2025 European Conference on Computing in Construction

CIB W78 Conference on IT in Construction

Porto, Portugal
July 14-17, 2025

GRAPH BASED DISASSEMBLY SEQUENCING WITH STRUCTURAL AND STABILITY CONSTRAINTS

Zain Karsan, Benjamin Dillenburger, and Catherine De Wolf
ETH Zurich, Zurich, Switzerland

Abstract

Disassembly is an important strategy in achieving material circularity and closing the loop in material flow. While disassembly sequencing is heavily studied in manufacturing, few examples consider the constraints present in construction. To address this gap we propose a graph based method for disassembly sequencing with constraints on stability and internal stress. We test our algorithm on three frame structures of increasing complexity and element count using construction specific heuristics to determine source nodes and disassembly direction. Our algorithm can compute feasible disassembly sequences with sufficient speed to support applications in online robotic path planning.

Introduction

Disassembly sequencing is a richly studied topic in manufacturing with several classes of problems ranging from 2D puzzles, to complex real world assemblies, each with as many strategies to find feasible solutions (Lambert, 2003). In the context of the circular economy, disassembly offers a viable pathway to element reuse, by considering end of life products as material banks, hence retaining products and materials for as long as possible before disposal,(Çetin et al., 2021). Furthermore, the study of disassembly sequencing provides a useful heuristic for assembly problems considering the reflection between assembly and disassembly for non-deformable parts (Tian et al., 2022). Within the Architecture, Engineering and Construction (AEC) community, disassembly sequencing is typically undertaken for EoL building and material management, or to facilitate and schedule on-site operations in a systematic way. Though the latter is more regularly used in line balancing optimization for robotic manufacturing (Zeng et al., 2022), the coordination of multiple actors on-site is a challenge in contemporary construction and demolition works.

Disassembly can be categorized into sequence planning, line balancing, and path planning according to the taxonomies identified by Ghandi and Masehian (2015). For this research we focus specifically on generating feasible sequences, with the aim of applying the algorithm developed here to a line balancing and path planning problem in the context of on-site disassembly robotics. We consider disassembly sequencing as the order in which an assembly of n discrete elements is reduced to zero elements. Our approach leverages existing frameworks for representing disassembly problems as graphs, with precedence relationships encoded in some way to guide the sequencing algorithm to feasible actions. However, the heuristics and feasibility checks typical of contemporary graph based disassembly are specific to product manufacturing, which may not always generalize to disassembly problems at the scale of construction.

Hence, with this paper, we aim to address the following research question: How should graph based disassembly sequencing methods adapt to support scenarios in construction and what heuristics and feasibility checks should be considered? To answer this question, we extend the current state of the art in disassembly sequence structure graphs by automating key elements of the problem formulation. We automate the construction of an assembly graph from an unordered collection of meshes, use various geometric manipulations to derive precedence relations and implement two feasibility checks involving structural stresses and stability.

Background

Graph based structures are a powerful way to analyze assemblies and the relationships between constituent parts. Representations of this kind originate from research in manufacturing optimization and experiments in geometric reasoning (Gengenbach et al., 1996)(Wilson and Latombe, 1994). Hence myriad representational strategies for encoding sequences and parts in graph structures exist. Bourjault proposes the use of AND/OR graphs to fully explore disassembly sequence space Bourjault (1988). Moore et al. (1998) postulate the use of petri nets, which function like finite state machines, to describe disassembly processes. Smith et al. (2012) use a collection of matrices to encode precedence relationships, and a series of expert rules to develop disassembly plans. Here, several auxiliary matrices encode the spatial relationships of the constituent parts to augment the topological relationships present in the assembly graph's adjacency matrix.

This approach, which has served as the basis for later work by Sanchez and Haas (2018) considers precedence relations as motion, projection, fastener, and constraint matri-

ces that are updated as parts are removed. While the expressive capability of this approach is used in the context of BIM, and enriched with additional layers of information (Sanchez et al., 2020), the encoding of precedence relations is done explicitly which severely limits applicability to more complex assemblies and elements. Nevertheless, the implementation of so-called disassembly structure sequence graphs in BIM represents a powerful decision making tool for applications like adaptive reuse and end of life planning for buildings (Sanchez et al., 2021). Indeed, embedding BIM models with additional layers of information to inform material flows and divert building materials to waste streams (Akanbi et al., 2019) supports an important principle of the Circular Economy (CE), termed 'closing' the material loop (Çetin et al., 2021). Furthermore, disassembly sequencing finds common applications for planning purposes in the AEC, typically with BIM.
However, we find very few applications of disassembly sequencing for automating construction and demolition processes on site. Bruun et al. (2024) use a graph based approach to inform the sequencing of robotic disassembly in a multi-robot work cell. While this approach is inherently off-site and akin to typical manufacturing, the scale of the assembly and disassembly process is indeed redolent of construction. The work described here makes use large payload industrial robots with many more degrees of freedom than are available to contemporary construction equipment, which may limit its applicability to on-site disassembly. This may be especially true for heavy structural components whose only feasible disassembly direction may be vertical. Hence, heuristics for disassembly that consider the limitations of contemporary construction equipment are currently missing in literature. Furthermore, the proportion of work dedicated to modeling the disassembly of elastic or deformable elements is minimal (Ghandi and Masehian, 2015). This is natural given the application space of the majority of disassembly research occurs at the scale of commercial products. Hence, center of mass stability checks provided by physics solver suffice (Tian et al., 2022). However, at the scale of construction, self weight and elasticity may play non-negligible roles in feasible disassembly sequencing. Therefore, our approach aims to address a crucial gap in translating disassembly research typically done off-site and at small scale, to the context of construction.

Methods and Experiment

We develop a graph based disassembly sequencing algorithm with structural and stability feasibility constraints. Furthermore, we propose heuristics that are relevant to construction and disassembly to steer our algorithm toward feasible solutions. We test this algorithm on several prototypical frame structures of increasing element count and complexity. We formulate the disassembly sequencing problem along the terminology identified by Ghandi and Masehian (2015). We assume the disassembly sequencing problems encountered in construction are monotone,

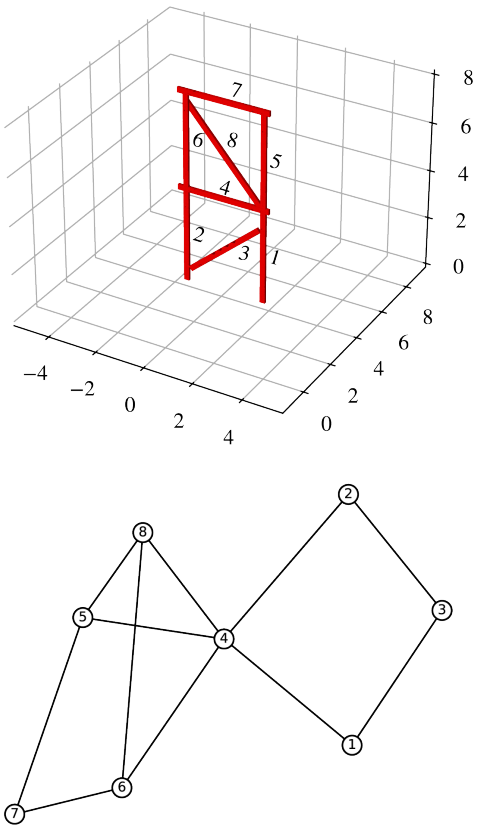

Figure 1: Simple Frame Example Axonometric Projection and Connectivity Graph

linear and sequential. Further, our model formulation involves only translational motion and consider only the part to be disassembled, without the presence of manipulators or hands performing the work, though this is certainly an endeavor for future work. Our disassembly problem is three dimensional with "toleranced" geometry.
Given an unordered collection of meshes representing discrete elements, we search for a feasible disassembly sequence. As an illustrative example, consider a simple collection of prismatic meshes resembling a structural frame and its connectivity graph shown in Figure 1. Our algorithm automates the encoding of precedence relations, performs structural feasibility calculations based on finite difference approximations of the partial differential equation (PDE) for linear elasticity. Additionally, we use geometric heuristics to determine stability. Here we show the pseudo-code used in the generation of auxiliary matrices to describe precedence relations in Algorithms 1 and 2, our various feasibility checks in Algorithms 3,and 4, and the sequencing planner itself in Algorithm 5. We compute the assembly graph by simply performing a collision check, inferring that meshes in collision correspond to connected parts. Assuming small errors in the mesh geometry, we scale each member by a factor proportional to their maximum principal direction. Next, we compute constraint and motion matrices which encode the precedence relationships. Our contact constraint and motion constraint

Algorithm 1 Build Contact Constraint Matrix

Require: Graph $G = (V,E)$ where each vertex $v \in V$ has a mesh attribute M_v, small displacement ε
Ensure: Returns constraint direction matrix C

```
D ← [+X, −X, +Y, −Y, +Z, −Z]
C ← ∅
for all n ∈ V do
    C[n] ← ∅
    Collision_Check ← child ∈ Neighbors(G, n)
    for all d ∈ D do
        v ← Scaled(v)
        v ← ApplyTranslation(v, ε · d)
        Collision_Check ← v
        for all c ∈ Collision_Check do
            C[n, d] ← c
        end for
    end for
end for
    return C
```

matrix follows the methodology outlined by Smith et al, however we dispense with fastener constraint and projection constraint matrices as less relevant for our disassembly problem. The columns of this matrix correspond to negative and positive directions along Cartesian axes, and each row corresponds to an element (Smith et al., 2012). The contact constraint matrix is essentially the adjacency matrix of the assembly organized into spatial directions. The contact constraint matrix for this example is given by (1).

$$M_c = \begin{bmatrix} 0 & 3 & 0 & 0 & 4 & 0 \\ 3 & 0 & 0 & 0 & 4 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6,8,5 & 2,1 \\ 0 & 8 & 0 & 0 & 7 & 4 \\ 8 & 0 & 0 & 0 & 7 & 4 \\ 0 & 0 & 0 & 0 & 0 & 6,5 \\ 5 & 6 & 0 & 0 & 0 & 4 \end{bmatrix} \tag{1}$$

To compute the motion constraint matrix, we require the element to be convex. We use the convex hull of a swept volume to define a collision object and record the results in the motion constraint matrix. The motion constraint matrix for this example is given by (2).

$$M_m = \begin{bmatrix} 0 & 2,3 & 0 & 0 & 4,5,7 & 0 \\ 3,1 & 0 & 0 & 0 & 6,4,7 & 0 \\ 1 & 2 & 0 & 0 & 4,8,7 & 0 \\ 0 & 0 & 0 & 0 & 6,8,5,7 & 2,3,1 \\ 0 & 6,8 & 0 & 0 & 7 & 4,1 \\ 8,5 & 0 & 0 & 0 & 7 & 4,2 \\ 0 & 0 & 0 & 0 & 0 & 4,6,8,5,3,2,1 \\ 5 & 6 & 0 & 0 & 7 & 4,3 \end{bmatrix} \tag{2}$$

A key difference between these matrices is the neighborhood of collision objects considered in each. The contact constraint matrix considers immediately adjacent elements, while the motion constraint matrix considers all objects that block the element under consideration in each direction. Hence, the motion constraint matrix provides an essential heuristic for identifying subsequent elements to disassemble if the source element is blocked. We update these matrices after each part is removed, and use the result to inform the next part to consider for disassembly. In general, we search for a candidate element with the least amount of connections and blocking elements given by the contact constraint and motion constraint matrices respectively. Next we check the feasibility of removing this ele-

Algorithm 2 Build Motion Constraint Matrix

Require: Graph $G = (V,E)$ where each vertex $v \in V$ has a mesh attribute
Ensure: Returns Motion Constraint matrix M

```
L ← max(get_bounds(G))
D ← [+X, −X, +Y, −Y, +Z, −Z]
M ← ∅
for all n ∈ V do
    M[n] ← ∅
    Collision_Check ← ∅
    for all v ∈ V, v ≠ n do
        v ← Scaled(v)
        Collision_Check ← v
    end for
    for all d ∈ D do
        ṽ ← Translate(v, d, L)        ▷ translate in each dir
        M_v ← ConvexHull(ṽ, v)
        Collision_Check ← M_v
        for all c ∈ Collision_Check do
            M[n, d] ← c
        end for
    end for
end for
    return M
```

ment as it effects the structure and stability of the remaining assembly. If both these conditions are feasible, the part is removed and the next candidate is sought. To determine the structural feasibility of a disassembly action, we compute the internal stresses of a voxel-based representation of the assembly without the candidate element. We use a framework developed by Erzmann et al which uses Finite Differences (FDM) over a quad-grid to solve the PDE for linear elasticity Erzmann et al. (2023). The goal of this framework is intended for deep learning in topology optimization, however the zero solution of the topology optimization problem provides a fast computation of the stresses in our assemblies. For our sequencing problems, the applied loads are merely the self weight of each voxel in the assembly which we assume to be steel. We use the values in the Table 1 to compute the von Mises stresses.
If the ratio of the maximum stress divided by the yield stress σ_y exceeds 1, the structural check fails. The Dirichlet boundary conditions for this problem are the bottom voxels of the assembly, for which the location in the Z

Table 1: Mechanical and Geometric Parameters

Quantity	Value
Elastic Modulus	200 GPa
Poisson Ratio	0.3
Yield Stress	250 MPa
Density	7850 $\frac{kg}{m^3}$
Voxel Size	0.05m

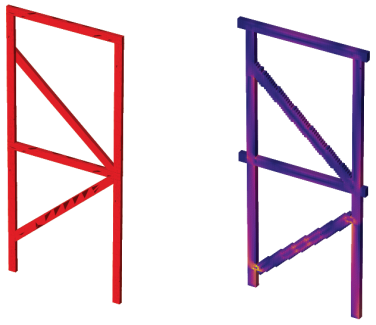

Figure 2: Mesh assembly (left) and stress states calculated on a voxelized representation (right)

axis is zero. This means that we assume no displacement at the notional ground plane or foundation of the assembly. A typical stress plot will show areas of high stress,

Algorithm 3 Check Structural Feasibility

Require: Graph $G = (V,E)$ where each vertex $v \in V$ has a mesh attribute, voxel size v_{dim}, maximum allowable stress σ_y
Ensure: Returns False if $\max(\sigma_{vm})/\sigma_y > 1$
  $\Omega \leftarrow$ Voxelize(G, v_{dim}) ▷ Get Voxel Grid
  $F \leftarrow \emptyset$
  for all $\omega_i \in \Omega$ **do**
    AssignMechanicalProperties(ω_i)
    $f_i \leftarrow -v_{dim}^3 \cdot \rho \cdot g$ ▷ Assign Self Weight
    $F \leftarrow f_i$ ▷ Aggregate Force Tensor
  end for
  $D \leftarrow$ SetDirichletBoundaries$(\Omega, z = 0)$
  $\sigma_{vm} \leftarrow$ SolvePDE(Ω, D, F)
  if $\max(\sigma_{vm})/\sigma_y > 1$ **then return** False
  end if

which may lead to infeasible disassembly sequences if the yield is exceeded, for example shown in Figure 2. The stability check is a simple geometric heuristic that searches for floating elements. We dilate each remaining mesh element in the assembly, perform a boolean union of these meshes and identify the lower bounds of the resulting vertices in each disjoint mesh. The resulting disjoint meshes can be thought of as sub-assemblies. If the lowest vertices of any subassembly is non-zero, the sub-assembly must be floating. Our disassembly sequencing algorithm also fol-

Algorithm 4 Stability Check

Require: Graph $G = (V,E)$ where each vertex $v \in V$ has a mesh attribute ϕ_v
Ensure: Returns False if any disjointed mesh has a lower bound z-value > 0
  for all $v \in V$ **do**
    $\phi_v \leftarrow$ Dilate(ϕ_v)
  end for
  $\Phi \leftarrow \emptyset$
  for all $v \in V$ **do**
    $\Phi \leftarrow$ BooleanUnion(Φ, ϕ_v)
  end for
  $DisjointMeshes \leftarrow$ SeparateDisjointMeshes(Φ)
  for all $M_d \in DisjointMeshes$ **do**
    $(LB, UB) \leftarrow$ GetBounds(M_d) ▷ Get lower and upper bound of mesh
    if $LB_z > \varepsilon$ **then return** False
    end if
  end for
    return True

lows a similar methodology to Smith et al. (2012), which is a depth first search, with the exception of our additional feasibility checks and our assumptions on optimal disassembly direction. While Smith et al determine the optimal direction for each part using projection constraints, we assume the nature of the part itself and constraints given by construction equipment imply an optimal disassembly direction. For example, in the disassembly of structural frames, we assume the optimal disassembly direction is vertical by craning, however, following previous methods, the removal direction may be horizontal, which would be unrealistic in the context of construction.
With spatial precedence relationships and feasibility checks computed for each disassembly action, we apply our algorithm to a series of structural frames of arbitrary complexity shown in figure 3.

Discussion and result analysis

We test our algorithm on three assemblies, and find feasible disassembly sequences, shown in figure 4 represented as directed graphs describing the sequence of part removal. The algorithm is able to compute feasible disassembly sequences for a variety of structural topologies in reasonable computation time summarized in Table 2. To identify bottlenecks in the process, we timed various operations of the disassembly sequence planner. We report the computation times of various operations of the disassembly sequencing planner in Figure 5. These include the time to construct the graph,calculate auxiliary matrices, and perform our feasibility checks. For these computations, several variations of the Tower assembly were used, where the number of storeys, and hence the number of elements was gradually increased. For our assembly graph computation, we pro-

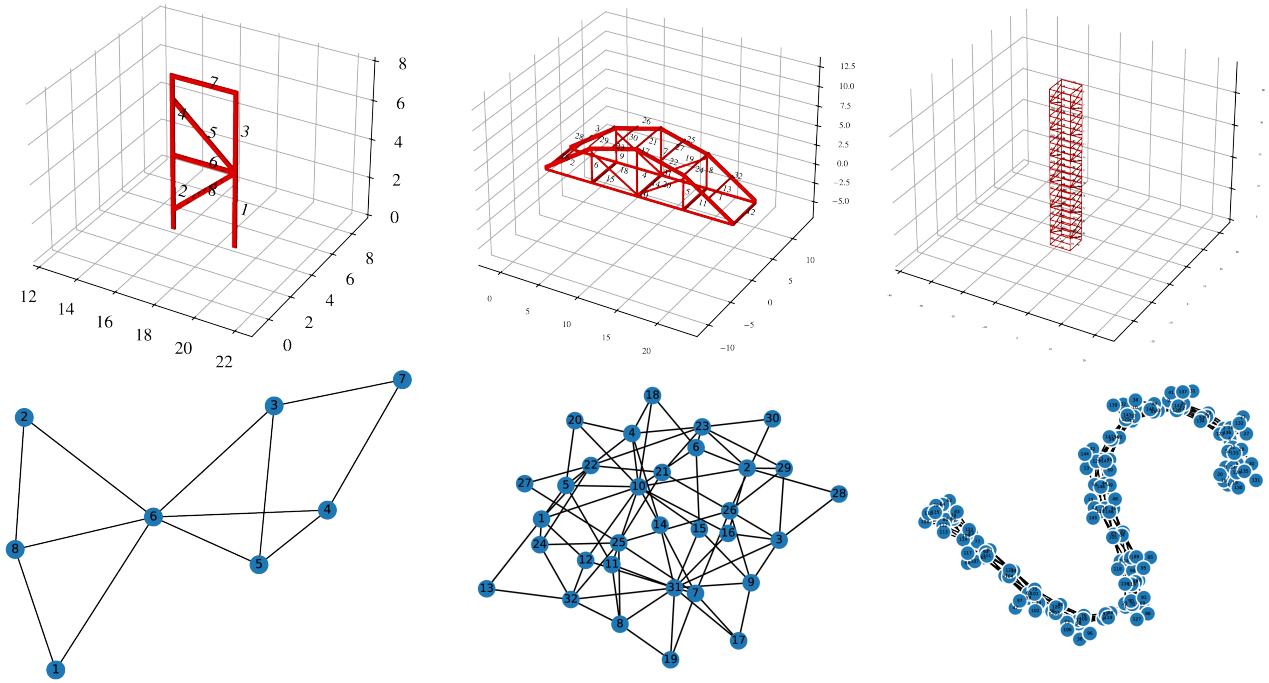

Figure 3: Disassembly experiments, axonometric projections and corresponding assembly graphs. Simple Frame (left), Bridge (middle) Tower (right)

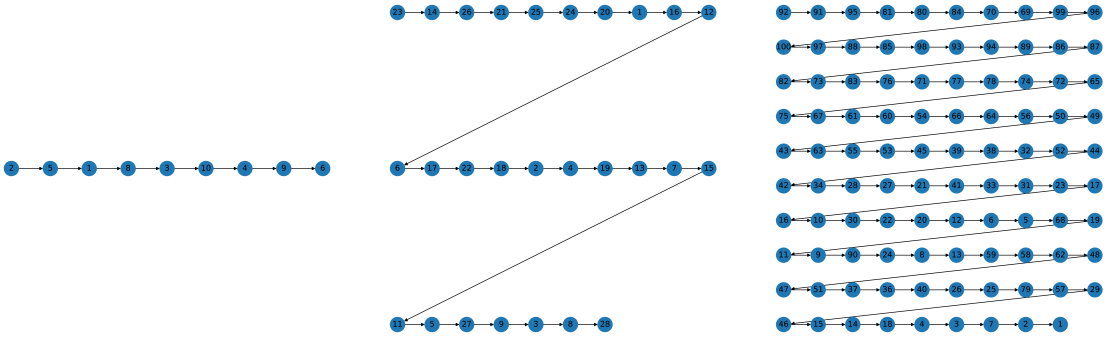

Figure 4: Disassembly sequences for a Simple Frame (left), Bridge (middle) and Tower (right)

Table 2: Disassembly Sequencing Results

Name	Number of Elements	Time
Simple Frame	10	0.7s
Bridge	28	4.5s
Tower	100	36.2s

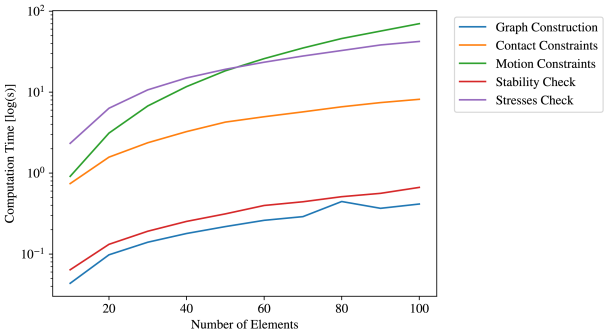

Figure 5: Computation times for processes in the disassembly sequencing algorithm

vide an unordered set of meshes which are assigned to the nodes of a graph as attributes, then queried for collisions, in order to recover an adjacency matrix. The next time intensive computations involve encoding precedence relationships in each auxiliary matrix. The contact constraints are similar in time complexity to the initial graph construction. Depending on the assembly however, the motion constraint can be significantly more intensive to calculate, potentially even exceeding the time necessary to compute internal stresses. We use a swept volume approach to compute the motion constraints, which means that for some

Algorithm 5 Disassembly Sequence Planner

Require: $G = (V, E), DSP = (), Q = ()$
Ensure: $DSP = (V)$ and $G \leftarrow \emptyset$
  function GETDSP(G, DSP, Q)
    while $V \neq \varnothing$ **do**
      $C \leftarrow BuildCC(G)$
      $M \leftarrow BuildMC(G)$
      $D \leftarrow \min P_{d,j}$ for $j = 0, 1 \cdots, 5$
      if $Q = \varnothing$ **then**
        $n \leftarrow G.selectNode()$ ▷ Decide by heuristic
        $Q \leftarrow n$ ▷ add n to the queue
      end if
      $n \leftarrow Q_0$ ▷ inspect first in queue
      $d \leftarrow D_n$ ▷ get optimal disassembly direction
      $DSP; n$ ▷ add n to the disassembly graph
      if $d \notin M$ & Stability(G) & $Structure(G)$ **then**
        $V \backslash n, E \backslash n$ ▷ remove n from G
        $DSP; m_{n,j}$ ▷ add blocking to DSP
        $DSP; (n, m_{n,j})$ ▷ add directed edge
      end if
      if $d \in M$ **then**
        for $m_{n,j}, j = 0$ to $5, m_{n,j} \notin Q$ & $\notin DSP$ **do**
          $Q; m_{n,j}$ ▷ add blocking to queue
        end for
      end if
      $getDSP(G, DSP, Q, X)$ ▷ recurse
    end while
    return DSP
  end function

parts projected in particular directions, the number of collisions can approach the total number of elements in magnitude. For example, the elements at the bottom of the tower would collide with every element from upper floors. The stability check and PDE solver can also be completed with similar time complexities, however the granularity of the voxel grid is naturally extremely sensitive, $\mathcal{O}(n^3)$. But because the voxel grid is sparse, in the case of our frame experiments, more efficient implementation of an sparse LU solver should be used. Furthermore, we find that assessing the stresses at each disassembly action is typically unnecessary, and would be a more relevant feasibility metric determined by some heuristic, perhaps by analyzing the center of mass of the assembly, the spread of elements still touching the ground, or by partitioning the graph and hence the assembly in immediate proximity to the disassembly action. Additionally, to simulate the stresses with higher accuracy, the voxel size must be calibrated, leading to a trade off with computation time. For example, we use rectangular cross section elements here, but in previous experiments, wide flange section profiles were used, and would require much finer voxel dimensions to capture the web and flange interactions. Nevertheless for assemblies like the Simple Frame or Bridge experiment, we can compute feasible disassembly sequences fast enough to support on-the-fly path planning potentially in robotic applications.

Our next steps involve including manipulators or 'hands' as temporary elements in the assembly graph, and as additional collision objects for the disassembly planner to consider. In the case of the frame structures considered for our experiments, we would include two manipulators, one which grasps the element to remove, for example mounting to a crane, and another to perform the disassembly of joints. In prior research, we postulate the use of two robots to perform on site disassembly tasks for welded steel structures. Here, heuristics gleaned from practical experiments in disassembly undertaken in prior research have informed our approach to disassembly directions and candidate source nodes in the current work. The addition of such manipulators to the disassembly problem would also change the stability and structural feasibility checks. For example, a subassembly may not be unstable if supported by a crane mounted attachment or manipulator. This is demonstrated by Bruun et al. (2024) for example. The structural assessment would also be augmented in this case to include additional Dirichlet boundaries at attachment points, where displacement can be assumed to be zero. Another challenge we encounter in practice, but which is currently outside the scope of our model is the presence of residual stresses being released as elements are removed from an assembly. This would be a critically important next step in understanding the consequences of a disassembly action. Incorporating this level of detail would be necessary for manipulator path planning, and would involve probabilistic modeling of residual stresses in an assembly, for example in welded steel structures, as undertaken by Shayan et al. (2014).

While we demonstrate our sequencing strategy to successfully disassemble three prototypical frame structures, additional experiments on different topologies would support the generalize-ability of the research. In further iterations of the work, we intend to explore multi-layered and variegated geometries, instead of the simple extrusions which characterize our current experiments. Indeed, experimenting on further examples could lead to the development of a synthetic dataset, and potentially support the transition of this graph based method to a deep learning problem. Finally, we intend to weight the edges of our assembly graph to support modularity analysis and potentially partition our graph into subassemblies. Currently, our disassembly depth is equal to the depth of the graph, however this approach may not always be appropriate in practice. While our current application is intended to use meshes extracted from on-site robotic point cloud and RGB data, alternative applications may consider the BIM based data to provide heuristics on disassembly depth and costs associated with disassembly actions, such as demonstrated in prior art by Sanchez et al. (2019).

Conclusions

We present a graph based approach for disassembly sequencing with the addition of feasibility checks upon sta-

bility and internal stress states. By automating the encoding of spatial precedent relationship, we extend current graph based approaches to become more scalable and handle greater complexity in topology. Furthermore, we use heuristics informed by constraints in contemporary construction practice to guide our disassembly algorithm to find feasible sequences. We demonstrate our algorithm on three prototypical frame structures of increasing complexity and element count, and show reasonable computation times for up to 100 elements.

References

Akanbi, L. A., Oyedele, L. O., Omoteso, K., Bilal, M., Akinade, O. O., Ajayi, A. O., Davila Delgado, J. M., and Owolabi, H. A. (2019). Disassembly and deconstruction analytics system (D-DAS) for construction in a circular economy. Journal of Cleaner Production, 223:386–396.

Bourjault, A. (1988). Methodology of Assembly Automation: A New Approach. In Radharamanan, R., editor, Robotics and Factories of the Future '87, pages 37–45. Springer Berlin Heidelberg, Berlin, Heidelberg.

Bruun, E. P. G., Besler, E., Adriaenssens, S., and Parascho, S. (2024). Scaffold-free cooperative robotic disassembly and reuse of a timber structure in the ZeroWaste project. Construction Robotics, 8(2):20.

Erzmann, D., Dittmer, S., Harms, H., and Maaß, P. (2023). DL4TO : A Deep Learning Library for Sample-Efficient Topology Optimization. In Nielsen, F. and Barbaresco, F., editors, Geometric Science of Information, volume 14071, pages 543–551. Springer Nature Switzerland, Cham. Series Title: Lecture Notes in Computer Science.

Gengenbach, V., Nagel, H.-H., Tonko, M., and Schafer, K. (1996). Automatic dismantling integrating optical flow into a machine vision-controlled robot system. In Proceedings of IEEE International Conference on Robotics and Automation, volume 2, pages 1320–1325, Minneapolis, MN, USA. IEEE.

Ghandi, S. and Masehian, E. (2015). Review and taxonomies of assembly and disassembly path planning problems and approaches. Computer-Aided Design, 67-68:58–86.

Lambert, A. J. D. (2003). Disassembly sequencing: A survey. International Journal of Production Research, 41(16):3721–3759.

Moore, K. E., Gungor, A., and Gupta, S. M. (1998). A Petri net approach to disassembly process planning. Computers & Industrial Engineering, 35(1-2):165–168.

Sanchez, B. and Haas, C. (2018). A novel selective disassembly sequence planning method for adaptive reuse of buildings. Journal of Cleaner Production, 183:998–1010.

Sanchez, B., Rausch, C., and Haas, C. (2019). "Deconstruction programming for adaptive reuse of buildings". Automation in Construction, 107:102921.

Sanchez, B., Rausch, C., Haas, C., and Hartmann, T. (2021). A framework for BIM-based disassembly models to support reuse of building components. Resources, Conservation and Recycling, 175:105825.

Sanchez, B., Rausch, C., Haas, C., and Saari, R. (2020). A selective disassembly multi-objective optimization approach for adaptive reuse of building components. Resources, Conservation and Recycling, 154:104605.

Shayan, S., Rasmussen, K., and Zhang, H. (2014). Probabilistic modelling of residual stress in advanced analysis of steel structures. Journal of Constructional Steel Research, 101:407–414.

Smith, S., Smith, G., and Chen, W.-H. (2012). Disassembly sequence structure graphs: An optimal approach for multiple-target selective disassembly sequence planning. Advanced Engineering Informatics, 26(2):306–316.

Tian, Y., Xu, J., Li, Y., Luo, J., Sueda, S., Li, H., Willis, K. D. D., and Matusik, W. (2022). Assemble Them All: Physics-Based Planning for Generalizable Assembly by Disassembly. ACM Transactions on Graphics, 41(6):1–11. arXiv:2211.03977 [cs].

Wilson, R. H. and Latombe, J.-C. (1994). Geometric reasoning about mechanical assembly. Artificial Intelligence, 71(2):371–396.

Zeng, Y., Zhang, Z., Yin, T., and Zheng, H. (2022). Robotic disassembly line balancing and sequencing problem considering energy-saving and high-profit for waste household appliances. Journal of Cleaner Production, 381:135209.

Çetin, S., De Wolf, C., and Bocken, N. (2021). Circular Digital Built Environment: An Emerging Framework. Sustainability, 13(11):6348.