+21] Zhao Ma, Simon Duenser, Christian Schumacher, Romana Rust, Moritz Bächer, Fabio Gramazio, Matthias Kohler, and Stelian Coros. Stylized robotic clay sculpting. *Computers & Graphics*, 98:150–164, Aug 2021.

[Mel17] Nathan; Kassabian Melenbrink. Towards force-aware robot collectives for on-site construction. In *ACADIA 2017: DISCIPLINES & DISRUPTION [Proceedings of the 37th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA) ISBN 978-0-692-96506-1] Cambridge, MA 2-4 November, 2017), pp. 382-391*. CUMINCAD, 2017.

[Men75] Max Mengeringhausen. *Raumfachwerke Aus Stäben Und Knoten: Theorie, Planung, Ausführung*. Number Bd. 1 in Komposition Im Raum. Bauverlag, Wiesbaden ; Berlin, 7., völlig neubearb. aufl edition, 1975.

[MGZ99] W McGuire, R H Gallagher, and R D Ziemian. *Matrix Structural Analysis*. Wiley, 1999.

[MHS+19] Long Ma, Ying He, Qian Sun, Yuanfeng Zhou, Caiming Zhang, and Wenping Wang. Constructing 3D Self-Supporting Surfaces with Isotropic Stress Using 4D Minimal Hypersurfaces of Revolution. *ACM Transactions on Graphics*, 38(5):1–13, November 2019.

[MIG+14] Stefanie Mueller, Sangha Im, Serafima Gurevich, Alexander Teibrich, Lisa Pfisterer, François Guimbretière, and Patrick Baudisch. WirePrint: 3D printed previews for fast prototyping. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 273–280. ACM, 2014.

[Mir16] Ammar Mirjan. *Aerial Construction: Robotic Fabrication of Tensile Structures with Flying Machines*. PhD thesis, ETH Zurich, 2016.

[MKB] Stefanie Mueller, Bastian Kruck, and Patrick Baudisch. LaserOrigami: Laser-Cutting 3D Objects. page 8.

[MLB16] Eder Miguel, Mathias Lepoutre, and Bernd Bickel. Computational design of stable planar-rod structures. *ACM Transactions on Graphics*, 35(4):86:1–86:11, July 2016.

[MLS+18] Alessandro Muntoni, Marco Livesu, Riccardo Scateni, Alla Sheffer, and Daniele Panozzo. Axis-Aligned Height-Field Block Decomposition of 3D Shapes. *ACM Transactions on Graphics*, 37(5):1–15, October 2018.

[MMZ18] Shiran Magrisso, Moran Mizrahi, and Amit Zoran. Digital joinery for hybrid carpentry. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 1–11, Montreal QC Canada, Apr 2018. ACM.

[MNP19] Wes McGee, Tsz Yan Ng, and Asa Peller. Hard + soft: Robotic needle felting for nonwoven textiles. In Jan Willmann, Philippe Block, Marco Hutter, Kendra Byrne, and Tim Schork, editors, *Robotic Fabrication in Architecture, Art and Design 2018*, page 192–204, Cham, 2019. Springer International Publishing.

[MO13] Caitlin Mueller and John Ochsendorf. An Integrated Computational Approach for Creative Conceptual Structural Design. *Proceedings of IASS Annual Symposia*, 2013(12):1–6, September 2013.

[MO15] Caitlin T. Mueller and John A. Ochsendorf. Combining structural performance and designer preferences in evolutionary design space exploration. *Automation in Construction*, 52:70–82, April 2015.

[MOOS00] Hiroshi Miyakawa, Jyunichi Ochiai, Katsuyuki Oohata, and Takashi Shiokawa. Application of automated building construction system for high-rise office building. In *Proceedings of ISARC*, Taipei, Taiwan, Sep 2000.

[MPBC16] Luigi Malomo, Nico Pietroni, Bernd Bickel, and Paolo Cignoni. FlexMolds: Automatic design of flexible shells for molding. *ACM Transactions on Graphics*, 35(6):1–12, November 2016.

[MS08] Michael Meredith and Mutsuro Sasaki. *From control to design: parametric/algorithmic architecture*. Actar-D, 2008.

[MSK16] Achim Menges, Tobias Schwinn, and Oliver David Krieg. *Advancing wood architecture*. Taylor & Francis, 2016.

[MSM11] James McCrae, Karan Singh, and Niloy J. Mitra. Slices: A shape-proxy based on planar sections. In *Proceedings of the 2011 SIGGRAPH Asia Conference on - SA '11*, page 1, Hong Kong, China, 2011. ACM Press.

[MSY+15] Nicolas Mellado, Peng Song, Xiaoqi Yan, Chi-Wing Fu, and Niloy J. Mitra. Computational design and construction of notch-free reciprocal frame structures. In Philippe Block, Jan Knippers, Niloy J. Mitra, and Wenping Wang, editors, *Advances in Architectural Geometry 2014*, page 181–197, Cham, 2015. Springer International Publishing.

[Mue14] Caitlin T. Mueller. *Computational Exploration of the Structural Design Space*. Thesis, Massachusetts Institute of Technology, 2014.

[MUS14] James McCrae, Nobuyuki Umetani, and Karan Singh. FlatFitFab: Interactive modeling with planar sections. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, pages 13–22, Honolulu Hawaii USA, October 2014. ACM.

[MVTT17] Wes Mcgee, Kathy Velikov, Geoffrey Thun, and DAN TISH. Infundibuliforms:: kinetic systems, additive manufacturing for cable nets and tensile surface control. In *Fabricate 2017*, page 84–91. UCL Press, Apr 2017.

[MX320] MX3D. MX3D | Connector for Takenaka. https://mx3d.com/industries/construction/connector-for-takenaka/, 2020.

[MX322] MX3D. Mx3d | robotic 3d metal printing. https://mx3d.com/, 2022.

[NAH+18] Vidya Narayanan, Lea Albaugh, Jessica Hodgins, Stelian Coros, and James Mccann. Automatic Machine Knitting of 3D Meshes. *ACM Transactions on Graphics*, 37(3):1–15, August 2018.

[NAI+18] Kazutaka Nakashima, Thomas Auzinger, Emmanuel Iarussi, Ran Zhang, Takeo Igarashi, and Bernd Bickel. CoreCavity: Interactive shell decomposition for fabrication with two-piece rigid molds. *ACM Transactions on Graphics*, 37(4):1–13, August 2018.

[NCSS21] Paul Nicholas, Ruxandra Stefania Chiujdea, Konrad Sonne, and Antonio Scaffidi. Design and fabrication methodologies for repurposing end of life metal via robotic incremental sheet metal forming. In *Proceedings of the 39th eCAADe Conference*, page 10, 2021.

[Nil14] Nils J Nilsson. *Principles of artificial intelligence*. Morgan Kaufmann, 2014.

[NKK84] Dana S. Nau, Vipin Kumar, and Laveen Kanal. General Branch and Bound, and its relation to A_* and AO_*. *Artificial Intelligence*, 23(1):29–58, May 1984.

[NKKS21] Roberto Naboni, Anja Kunic, Aljaz Kramberger, and Christian Schlette. Design, simulation and robotic assembly of reversible timber structures. *Construction Robotics*, 5(1):13–22, Mar 2021.

[NS+72] Allen Newell, Herbert Alexander Simon, et al. *Human problem solving*, volume 104. Prentice-hall Englewood Cliffs, NJ, 1972.

[NSN+16] Paul Nicholas, David Stasiuk, Esben Nørgaard, Christopher Hutchinson, and Mette Ramsgaard Thomsen. An integrated modelling and toolpathing approach for a frameless stressed skin structure, fabricated using robotic incremental sheet forming. In Dagmar Reinhardt, Rob Saunders, and Jane Burry, editors, *Robotic Fabrication in Architecture, Art and Design 2016*, page 62–77. Springer International Publishing, Cham, 2016.

[NTM+21] S. Nuvoli, A. Tola, A. Muntoni, N. Pietroni, E. Gobbetti, and R. Scateni. Automatic Surface Segmentation for Seamless Fabrication Using 4-axis Milling Machines. *Computer Graphics Forum*, 40(2):191–203, 2021.

[NW99] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[NZN+17] Paul Nicholas, Mateusz Zwierzycki, Esben Clausen Nørgaard, David Stasiuk, and Mette Thomsen. Adaptive robotic fabrication for conditions of material inconsistency: increasing the geometric accuracy of incrementally formed metal panels. In *Fabricate 2017*, pages 114–121. UCL Press, 2017.

[Och02] John Allen Ochsendorf. *Collapse of masonry structures*. PhD thesis, University of Cambridge, Jul 2002.

[ODM21] Bryan W X Ong, Renaud Danhaive, and Caitlin Mueller. Machine learning for human design: Sketch interface for structural morphology ideation using neural networks. In *Proceedings of the International Association for Shell and Spatial Structures (IASS) Symposium*, page 13, 2021.

[OLK+13] Neri Oxman, Jared Laucks, Markus Kayser, Elizabeth Tsai, and Michal Firstenberg. Freeform 3D printing: Towards a sustainable approach to additive manufacturing. *Green Design, Materials and Manufacturing Processes*, 479, 2013.

[OOS16] Patrick Ole Ohlbrock and Joseph Schwartz. Combinatorial equilibrium modeling. *International Journal of Space Structures*, 31(2-4):177–189, June 2016.

[ORM+19] R. Oval, M. Rippmann, R. Mesnil, T. Van Mele, O. Baverel, and P. Block. Feature-based topology finding of patterns for shell structures. *Automation in Construction*, 103:185–201, July 2019.

[Pag17] Mitchell Page. An accessible & bespoke digital fabrication process for robotically- milled dovetail & finger joints. In *Proceedings of the 37th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, page 8, 2017.

[Par19] Stefana Parascho. *Cooperative Robotic Assembly: Computational Design and Robotic Fabrication of Spatial Metal Structures*. Doctoral Thesis, ETH Zurich, 2019.

[Paw13] Tom Pawlofsky. Rhino2krl: A simple cad to robot interface for fast process prototyping. In Sigrid Brell-Çokcan and Johannes Braumann, editors, *Rob | Arch 2012*, page 167–172, Vienna, 2013. Springer.

[Pay11] Andrew Payne. A five-axis robotic motion controller for designers. In *Proceedings of ACADIA*, page 162–169, Banff (Alberta), Canada, 2011.

[PBS13] Daniele Panozzo, Philippe Block, and Olga Sorkine-Hornung. Designing unreinforced masonry models. *ACM Transactions on Graphics*, 32(4):1–12, July 2013.

[PBSM16] Paul Poinet, Ehsan Baharlou, Tobias Schwinn, and Achim Menges. Adaptive pneumatic shell structures - feedback-driven robotic stiffening of inflated extensible membranes and further rigidification for architectural applications. In *Proceedings of eCAADe*, page 549–558, Oulu, Finland, 2016.

[PDKG96] G. Pritschow, M. Dalacker, J. Kurz, and M. Gaenssle. Technological aspects in the development of a mobile bricklaying robot. *Automation in Construction*, 5(1):3–13, Mar 1996.

[PDS+14] Marshall Prado, Moritz Dörstelmann, Tobias Schwinn, Achim Menges, and Jan Knippers. Core-less filament winding. In Wes McGee and Monica Ponce de Leon, editors, *Robotic Fabrication in Architecture, Art and Design 2014*, page 275–289. Springer International Publishing, Cham, 2014.

[PE17] Samuel Pietri and Alessio Erioli. Study of spiderweb strategies for the design of architectural envelopes using swarms of drones and inflatable formworks. In *Proceedings of eCAADe*, page 10, 2017.

[Pee16] Nadya (Nadya Meile) Peek. *Making Machines That Make : Object-Oriented Hardware Meets Object-Oriented Software*. PhD thesis, Massachusetts Institute of Technology, 2016.

[PEVW15] Helmut Pottmann, Michael Eigensatz, Amir Vaxman, and Johannes Wallner. Architectural geometry. *Computers & Graphics*, 47:145–164, April 2015.

[PFL+16] Wiktor Mateusz Piotrowski, Maria Fox, Derek Long, Daniele Magazzeni, and Fabio Mercorio. Heuristic planning for pddl+ domains. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[PGM+17] Stefana Parascho, Augusto Gandia, Ammar Mirjan, Fabio Gramazio, and Matthias Kohler. Cooperative fabrication of spatial metal structures. In *Fabricate 2017*, pages 24–29. UCL Press, 2017.

[PH10] E Plaku and G D Hager. Sampling-Based Motion and Symbolic Action Planning with geometric and differential constraints. In *2010 IEEE International Conference on Robotics and Automation*, pages 5002–5008, 5 2010.

[PHW+20] Stefana Parascho, Isla Xi Han, Samantha Walker, Alessandro Beghini, Edvard P. G. Bruun, and Sigrid Adriaenssens. Robotic vault: a cooperative robotic assembly method for brick vault construction. *Construction Robotics*, 4(3):117–126, Dec 2020.

[Pia14] Marta; Karaivanov Piaseczynska. Vertiguous interiors. In *Proceedings of the 34th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, 2014.

[PIC+21] Julian Panetta, Florin Isvoranu, Tian Chen, Emmanuel Siéfert, Benoît Roman, and Mark Pauly. Computational inverse design of surface-based inflatables. *ACM Transactions on Graphics*, 40(4):40:1–40:14, July 2021.

[Pik22] Daniel Piker. Kangaroo Physics. https://www.food4rhino.com/app/kangaroo-physics, 2022.

[PJP+19] Luka Piškorec, David Jenny, Stefana Parascho, Hannes Mayer, Fabio Gramazio, and Matthias Kohler. The brick labyrinth. In Jan Willmann, Philippe Block, Marco Hutter, Kendra Byrne, and Tim Schork, editors, *Robotic Fabrication in Architecture, Art and Design 2018*, page 489–500, Cham, 2019. Springer International Publishing.

[PKC+18] Stefana Parascho, Thomas Kohlhammer, Stelian Coros, Fabio Gramazio, and Matthias Kohler. Computational Design of Robotically Assembled Spatial Structures: A sequence based method for the generation and evaluation of structures fabricated with cooperating robots. In *AAG 2018: Advances in Architectural Geometry*

2018, pages 112–139. Klein Publishing, 2018. Accepted: 2018-10-29T05:56:52Z.

[PKD+15] Stefana Parascho, Jan Knippers, Moritz Dörstelmann, Marshall Prado, and Achim Menges. Modular fibrous morphologies: Computational design, simulation and fabrication of differentiated fibre composite building components. In Philippe Block, Jan Knippers, Niloy J. Mitra, and Wenping Wang, editors, *Advances in Architectural Geometry 2014*, page 29–45, Cham, 2015. Springer International Publishing.

[PKI+19] J. Panetta, M. Konaković-Luković, F. Isvoranu, E. Bouleau, and M. Pauly. X-Shells: A new class of deployable beam structures. *ACM Transactions on Graphics*, 38(4):1–15, July 2019.

[PM11] David Pigram and Wes McGee. Formation embedded design: A methodology for the integration of fabrication constraints into architectural design. In *Proceedings of ACADIA*, page 122–131, Banff (Alberta), Canada, 2011.

[PM20a] Daniel Piker and Richard Maddock. Continuous robotic spatial 3d printing of topologically irregular space frames. In Christoph Gengnagel, Olivier Baverel, Jane Burry, Mette Ramsgaard Thomsen, and Stefan Weinzierl, editors, *Impact: Design With All Senses, Proceedings of Design Modeling Symposium*, page 502–516, Cham, 2020. Springer International Publishing.

[PM20b] Daniel Piker and Richard Maddock. Continuous Robotic Spatial 3D Printing of Topologically Irregular Space Frames. In Christoph Gengnagel, Olivier Baverel, Jane Burry, Mette Ramsgaard Thomsen, and Stefan Weinzierl, editors, *Impact: Design With All Senses*, pages 502–516, Cham, 2020. Springer International Publishing.

[PMM+13] David Pigram, Iain Maxwell, Wes McGee, Ben Hagenhofer-Daniell, and Lauren Vasey. Protocols, pathways, and production. In Sigrid Brell-Çokcan and Johannes Braumann, editors, *Rob | Arch 2012*, page 143–148, Vienna, 2013. Springer.

[Pou19] Ebrahim Poustinchi. Robotically augmented imaging (rai alpha). In *Proceedings of ACADIA*, page 352–359, Austin (Texas), USA, 2019.

[Pre13] Clemens Preisinger. Linking structure and parametric geometry. *Architectural Design*, 83(2):110–113, 2013.

[PSR21] Jens Pedersen, Asbjørn Søndergaard, and Dagmar Reinhardt. Hand-drawn digital fabrication: calibrating a visual communication method for robotic on-site fabrication. *Construction Robotics*, 5(2):159–173, Jun 2021.

[PSSA12] Amit Kumar Pandey, Jean-Philippe Saut, Daniel Sidobre, and Rachid Alami. Towards Planning Human-Robot Interactive Manipulation Tasks: Task Dependent and Human Oriented Autonomous Selection of Grasp and Placement. In *RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2012.

[PWMG16] Huaishu Peng, Rundong Wu, Steve Marschner, and François Guimbretière. On-the-fly print: Incremental printing while modelling. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 887–896. ACM, 2016.

[PZM+15] Julian Panetta, Qingnan Zhou, Luigi Malomo, Nico Pietroni, Paolo Cignoni, and Denis Zorin. Elastic textures for additive fabrication. *ACM Transactions on Graphics*, 34(4):1–12, July 2015.

[RA15] Ronald Richter and Marc Alexa. Beam meshes. *Computers & Graphics*, 53:28–36, December 2015.

[RAP14] Felix Raspall, Felix Amtsberg, and Stefan Peters. Material feedback in robotic production: Plastic and elastic formation of materials for reusable mold-making. In Wes McGee and Monica Ponce de Leon, editors, *Robotic Fabrication in Architecture, Art and Design 2014*, page 333–345. Springer International Publishing, Cham, 2014.

[RCJW16] Dagmar Reinhardt, Densil Cabrera, Alexander Jung, and Rod Watt. Towards a micro design of acoustic surfaces. In Dagmar Reinhardt, Rob Saunders, and Jane Burry, editors, *Robotic Fabrication in Architecture, Art and Design 2016*, page 136–149. Springer International Publishing, Cham, 2016.

[RCP+18] R. Rust, G. Casas, S. Parascho, D. Jenny, K. Dörfler, M. Helmreich, A. Gandia, Z. Ma, I. Ariza, M. Pacher, B. Lytle, and Y. Huang. COMPAS FAB: Robotic fabrication package for the compas framework. https://github.com/compas-dev/compas_fab/, 2018. Gramazio Kohler Research, ETH Zürich.

[RH09] Silvia Richter and Malte Helmert. Preferred Operators and Deferred Evaluation in Satisficing Planning. In *ICAPS*, 2009.

[RKP+22] Yingying Ren, Uday Kusupati, Julian Panetta, Florin Isvoranu, Davide Pellis, Tian Chen, and Mark Pauly. Umbrella Meshes: Elastic Mechanisms for Freeform Shape Deployment. *ACM Transactions on Graphics*, 41(4):15, 2022.

[RLB12] Matthias Rippmann, Lorenz Lachauer, and Philippe Block. RhinoVAULT - Interactive Vault Design. *International Journal of Space Structures*, 27(4):219–230, December 2012.

[RLW21] Nicolas Rogeau, Pierre Latteur, and Yves Weinand. An integrated design tool for timber plate structures to generate joints geometry, fabrication toolpath, and robot trajectories. *Automation in Construction*, 130:103875, Oct 2021.

[RMS17] David Rosenwasser, Sonya Mantell, and Jenny Sabin. Robotic fabrication and digital ceramics. In *Proceedings of Association for Computer Aided Design in Architecture*, page 10, 2017.

[RN16] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[RN18] Gabriella Rossi and Paul Nicholas. Re/learning the wheel. methods to utilize neural networks as design tools for doubly curved metal surfaces. In *Proceedings of ACADIA*, page 146–155, Mexico City, Mexico, 2018.

[Rob22] Robert McNeel & Associates. Rhino3D®and Grasshopper3D®. https://www.rhino3d.com, 2022.

[Ros22] Andrea Rossi. Wasp - Discrete Design for Grasshopper. `https://github.com/ar0551/Wasp`, June 2022.

[RPC+21] Yingying Ren, Julian Panetta, Tian Chen, Florin Isvoranu, Samuel Poincloux, Christopher Brandt, Alison Martin, and Mark Pauly. 3D weaving with curved ribbons. *ACM Transactions on Graphics*, 40(4):1–15, August 2021.

[RPS+13] Mostapha Sadeghipour Roudsari, Michelle Pak, Adrian Smith, et al. Ladybug: a parametric environmental plugin for grasshopper to help designers create an environmentally-conscious design. In *Proceedings of the 13th international IBPSA conference held in Lyon, France Aug*, pages 3128–3135, 2013.

[RRS13] Bernhard Reinert, Tobias Ritschel, and Hans-Peter Seidel. Interactive by-example design of artistic packing layouts. *ACM Transactions on Graphics*, 32(6):1–7, November 2013.

[RTB+19] Dagmar Reinhardt, Ninotschka Titchkosky, Chris Bickerton, Rodney Watt, Dylan Wozniak-O'Connor, Christhina Candido, Densil Cabrera, Mitchell Page, and Sascha Bohnenberger. Towards onsite, modular robotic carbon-fibre winding for an integrated ceiling structure. *Construction Robotics*, 3(1):23–40, Dec 2019.

[Rus17] Romana Rust. *Spatial Wire Cutting: Integrated Design, Simulation and Force-adaptive Fabrication of Double Curved Formwork Components*. Doctoral thesis, ETH Zurich, 2017.

[RWH+17] Christopher Robeller, Yves Weinand, Volker Helm, Andreas Thoma, Fabio Gramazio, and Matthias Kohler. Robotic integral attachment. In *Fabricate 2017*, pages 92–97. UCL Press, 2017.

[RWS+21] Gabriella Rossi, James Walker, Asbjørn Søndergaard, Isak Worre Foged, Anke Pasold, and Jacob Hilmer. Oscillating wire cutting and robotic assembly of bespoke acoustic tile systems. *Construction Robotics*, 5(1):63–72, Mar 2021.

[RZBS09] Nathan Ratliff, Matthew Zucker, J Andrew Bagnell, and Siddhartha Srinivasa. CHOMP: Gradient optimization techniques for efficient motion planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 489–494. IEEE, 2009.

[SAE+16] Asbjorn Søndergaard, Oded Amir, Phillip Eversmann, Luka Piškorec, Florin Stan, Fabio Gramazio, and Matthias Kohler. Topology optimization and robotic fabrication of advanced timber space-frame structures. In *Robotic Fabrication in Architecture, Art and Design 2016*, pages 190–203. Springer, 2016.

[San18] Timothy Sandy. *High-Accuracy Mobile Manipulation for On-Site Robotic Building Construction*. Doctoral thesis, ETH Zurich, 2018. Accepted: 2018-12-03T10:15:23Z.

[Sas07] Lawrence Sass. Synthesis of design production with integrated digital fabrication. *Automation in construction*, 16(3):298–310, 2007.

[SAS17] Shani Sharif, Varun Agrawal, and Larry Sweet. Adaptive industrial robot control for designers. In *Proceedings of eCAADe*, page 8, 2017.

[SB12] Eftychios Sifakis and Jernej Barbic. Fem simulation of 3d deformable solids: a practitioner's guide to theory, discretization and model reduction. In *ACM SIGGRAPH 2012 courses*, pages 1–50. 2012.

[SB19] Sven Stumm and Sigrid Brell-Çokcan. Haptic Programming. In Jan Willmann, Philippe Block, Marco Hutter, Kendra Byrne, and Tim Schork, editors, *Robotic Fabrication in Architecture, Art and Design 2018*, pages 44–58, Cham, 2019. Springer International Publishing.

[SBB+16] Gregor Steinhagen, Johannes Braumann, Jan Brüninghaus, Matthias Neuhaus, Sigrid Brell-Cokcan, and Bernd Kuhlenkötter. Path planning for robotic artistic stone surface production. In Dagmar Reinhardt, Rob Saunders, and Jane Burry, editors, *Robotic Fabrication in Architecture, Art and Design 2016*, page 122–135. Springer International Publishing, Cham, 2016.

[SBR+15] Christian Schumacher, Bernd Bickel, Jan Rys, Steve Marschner, Chiara Daraio, and Markus Gross. Microstructures to control elasticity in 3D printing. *ACM Transactions on Graphics*, 34(4):1–13, July 2015.

[SBSS20] Tom Shaked, Karen Lee Bar-Sinai, and Aaron Sprecher. Craft to site: Middleware for in situ robotic stonework. *Construction Robotics*, 4(3):141–150, Dec 2020.

[SBT16] Anton Savov, Ben Buckton, and Oliver Tessmann. 20,000 blocks: Can gameplay be used to guide non-expert groups in creating architecture? In *Proceedings of ACADIA*, page 24–33, Ann Arbor (Michigan), USA, 2016.

[SC99] Kristina Shea and Jonathan Cagan. Languages and semantics of grammatical discrete structures. *AI EDAM*, 13(4):241–251, September 1999.

[SC18] Ioan A Sucan and Sachin Chitta. Moveit! `http://moveit.ros.org`, 2018.

[Sch74] H. J. Schek. The force density method for form finding and computation of general networks. *Computer Methods in Applied Mechanics and Engineering*, 3(1):115–134, January 1974.

[Sch12] Thibault Schwartz. HAL: Extension of a visual programming language to support teaching and research on robotics applied to construction. In *Robotic Fabrication in Architecture, Art and Design 2012*, pages 92–101. Springer, 2012.

[Sch15] Thibault Schwartz. A generic communication library for human-robot interaction on construction sites. In Mette Ramsgaard Thomsen, Martin Tamke, Christoph Gengnagel, Billie Faircloth, and Fabian Scheurer, editors, *Modelling Behaviour: Design Modelling Symposium 2015*, page 341–351. Springer International Publishing, Cham, 2015.

[Sch20] Wolfgang Schwarzmann. Traditional knowledge on modern milling robots: How cnc-joinery machines promote a renaissance to lost techniques in the profession of a carpenter. In *Proceedings of the 38th eCAADe Conference*, page 8, Berlin, 2020.

[SDH+14] John Schulman, Yan Duan, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, Jia Pan, Sachin Patil, Ken Goldberg, and Pieter Abbeel. Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research*, 33(9):1251–1270, 2014.

[SDW+16] Peng Song, Bailin Deng, Ziqi Wang, Zhichao Dong, Wei Li, Chi-Wing Fu, and Ligang Liu. CofiFab: Coarse-to-fine Fabrication of Large 3D Objects. *ACM Trans. Graph.*, 35(4):45:1–45:11, July 2016.

[SE16] Andrew Saunders and Gregory Epps. Robotic lattice smock. In Dagmar Reinhardt, Rob Saunders, and Jane Burry, editors, *Robotic Fabrication in Architecture, Art and Design 2016*, page 78–91. Springer International Publishing, Cham, 2016.

[Sen08] Richard Sennett. *The craftsman*. Yale University Press, 2008.

[SF94] Daniel Sabin and Eugene C Freuder. Contradicting conventional wisdom in constraint satisfaction. In *International Workshop on Principles and Practice of Constraint Programming*, pages 10–20. Springer, 1994.

[SF17] Asbjørn Søndergaard and Jelle Feringa. Scaling architectural robotics: construction of the kirk kapital headquarters. In *Fabricate 2017*, page 264–271. UCL Press, 2017.

[SFCO12] Peng Song, Chi-Wing Fu, and Daniel Cohen-Or. Recursive interlocking puzzles. *ACM Transactions on Graphics (TOG)*, 31(6):128, 2012.

[SFJ+17] Peng Song, Chi-Wing Fu, Yueming Jin, Hongfei Xu, Ligang Liu, Pheng-Ann Heng, and Daniel Cohen-Or. Reconfigurable interlocking furniture. *ACM Transactions on Graphics*, 36(6):1–14, November 2017.

[SFLF15] Peng Song, Zhongqi Fu, Ligang Liu, and Chi-Wing Fu. Printing 3D objects with interlocking parts. *Computer Aided Geometric Design*, 35–36:137–148, May 2015.

[SFN+16] Asbjørn Søndergaard, Jelle Feringa, Toke Nørbjerg, Kasper Steenstrup, David Brander, Jens Graversen, Steen Markvorsen, Andreas Bærentzen, Kiril Petkov, Jesper Hattel, Kenn Clausen, Kasper Jensen, Lars Knudsen, and Jacob Kortbek. Robotic hot-blade cutting. In Dagmar Reinhardt, Rob Saunders, and Jane Burry, editors, *Robotic Fabrication in Architecture, Art and Design 2016*, page 150–164. Springer International Publishing, Cham, 2016.

[SFR+14] Siddharth Srivastava, Eugene Fang, Lorenzo Riano, Rohan Chitnis, Stuart Russell, and Pieter Abbeel. Combined Task and Motion Planning Through an Extensible Planner-Independent Interface Layer. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[SFSM18] Asbjørn Søndergaard, Jelle Feringa, Florin Stan, and Dana Maier. Robotic abrasive wire cutting of polymerized styrene formwork systems for cost-effective realization of topology-optimized concrete structures. *Construction Robotics*, 2(1):81–92, Dec 2018.

[SGC18] Oded Stein, Eitan Grinspun, and Keenan Crane. Developability of triangle meshes. *ACM Transactions on Graphics*, 37(4):1–14, August 2018.

[SHWP09] Alexander Schiftner, Mathias Höbinger, Johannes Wallner, and Helmut Pottmann. Packing circles and spheres on surfaces. *ACM Transactions on Graphics*, 28(5):1–8, December 2009.

[Sim16] Robert Simunic. Powermill robot: Simplifying the complex issue of robotic programming. In Dagmar Reinhardt, Rob Saunders, and Jane Burry, editors, *Robotic Fabrication in Architecture, Art and Design 2016*, page 468–472. Springer International Publishing, Cham, 2016.

[SIS21] Madlaina Signer, Alexandra Ion, and Olga Sorkine-Hornung. Developable Metamaterials: Mass-fabricable Metamaterials by Laser-Cutting Elastic Structures. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–13, Yokohama Japan, May 2021. ACM.

[SJ11] G Suwała and Ł Jankowski. A benchmark for identification of structural modifications and inelastic impacts: The structure, test data and an example solution. In *Proceedings of the 5th ECCOMAS Thematic Conference on Smart Structures and Materials (SMART 2011)*, 2011.

[SJ16] Roland Snooks and Gwyllim Jahn. Closeness: On the relationship of multi-agent algorithms and robotic fabrication. In Dagmar Reinhardt, Rob Saunders, and Jane Burry, editors, *Robotic Fabrication in Architecture, Art and Design 2016*, page 218–229. Springer International Publishing, Cham, 2016.

[SJ19] Kyle Schumann and Ryan Luke Johns. Airforming - adaptive robotic molding of freeform surfaces through incremental heat and variable pressure. In *Proceedings of CAADRIA*, page 33–42, Wellington, New Zealand, 2019.

[SJG19] Oded Stein, Alec Jacobson, and Eitan Grinspun. Interactive design of castable shapes using two-piece rigid molds. *Computers & Graphics*, 80:51–62, May 2019.

[SK08] Mike Stilman and James J Kuffner. Planning Among Movable Obstacles with Artificial Constraints. *The International Journal of Robotics Research*, 27(11-12):1295–1307, 2008.

[SKM16] Tobias Schwinn, Oliver David Krieg, and Achim Menges. A textile approach towards the computational design and fabrication of lightweight timber shells. In *Proceedings of ACADIA*, page 10, 2016.

[SLCS04] Thierry Siméon, Jean-Paul Laumond, Juan Cortés, and Anis Sahbani. Manipulation planning with probabilistic roadmaps. *International Journal of Robotics Research (IJRR)*, 23(7-8):729–746, 2004.

[SLU17] Alvise Simondetti, Chris Luebkeman, and Gereon Uerz. 2060: An autonomously crafted built environment. *Architectural Design*, 87(4):120–127, 2017.

[SM78] G Stiny and W J Mitchell. The Palladian Grammar. *Environment and Planning B: Planning and Design*, 5(1):5–18, June 1978.

[SM14] Thibault Schwartz and Lucia Mondardini. Integrated design and robotized prototyping of abeille's vaults. In Wes McGee and Monica Ponce de Leon, editors, *Robotic Fabrication in Architecture, Art and Design 2014*, page 199–209. Springer International Publishing, Cham, 2014.

[SMDAV16] José Pedro Sousa, Pedro Martins, and Pedro De Azambuja Varela. The corkcrete arch project: The digital design and robotic fabrication of a novel building system made out of cork and glass-fibre reinforced concrete. page 735–744, Melbourne, Australia, 2016.

[SMK12] Ioan A Sucan, Mark Moll, and Lydia E Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 12 2012.

[SNF+17] Philipp S Schmitt, Werner Neubauer, Wendelin Feiten, Kai M Wurm, Georg V Wichert, and Wolfram Burgard. Optimal, sampling-based manipulation planning. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3426–3432. IEEE, 2017.

[Sol22a] Solemma LLC. Climate Studio. https://www.solemma.com/climatestudio, 2022.

[Sol22b] Vicente Soler. robots. https://github.com/visose/Robots, Jun 2022.

[SP13a] Mathew Schwartz and Jason Prasad. Robosculpt. In Sigrid Brell-Çokcan and Johannes Braumann, editors, *Rob | Arch 2012*, page 230–237, Vienna, 2013. Springer.

[SP13b] Yuliy Schwartzburg and Mark Pauly. Fabrication-aware Design with Intersecting Planar Pieces. In *Computer Graphics Forum*, volume 32, pages 317–326. Wiley Online Library, 2013.

[SPG+16] Christian Schüller, Daniele Panozzo, Anselm Grundhöfer, Henning Zimmer, Evgeni Sorkine, and Olga Sorkine-Hornung. Computational thermoforming. *ACM Transactions on Graphics*, 35(4):1–9, July 2016.

[SPM+16] José Pedro Sousa, Cristina Gassó Palop, Eduardo Moreira, Andry Maykol Pinto, José Lima, Paulo Costa, Pedro Costa, Germano Veiga, and A. Paulo Moreira. The spiderobot: A cable-robot system for on-site construction in architecture. In Dagmar Reinhardt, Rob Saunders, and Jane Burry, editors, *Robotic Fabrication in Architecture, Art and Design 2016*, page 230–239. Springer International Publishing, Cham, 2016.

[SPS18] Christian Schüller, Roi Poranne, and Olga Sorkine-Hornung. Shape representation by zippables. *ACM Transactions on Graphics*, 37(4):1–13, August 2018.

[SPV+16] Hijung V. Shin, Christopher F. Porst, Etienne Vouga, John Ochsendorf, and Frédo Durand. Reconciling elastic and equilibrium methods for static analysis. *ACM Transactions on Graphics*, 35(2):1–16, May 2016.

[SRJG17] Vicente Soler, Gilles Retsin, and Manuel Jimenez Garcia. A Generalized Approach to Non-Layered Fused Filament Fabrication. *Proceedings of the 36st Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, pages 562–571, 2017.

[SRLL20] Kristina Schinegger, Stefan Rutzinger, Johannes Ladinig, and Meizi Li. Becoming structure. In Christoph Gengnagel, Olivier Baverel, Jane Burry, Mette Ramsgaard Thomsen, and Stefan Weinzierl, editors, *Impact: Design With All Senses*, *Proceedings of Design Modeling Symposium*, page 214–223, Cham, 2020. Springer International Publishing.

[SRP05] Kenneth Sörensen, M Reimann, and C Prins. Permutation distance measures for memetic algorithms with population management. In *Proceedings of MIC, 2005, 6th Metaheuristics International Conference, Vienna, Austria, August 22-26, 2005*, 2005.

[SS07] Tommaso Schiavinotto and Thomas Stützle. A review of metrics on permutations for search landscape analysis. *Computers & Operations Research*, 34(10):3143–3153, October 2007.

[SSKA07] Mike Stilman, Jan-Ulrich Schamburek, James J Kuffner, and Tamim Asfour. Manipulation Planning Among Movable Obstacles. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2007.

[Ste02] Rolf Steinmetzger. Neue baumaschinen-generationen mit einsatz modernster kommunikationstechnik. 2002.

[Sti77] G Stiny. Ice-Ray: A Note on the Generation of Chinese Lattice Designs. *Environment and Planning B: Planning and Design*, 4(1):89–98, June 1977.

[Sti10] Mike Stilman. Global manipulation planning in robot joint space with task constraints. *IEEE Transactions on Robotics*, 26(3):576–584, 2010.

[STK+14] Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus Gross. Designing Inflatable Structures. *ACM Trans. Graph.*, 33(4):63:1–63:10, July 2014.

[STTP14] Yuliy Schwartzburg, Romain Testuz, Andrea Tagliasacchi, and Mark Pauly. High-contrast computational caustic design. *ACM Transactions on Graphics*, 33(4):1–11, July 2014.

[SVM+17] Martin Self, Emmanuel Vercruysse, Achim Menges, Bob Sheil, Ruairi Glynn, and Marilena Skavara. Infinite variations, radical strategies. In *Fabricate 2017*, page 30–35. UCL Press, 2017.

[Sza20] Anna Szabo. *Design and Fabrication of Thin Folded Members with Digital Concrete Processes*. Doctoral thesis, ETH Zurich, Nov 2020.

[SZB18] Christian Schumacher, Jonas Zehnder, and Moritz Bächer. Set-in-stone: Worst-case optimization of structures weak in tension. *ACM Transactions on Graphics*, 37(6):1–13, December 2018.

[TAH+18] Andreas Thoma, Arash Adel, Matthias Helmreich, Thomas Wehrle, Fabio Gramazio, and Matthias Kohler. Robotic fabrication of bespoke timber frame modules. In *Robotic Fabrication in Architecture, Art and Design*, pages 447–458. Springer, 2018.

[TAH+19] Andreas Thoma, Arash Adel, Matthias Helmreich, Thomas Wehrle, Fabio Gramazio, and Matthias Kohler. Robotic fabrication of bespoke timber frame modules. In Jan Willmann, Philippe Block, Marco Hutter, Kendra Byrne, and Tim Schork, editors, *Robotic Fabrication in Architecture, Art and Design 2018*, page 447–458, Cham, 2019. Springer International Publishing.

[TAST18] Marc A Toussaint, Kelsey Rebecca Allen, Kevin A Smith, and Joshua B Tenenbaum. Differentiable physics and stable modes for tool-use and manipulation planning. In *Robotics: Science and Systems (RSS)*, 2018.

[TBM16] Stavros Tseranidis, Nathan C. Brown, and Caitlin T. Mueller. Data-driven approximation algorithms for rapid performance evaluation and optimization of civil structures. *Automation in Construction*, 72:279–293, December 2016.

[TCFM16] Kam-Ming Mark Tam, James R. Coleman, Nicholas W. Fine, and Caitlin T. Mueller. Robotics-enabled stress line additive manufacturing. In Dagmar Reinhardt, Rob Saunders, and Jane Burry, editors, *Robotic Fabrication in Architecture, Art and Design 2016*, page 350–361. Springer International Publishing, Cham, 2016.

[TDKKCEK16] Neil T. Dantam, Zachary K. Kingston, Swarat Chaudhuri, and Lydia E. Kavraki. Incremental Task and Motion Planning: A Constraint-Based Approach. In *Robotics: Science and Systems (RSS)*, 2016.

[Ted14] Arturo Tedeschi. *AAD, Algorithms-aided design: parametric strategies using Grasshopper*. Le penseur publisher, 2014.

[TH22] Mohamed Tarek and Yijiang Huang. Simplifying deflation for non-convex optimization with applications in bayesian inference and topology optimization. *arXiv preprint arXiv:2201.11926*, 2022.

[THD20] Marc Toussaint, Jung-Su Ha, and Danny Driess. Describing physics for physical reasoning: Force-based sequential manipulation planning. *IEEE Robotics and Automation Letters*, 5(4):6209–6216, 2020.

[THM19] Lavender Tessmer, Yijiang Huang, and Caitlin Mueller. Additive casting of mass-customizable bricks. In *Proceedings of ACADIA*, page 198–207, Austin (Texas), USA, 2019.

[TKC20] Daniel Tish, Nathan King, and Nicholas Cote. Highly accessible platform technologies for vision-guided, closed-loop robotic assembly of unitized enclosure systems. *Construction Robotics*, 4(1):19–29, Jun 2020.

[TKH19] Hiroki Takabayashi, Keita Kado, and Gakuhito Hirasawa. Versatile robotic wood processing based on analysis of parts processing of japanese traditional wooden buildings. In Jan Willmann, Philippe Block, Marco Hutter, Kendra Byrne, and Tim Schork, editors, *Robotic Fabrication in Architecture, Art and Design 2018*, page 221–231, Cham, 2019. Springer International Publishing.

[TL17] Marc Toussaint and Manuel Lopes. Multi-bound tree search for logic-geometric programming in cooperative manipulation domains. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4044–4051, 2017.

[TMG+18] Kam-Ming Tam, Daniel J M Marshall, Mitchell Gu, Jasmine Kim, Yijiang Huang, Justin A Lavallee, and Caitlin T Mueller. Fabrication-aware structural optimisation of lattice additive-manufactured with robot-arm. *International Journal of Rapid Manufacturing*, 7(2-3), 2018.

[TMV+16] Mary Kathryn Thompson, Giovanni Moroni, Tom Vaneker, Georges Fadel, R Ian Campbell, Ian Gibson, Alain Bernard, Joachim Schulz, Patricia Graf, Bhrigu Ahuja, et al. Design for additive manufacturing: Trends, opportunities, considerations, and constraints. *CIRP annals*, 65(2):737–760, 2016.

[Tob15] Bonwetsch Tobias. *Robotically assembled brickwork: Manipulating assembly processes of discrete elements*. PhD thesis, ETH Zurich, 2015.

[Tou15] Marc Toussaint. Logic-geometric programming: an optimization-based approach to combined task and motion planning. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 1930–1936. AAAI Press, 2015.

[TPBVM21] Iremnur Tokac, Johan Philips, Herman Bruyninckx, and Andrew Vande Moere. Fabrication grammars: bridging design and robotics to control emergent material expressions. *Construction Robotics*, 5(1):35–48, Mar 2021.

[TPCS16] Davide Tonelli, Nico Pietroni, Paolo Cignoni, and Roberto Scopigno. Design and Fabrication of Grid-shells Mockups. In *Proceedings of Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference*, page 7, 2016.

[TSA+86] N. Tanaka, M. Saito, K. Arai, K. Banno, T. Ochi, and S. Kikuchi. The development of the "mark ii" mobile robot for concrete slab finishing. In A. et al. Bijl, editor, *CAD and Robotics in Architecture and Construction: Proceedings of the Joint International Conference at Marseilles, 25–27 June 1986*, page 249–257. Springer US, Boston, MA, 1986.

[TSP13] Romain Testuz, Yuliy Schwartzburg, and Mark Pauly. Automatic Generation of Constructable Brick Sculptures. In *Computer Graphics Forum (Proc. Eurographics) 2013*, page 4, 2013.

[TSW+19] Keke Tang, Peng Song, Xiaofei Wang, Bailin Deng, Chi-Wing Fu, and Ligang Liu. Computational Design of Steady 3D Dissection Puzzles. *Computer Graphics Forum*, 38(2):291–303, 2019.

[TWS03] Mark Taylor, Sam Wamuziri, and Ian Smith. Automated construction in japan. *Proceedings of the Institution of Civil Engineers - Civil Engineering*, 156(1):34–41, Feb 2003.

[UMK17] Erva Ulu, James Mccann, and Levent Burak Kara. Lightweight structure design under force location uncertainty. *ACM Transactions on Graphics*, 36(4):1–13, July 2017.

[VB12] D. Veenendaal and P. Block. An overview and comparison of structural form finding methods for general networks. *International Journal of Solids and Structures*, 49(26):3741–3753, December 2012.

[VBD+15] Lauren Vasey, Ehsan Baharlou, Moritz Dörstelmann, Valentin Koslowski, Marshall Prado, Gundula Schieber, Achim Menges, and Jan Knippers. Behavioral design and adaptive robotic fabrication of a fiber composite compression shell with pneumatic formwork. In *Proceedings of ACADIA*, page 13, 2015.

[VBR20] William Vega-Brown and Nicholas Roy. Asymptotically optimal planning under piecewise-analytic constraints. In *Algorithmic Foundations of Robotics XII*, pages 528–543. Springer, 2020.

[Ves21] Petras Vestartas. *Design-to-Fabrication Workflow for Raw-Sawn-Timber using Joinery Solver*. Doctoral thesis, EPFL, 2021.

[VFP+20] Lauren Vasey, Benjamin Felbrich, Marshall Prado, Behrooz Tahanzadeh, and Achim Menges. Physically distributed multi-robot coordination and collaboration in construction. *Construction Robotics*, 4(1):3–18, Jun 2020.

[VGB+14] Juraj Vanek, JA Garcia Galicia, Bedrich Benes, R Měch, N Carr, Ondrej Stava, and GS Miller. Packmerger: A 3d print volume optimizer. In *Computer Graphics Forum*, volume 33, pages 322–332. Wiley Online Library, 2014.

[VHWP12] Etienne Vouga, Mathias Höbinger, Johannes Wallner, and Helmut Pottmann. Design of self-supporting surfaces. *ACM Transactions on Graphics*, 31(4):1–11, August 2012.

[Vmo21] Tom Van Mele and many others. COMPAS: A framework for computational research in architecture and structures., 2017-2021. http://compas.dev.

[VS18] Elena Vazquez and Marcus Shaffer. Bring in the noise - a robotic-aided framework for the indirect shape translation and molding of inexact geometries. page 827–834, Łódź, Poland, 2018.

[VVB+14] Max Vomhof, Lauren Vasey, Stefan Brauer, Kurt Eggenschwiler, Jurgen Strauss, Fabio Gramazio, and Matthias Kohler. Robotic fabrication of acoustic brick walls. In *Proceedings of ACADIA*, page 555–564, Los Angeles (Califormia), USA, 2014.

[VZF+19] Josh Vekhter, Jiacheng Zhuo, Luisa F Gil Fandino, Qixing Huang, and Etienne Vouga. Weaving geodesic foliations. *ACM Transactions on Graphics*, 38(4):1–22, August 2019.

[WAC+12] Jan Willmann, Federico Augugliaro, Thomas Cadalbert, Raffaello D'Andrea, Fabio Gramazio, and Matthias Kohler. Aerial robotic construction towards a new field of architectural research. *International Journal of Architectural Computing*, 10(3):439–459, Sep 2012.

[WAGM20] Hans Jakob Wagner, Martin Alvarez, Abel Groenewolt, and Achim Menges. Towards digital automation flexibility in large-scale timber construction: integrative robotic prefabrication and co-design of the buga wood pavilion. *Construction Robotics*, 4(3):187–204, Dec 2020.

[Wan20] Joann Wang. Digital architextiles: Nonwoven textile thermoforming in robotic fabrication. In *25th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2020*, Hong Kong, 2020.

[WAWS17] Jun Wu, Niels Aage, Rüdiger Westermann, and Ole Sigmund. Infill optimization for additive manufacturing—approaching bone-like porous structures. *IEEE transactions on visualization and computer graphics*, 24(2):1127–1140, 2017.

[WBS+19] Bastian Wibranek, Boris Belousov, Alymbek Sadybakasov, Jan Peters, and Oliver Tessmann. Interactive structure robotic repositioning of vertical elements in man-machine collaborative assembly through vision-based tactile sensing. In *Blucher Design Proceedings*, page 705–714, Porto, Portugal, Dec 2019. Editora Blucher.

[WC16] Nicholas Williams and John Cherrey. Crafting robustness: Rapidly fabricating ruled surface acoustic panels. In Dagmar Reinhardt, Rob Saunders, and Jane Burry, editors, *Robotic Fabrication in Architecture, Art and Design 2016*, page 294–303. Springer International Publishing, Cham, 2016.

[WCT+15] Weiming Wang, Haiyuan Chao, Jing Tong, Zhouwang Yang, Xin Tong, Hang Li, Xiuping Liu, and Ligang Liu. Saliency-Preserving

Slicing Optimization for Effective 3D Printing. *Computer Graphics Forum*, 34(6):148–160, 2015.

[WDF+20] Chenming Wu, Chengkai Dai, Guoxin Fang, Yong-Jin Liu, and Charlie C. L. Wang. General Support-Effective Decomposition for Multi-Directional 3-D Printing. *IEEE Transactions on Automation Science and Engineering*, 17(2):599–610, April 2020.

[WDK14] James Warton, Rajeev Dwivedi, and Radovan Kovacevic. Additive manufacturing of metallic alloys. In Wes McGee and Monica Ponce de Leon, editors, *Robotic Fabrication in Architecture, Art and Design 2014*, page 147–161. Springer International Publishing, Cham, 2014.

[Web20] Ramon Elias Weber. *Geometries of light*. Master thesis, Massachusetts Institute of Technology, 2020.

[Wei14] Renate Weissenböck. D-form: Exploring the combination of laser cutting and robotic thermoforming as a technique for architectural envelopes. In Wes McGee and Monica Ponce de Leon, editors, *Robotic Fabrication in Architecture, Art and Design 2014*, page 249–260. Springer International Publishing, Cham, 2014.

[Wei16] Yves Weinand. *Advanced timber structures: architectural designs and digital dimensioning*. Birkhäuser, 2016.

[Wei17] Renate Weissenböck. Robotrack - linking manual and robotic design processes by motion-tracking. In *Proceedings of eCAADe*, page 651–660, Rome, Italy, 2017.

[Wel94] Daniel S Weld. An introduction to least commitment planning. *AI magazine*, 15(4):27, 1994.

[WFIS00] Tatsuya Wakisaka, Noriyuki Furuya, Yasuo Inoue, and Takashi Shiokawa. Automated construction system for high-rise reinforced concrete buildings. *Automation in Construction*, 9(3):229–250, May 2000.

[WGKLP21] Zi Wang, Caelan Reed Garrett, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Learning compositional models of robot skills for task and motion planning. *The International Journal of Robotics Research*, 40(6-7):866–894, 2021.

[WGZ+19] Xiang Wang, Zhe Guo, Xiao Zhang, Jinxi Jin, and Philip F. Yuan. Design, analysis and robotic fabrication of a bending-active shell structure with thin sheets based on curved-crease-folding technique. In *Proceedings of CAADRIA*, page 63–72, Wellington, New Zealand, 2019.

[Wil91] Gordon Wilfong. Motion planning in the presence of movable obstacles. *Annals of Mathematics and Artificial Intelligence*, 3(1):131–150, 3 1991.

[Wil92] Randall H Wilson. *On Geometric Assembly Planning*. PhD thesis, Stanford University, 1992.

[WJVS21] Ludwig Wilhelm Wall, Alec Jacobson, Daniel Vogel, and Oliver Schneider. Scrappy: Using Scrap Material as Infill to Make Fabrication More Sustainable. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–12, Yokohama Japan, May 2021. ACM.

[WK18] Kaicong Wu and Axel Kilian. Robotic equilibrium: Scaffold free arch assemblies. In [*Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, page 8, 2018.

[WK19] Kaicong Wu and Axel Kilian. Designing natural wood log structures with stochastic assembly and deep learning. In Jan Willmann, Philippe Block, Marco Hutter, Kendra Byrne, and Tim Schork, editors, *Robotic Fabrication in Architecture, Art and Design 2018*, page 16–30, Cham, 2019. Springer International Publishing.

[WKB+16] Jan Willmann, Michael Knauss, Tobias Bonwetsch, Anna Aleksandra Apolinarska, Fabio Gramazio, and Matthias Kohler. Robotic timber construction — expanding additive fabrication to new dimensions. *Automation in Construction*, 61(Supplement C):16–23, Jan 2016. 00000.

[WL17] De Wolf and Catherine (Catherine Elvire Lieve). *Low Carbon Pathways for Structural Design : Embodied Life Cycle Impacts of Building Structures*. Thesis, Massachusetts Institute of Technology, 2017.

[WM92] Randall H Wilson and Toshihiro Matsui. Partitioning an assembly for infinitesimal motions in translation and rotation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1311–1318. IEEE, 1992.

[WM21] Eamon Whalen and Caitlin Mueller. Toward Reusable Surrogate Models: Graph-Based Transfer Learning on Trusses. *Journal of Mechanical Design*, 144(2), September 2021.

[WMF16] Simon Weir, Dion Moult, and Shayani Fernando. Stereotomy of wave jointed blocks. In Dagmar Reinhardt, Rob Saunders, and Jane Burry, editors, *Robotic Fabrication in Architecture, Art and Design 2016*, page 284–293. Springer International Publishing, Cham, 2016.

[WMW+20] Weiming Wang, Dirk Munro, Charlie C. L. Wang, Fred van Keulen, and Jun Wu. Space-time topology optimization for additive manufacturing: Concurrent optimization of structural layout and fabrication sequence. *Structural and Multidisciplinary Optimization*, 61(1):1–18, Jan 2020.

[WOD09] Emily Whiting, John Ochsendorf, and Frédo Durand. Procedural modeling of structurally-sound masonry buildings. In *ACM SIGGRAPH Asia 2009 papers*, pages 1–9. 2009.

[Wol89] Jan D Woltery. On the Automatic Generation of Assembly Plans. In *Proceedings, 1989 International Conference on Robotics and Automation*, pages 62–68, 1989.

[WPGM16] Rundong Wu, Huaishu Peng, François Guimbretière, and Steve Marschner. Printing arbitrary meshes with a 5DOF wireframe printer. *ACM Transactions on Graphics (TOG)*, 35(4):101, 2016.

[WPN14] Justin Werfel, Kirstin Petersen, and Radhika Nagpal. Designing collective behavior in a termite-inspired robot construction team. *Science*, 343(6172):754–758, Feb 2014.

[WQZ+18] Xiangzhi Wei, Siqi Qiu, Lin Zhu, Ruiliang Feng, Yaobin Tian, Juntong Xi, and Youyi Zheng. Toward Support-Free 3D Printing: A Skeletal Approach for Partitioning Models. *IEEE Transactions on Visualization and Computer Graphics*, 24(10):2799–2812, October 2018.

[WSF22] Shih-Yuan Wang, Yu-Ting Sheng, and Florian Frank. Taco: Abb robot programming for grasshopper. http://blickfeld7.com/architecture/rhino/grasshopper/Taco, 2022.

[WSIP19] Ziqi Wang, Peng Song, Florin Isvoranu, and Mark Pauly. Design and structural optimization of topological interlocking assemblies. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, November 2019.

[WSP18] Ziqi Wang, Peng Song, and Mark Pauly. Desia: a general framework for designing interlocking assemblies. In *SIGGRAPH Asia 2018 Technical Papers on - SIGGRAPH Asia '18*, page 1–14, Tokyo, Japan, 2018. ACM Press.

[WSP21] Ziqi Wang, Peng Song, and Mark Pauly. State of the Art on Computational Design of Assemblies with Rigid Parts. *Computer Graphics Forum*, 40(2):633–657, May 2021.

[WSS+13] Jakob Weigele, Manuel Schloz, Tobias Schwinn, Steffen Reichert, Riccardo LaMagna, Frédéric Waimer, Jan Knippers, and Achim Menges. Fibrous morphologies. In *Proceedings of eCAADe*, page 549–558, Delft, The Netherlands, 2013.

[WSW+12] Emily Whiting, Hijung Shin, Robert Wang, John Ochsendorf, and Frédo Durand. Structural optimization of 3D masonry buildings. *ACM Transactions on Graphics*, 31(6):1–11, November 2012.

[WT16] Seth Watts and Daniel A Tortorelli. An n-material thresholding method for improving integerness of solutions in topology optimization. *International Journal for Numerical Methods in Engineering*, 108(12):1498–1524, 2016.

[WYA+19] Dylan Wood, Maria Yablonina, Miguel Aflalo, Jingcheng Chen, Behrooz Tahanzadeh, and Achim Menges. Cyber physical macro material as a uav [re]configurable architectural system. In Jan Willmann, Philippe Block, Marco Hutter, Kendra Byrne, and Tim Schork, editors, *Robotic Fabrication in Architecture, Art and Design 2018*, page 320–335, Cham, 2019. Springer International Publishing.

[WYAM20] Maria Wyller, Maria Yablonina, Martin Alvarez, and Achim Menges. Adaptive kinematic textile architecture. *Construction Robotics*, 4(3):227–237, Dec 2020.

[WYS18] Yaowu Wang, Zhenmin Yuan, and Chengshuang Sun. Research on assembly sequence planning and optimization of precast concrete buildings. *Journal of Civil Engineering and Management*, 24(22):106–115, Mar 2018.

[WZK16] W. M. Wang, C. Zanni, and L. Kobbelt. Improved Surface Quality in 3D Printing by Optimizing the Printing Direction. *Computer Graphics Forum*, 35(2):59–70, 2016.

[XHP20] Ziju Xian, Nicholas Hoban, and Brady Peters. Robotically fabricated reciprocal frame wall. In *Proceedings of eCAADe*, page 10, 2020.

[XLG19] Weiguo Xu, Dan Luo, and Yuan Gao. Automatic brick masonry system and its application in on-site construction. In *proceedings of CAADRIA*, page 83–92, Wellington, New Zealand, 2019.

[YCL+15] Miaojun Yao, Zhili Chen, Linjie Luo, Rui Wang, and Huamin Wang. Level-set-based partitioning and packing optimization of a printable model. *ACM Transactions on Graphics*, 34(6):1–11, November 2015.

[YCXW17] Miaojun Yao, Zhili Chen, Weiwei Xu, and Huamin Wang. Modeling, evaluation and optimization of interlocking shell pieces. In *Computer Graphics Forum*, volume 36, pages 1–13. Wiley Online Library, 2017.

[YG07] Zhenwang Yao and Kamal Gupta. Path planning with general end-effector constraints. *Robotics and Autonomous Systems*, 55(4):316–327, 2007.

[YHL+16] Lei Yu, Yijiang Huang, Zhongyuan Liu, Sai Xiao, Ligang Liu, Guoxian Song, and Yanxin Wang. Highly Informed Robotic 3D Printed Polygon Mesh: A Novel Strategy of 3D Spatial Printing. In *Proceedings of the 36st Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, pages 298–307, 2016.

[YKGA17] Jiaxian Yao, Danny M. Kaufman, Yotam Gingold, and Maneesh Agrawala. Interactive design and stability analysis of decorative joinery for furniture. *ACM Transactions on Graphics*, 36(2):1–16, Apr 2017.

[YLL19] Xuyou Yang, Paul Loh, and David Leggett. Robotic variable fabric formwork. *Journal of Computational Design and Engineering*, 6(3):404–413, Jul 2019.

[YLX19] Lei Yu, Dan Luo, and Weiguo Xu. Dynamic robotic slip-form casting and eco-friendly building façade design. In Jan Willmann, Philippe Block, Marco Hutter, Kendra Byrne, and Tim Schork, editors, *Robotic Fabrication in Architecture, Art and Design 2018*, page 421–433, Cham, 2019. Springer International Publishing.

[YM98a] Yusuke Yamazaki and Junichiro Maeda. The smart system: an integrated application of automation and information technology in production process. *Computers in Industry*, 35(1):87–99, Feb 1998.

[YM98b] Yusuke Yamazaki and Junichiro Maeda. The smart system: an integrated application of automation and information technology in production process. *Computers in Industry*, 35(1):87–99, Feb 1998.

[YM19] Maria Yablonina and Achim Menges. Towards the development of fabrication machine species for filament materials. In Jan Willmann, Philippe Block, Marco Hutter, Kendra Byrne, and Tim Schork, editors, *Robotic Fabrication in Architecture, Art and Design 2018*, page 152–166, Cham, 2019. Springer International Publishing.

[YMD14] Philip F. Yuan, Hao Meng, and Pradeep Devadass. Performative tectonics. In Wes McGee and Monica Ponce de Leon, editors, *Robotic Fabrication in Architecture, Art and Design 2014*, page 181–195. Springer International Publishing, Cham, 2014.

[YMYZ16] Philip F Yuan, Hao Meng, Lei Yu, and Liming Zhang. Robotic Multi-dimensional Printing Based on Structural Performance. In

Robotic Fabrication in Architecture, Art and Design 2016, pages 92–105. Springer, 2016.

[Yos06] Tetsuji Yoshida. A short history of construction robots research & development in a japanese company. In *Proceedings of ISARC*, page 188–193. IAARC, Oct 2006.

[YUNY84] Tetsuji Yoshida, Takatoshi Ueno, Minoru Nonaka, and Shinobu Yamazaki. Development of spray robot for fireproof cover work. In *Proceedings of ISARC*, Pittsburgh, PA, USA, Jun 1984.

[YYT+17] Eric A. Yu, Jin Yeom, Cem C. Tutum, Etienne Vouga, and Risto Miikkulainen. Evolutionary decomposition for 3D printing. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '17, pages 1272–1279, New York, NY, USA, July 2017. Association for Computing Machinery.

[ZB16] Yinan Zhang and Devin Balkcom. Interlocking structure assembly with voxels. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2173–2180, October 2016.

[ZFS+19] Xiaoting Zhang, Guoxin Fang, Melina Skouras, Gwenda Gieseler, Charlie C. L. Wang, and Emily Whiting. Computational design of fabric formwork. *ACM Transactions on Graphics*, 38(4):1–13, August 2019.

[ZLB+20] SASA ZIVKOVIC, LESLIE LOK, JANE BURRY, JENNY SABIN, BOB SHEIL, and MARILENA SKAVARA. Making form work: Experiments along the grain of concrete and timber. In *Fabricate 2020*, Making Resilient Architecture, page 116–123. UCL Press, 2020.

[ZOL21] Guanqi Zhu, Ya Ou, and Dan Luo. Robotic weaving of customizable frp formworks for large-scale optimized structure. In *Proceedings of the 26th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2021*, page 10, 2021.

[ZPC21] Simon Zimmermann, Roi Poranne, and Stelian Coros. Go fetch! - dynamic grasps using boston dynamics spot with external robotic arm. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, page 4488–4494, Xi'an, China, May 2021. IEEE.

[ZPZ13] Qingnan Zhou, Julian Panetta, and Denis Zorin. Worst-case structural analysis. *ACM Transactions on Graphics*, 32(4):1–12, July 2013.

[ZSB14] Martin Zaefferer, Jörg Stork, and Thomas Bartz-Beielstein. Distance Measures for Permutations in Combinatorial Efficient Global Optimization. In Thomas Bartz-Beielstein, Jürgen Branke, Bogdan

Filipič, and Jim Smith, editors, *Parallel Problem Solving from Nature – PPSN XIII*, Lecture Notes in Computer Science, pages 373–383, Cham, 2014. Springer International Publishing.

[ZWT21] Lufeng Zhu, Bastian Wibranek, and Oliver Tessmann. Robo-sheets - double-layered structure based on robot-aided plastic sheet thermoforming. In *Proceedings of CAADRIA*, page 643–652, Hong Kong, 2021.

[ZZY21] Qiang Zhan, Xinjie Zhou, and Philip F. Yuan. Digital design and fabrication of a 3d concrete printed prestressed bridge. In *Proceedings of the 26th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2021*, page 663–672, Hong Kong, 2021.

[Öz14] Güvenç Özel. Synchronization of robotically actuated motion, sense-based interaction and computational inference. In *Proceedings of ACADIA*, page 10, 2014.