Karim, M. R., and Mohamed, Y. (2026). Physics-Based Simulation for Construction Activity Sequence Planning. *Lean Construction Journal* 2026 Special Issue CRC-CSCE pp 170-189 https://doi.org/10.60164/ojo96vp61
(Submitted 10Jan2026; Accepted 11Sep2026)
www.leanconstructionjournal.org

Physics-Based Simulation for Construction Activity Sequence Planning

Mohammad Rezaul Karim[1] and Yasser Mohamed[2]

Abstract

Research Question: How can construction activity sequences be generated automatically based on physical constructability constraints, rather than relying on predefined precedence logic embedded in conventional scheduling methods?

Purpose: To develop a physics-based simulation framework in which construction activity sequences emerge from physical constraint satisfaction.

Research Method: A BIM-derived industrial module is transformed into a physics-enabled representation by exporting component geometry and enriching it with construction metadata for rigid-body simulation. A PyBullet-based environment models gravity, contact, collision, support, and spatial clearance constraints. Activity sequences are generated using a baseline brute-force search in which component placements are iteratively evaluated for physical feasibility and reordered when violations occur. Seven module configurations of increasing complexity are tested to assess feasibility and computational behavior.

Findings: The results show that physically executable construction sequences can be generated through physics-based evaluation of stability and installation clearance without predefined precedence rules. Brute-force search scales poorly with increasing complexity, indicating limited practicality at larger scales.

Limitations: The implementation is limited by the poor scalability of brute-force search and simplified construction assumptions, including prefabricated components, idealized installation paths, and simplified connection behavior. Temporary works and non-physical planning constraints such as trade coordination, productivity, safety, and regulatory requirements are not modeled. In addition, missing physics-relevant attributes in BIM models require assumptions and manual data augmentation during BIM-to-simulation transformation.

Implications: The results suggest that physics-based simulation can function as a practical feasibility layer between design models and detailed schedules, allowing sequencing logic to be tested against real physical constraints before being finalized. This creates a pathway for integrating constructability checks earlier in planning, rather

[1] Ph.D Candidate, Department of Civil & Environmental Engineering, University of Alberta, 116 St and 85 Ave, Edmonton, Alberta, Canada, T6G 2R3 (corresponding author), mrkarim1@ualberta.ca

[2] Professor, Department of Civil & Environmental Engineering, University of Alberta, 116 St and 85 Ave, Edmonton, Alberta, Canada, T6G 2R3, yaly@ualberta.ca

http://creativecommons.org/licenses/by-nc-nd/4.0/

than addressing issues during execution.
Value for Practitioners: The approach offers a way to test and refine proposed sequences in a virtual environment, supporting early exploration of alternative construction strategies and reducing reliance on trial-and-error during planning. It can be used alongside existing tools to improve confidence in sequence feasibility prior to detailed scheduling and execution.
Keywords: Activity sequencing; physics-based simulation; constraint-satisfaction process; brute-force search
Paper type: Full paper

Introduction

Construction activity sequence planning is a structured approach that organizes the execution of construction activities by defining their order, dependencies, and constraints, ensuring a systematic and lean workflow for project implementation (Echeverry et al., 1991; Amer & Golparvar-Fard, 2019a; Hong et al., 2023). The process is part of construction project schedule development and begins with the development of a Work Breakdown Structure (WBS), which is a hierarchical framework that divides the project into deliverables or phases and work packages (Wang & Rezazadeh Azar, 2019). Activities are defined by decomposing work packages into discrete schedule activities required to produce the project deliverables, with each activity characterized by its scope of work, estimated duration, and associated resource requirements (Baker & Trietsch, 2009; PMI, 2021).

To derive these activities, a planner depends on predefined fragnets, standardized work templates, or automated methods that use natural language processing (NLP), large language models (LLM), and machine learning (ML); utilizing historical data, pattern recognition, and predictive modeling, complemented and verified by experience-based judgment (Amer et al., 2021). However, activity identification does not follow a fixed rule or universal standard and varies across a wide range of project types, industry standards, and planning methodologies (Fischer & Aalami, 1996). The level of detail and granularity required for defining construction activities is also influenced by various project-specific factors such as project complexity, construction methods, stakeholder requirements, and scheduling levels.

Since the primary goal of construction implementation planning is to transform designs into a physical structure, construction activities are inherently linked, directly or indirectly, to the realization of tangible building elements. Accordingly, construction activities represent abstractions of one or more component-level assembly operations, together with any required supporting operations such as temporary work, material handling, inspections, or quality control. Individual components constitute the smallest physically meaningful units of a constructed facility and serve as the fundamental production units of construction. These components can be aggregated into assemblies that define the scope of a construction activity. From this perspective, activity definition can be performed in a bottom-up manner by grouping related component-level assembly operations to satisfy planning and scheduling requirements (Faghihi, 2014). Consequently, sequencing construction activities implicitly require sequencing the underlying components and assemblies, and activity-level precedence relationships can be understood as abstractions of component-level physical and spatial dependencies.

http://creativecommons.org/licenses/by-nc-nd/4.0/

When sequencing construction activities to establish the precedence network for scheduling, four key factors are to be considered: (1) the physical and spatial relationships among structural components, (2) interactions and dependencies among construction trades, (3) the requirement for interference-free movement of materials, equipment, and workers on site and in place, and (4) building codes and safety regulations that prescribe or constrain the order of execution (Echeverry et al., 1991; Koo, 2004). Among these factors, physical interdependence among components exerts the strongest influence on activity sequencing, as it directly governs the realization of the built structure and therefore forms the basis of the sequencing framework (Echeverry et al., 1991).

The precedence network, or project network diagram, depicting activities and their logical precedence relationships, is a fundamental tool in the Critical Path Method (CPM). CPM analyzes construction activity sequences by identifying the critical path and activities that govern overall project duration, thereby enabling schedule optimization and effective project control (Russell et al., 2009; Bagshaw, 2021). However, CPM relies on predefined precedence network, requiring planners to manually assign dependencies among activities rather than having them automatically determined. To address this limitation, researchers have explored various approaches for automating dependency development. For instance, NLP and LLMs analyze unstructured textual data, such as project specifications and descriptions, to extract activities and infer logical sequences based on domain ontologies, knowledge graphs, and rule-based semantic representations. ML models also further enhance automation by using historical project data to predict task dependencies and durations. Moreover, when integrated with building information modeling (BIM), ML-based methods facilitate data extraction, activity generation, and automated sequencing, creating a more systematic approach to schedule development (Weldu, 2016; Amer et al., 2021; Al-Sinan et al., 2024).

Despite these advancements, significant limitations persist. Indeed, most current methods rely on large ML-based historical datasets of similar projects or predefined sequencing rules based on BIM, which often fail to capture project and site-specific constructability challenges, interference, and path obstructions. In this context, researchers have utilized 4D BIM simulations, aiming to identify schedule conflicts generated from CPM-based methods. These simulations primarily function as visualization tools, though, rather than dynamically generating or adjusting sequences that consider actual structural and environmental constraints in real-time (Koo & Fischer, 2000; de Vries & Harink, 2007; Staub-French et al., 2008; Kim et al., 2013).

With the rapid increase in computational capabilities, physics-based simulation provides a novel approach for the autonomous generation of activity sequences within a virtual physical environment, where constructability is ensured through the satisfaction of real-world physical constraints (Osorio-Sandoval et al., 2022; Xu et al., 2024). By modeling construction activity as physical actions of a rigid body governed by gravity, structural support, collision behavior, and spatial clearance requirement, sequencing feasibility can be evaluated directly from physical constraints rather than predefined logic (Wang et al., 2022; Tian et al., 2023; Liu & Negrut, 2021). Based on this formulation, construction sequencing is represented as a constraint satisfaction problem (CSP), where candidate actions constitute decision variables and physical laws and constructability requirements define the governing constraints.

Accordingly, a constructability-driven sequencing methodology based on a baseline brute-force search strategy is formulated, in which available actions are systematically

http://creativecommons.org/licenses/by-nc-nd/4.0/

examined, infeasible options are filtered through physical constraint checking, and valid construction sequences are incrementally generated. While intentionally simple, the approach provides a reference framework for benchmarking and for enabling the development of more efficient and scalable sequencing algorithms in future work.

Within this context, the paper introduces a physics-based simulation framework for automating construction activity sequencing, including the conversion of a 3D BIM model into a physics-enabled simulation environment. The implementation of the brute-force search strategy for sequence generation is demonstrated through a case study of an industrial module, and the results are used to assess performance, identify computational limitations, and discuss potential optimization strategies and future research directions.

Background

Across the project scheduling literature, activity sequencing is frequently presented as an integral part of schedule generation, even though sequencing and scheduling address fundamentally different analytical problems in construction project management research (Faghihi et al.,2015; Hong et al., 2023). Sequencing determines the required order of activities, whereas scheduling assigns these ordered activities to specific time periods subject to temporal and resource constraints (Koo, 2004). In practice, however, most scheduling models do not explicitly generate activity sequences. Instead, sequencing is treated as an external input to the scheduling algorithm, with precedence relationships either manually specified or inferred using data-driven and semantic approaches, rather than derived from explicit constraint-satisfaction considerations (Faghihi, 2014; Hong et al., 2023; Al-Sinan et al., 2024). Hence, these precedent relationships are typically treated as fixed inputs throughout the scheduling analysis. As a result, the analytical focus of most scheduling methods lies in time–resource allocation and optimization given an assumed logical structure, rather than in the derivation, validation, or adaptation of activity precedence relationships themselves (Faghihi, 2014; Amer et al., 2019b).

Empirical studies in construction planning show that activity sequencing relies heavily on practitioner tacit knowledge and expertise. Foundational work by Echeverry et al. (1991) demonstrated that schedulers define precedence relationships in project networks based on domain heuristics, including structural logic, accessibility, safety considerations, and trade coordination. To reduce reliance on tacit knowledge in automated planning processes, early research developed rule-based sequencing approaches that explicitly encode common precedence patterns, such as “foundations before superstructure” or “columns before beams”(Alshawi & Jagger, 1991). With advancements in semantic technologies, these rule-based methods were extended into ontology-driven knowledge graphs that represent construction entities and constraints as semantic relationships, enabling automated reasoning about feasible activity orders (Koo, 2004; Faghihi, 2014; Amer et al., 2021; Al-Sinan et al., 2024).

Although ontology-based models can represent high-level domain logic, they remain limited in their ability to capture construction conditions in sufficient operational detail. These systems depend on manually specified knowledge, which is often incomplete, difficult to maintain, and slow to adapt as project conditions evolve. Rule-based and ontology-driven systems are inherently static: once rules are defined, they are applied uniformly, despite constructability being influenced by transient and context-specific factors such as temporary works, partial assemblies, equipment reachability, workspace

http://creativecommons.org/licenses/by-nc-nd/4.0/

congestion, and evolving geometry. Consequently, knowledge-driven approaches primarily encode abstract logical relationships and still require extensive manual review and adjustment by practitioners to ensure that generated sequences are physically executable and constructible on site(Hong et al., 2023).

In parallel, ML and NLP approaches have been explored to automate sequencing by learning from historical project data. Liu & Luo (2026) revealed that ML models can learn typical activity templates, identify pairwise precedence relationships, and diagnose schedule-logic inconsistencies using large collections of CPM schedules. Deep learning–based sequence models further show promise in predicting successor activities with cross-project generalizability, while similarity-based methods enable the transfer of planning knowledge across related projects through WBS matching(Faghihi, 2014; Wang et al., 2020). Despite these advances, data-driven approaches operate primarily at semantic and statistical levels. They infer typical activity successions but cannot evaluate whether a predicted sequence is operationally feasible for a specific project or constructible under actual site conditions.

The widespread adoption of BIM has motivated geometry-based sequencing methods that link design information with planning workflows. Studies have proposed generating draft schedules by combining geometric attributes with construction rules aligning BIM elements with WBS structures for automated schedule generation (Desgagné-Lebeuf et al., 2019; Faghihi et al., 2015; Wang et al., 2020; Weber et al., 2020), and developing dynamic 4D simulations that regenerate sequences in response to design changes (Koo & Fischer, 2000). However, even within these approaches, sequencing logic is typically driven by predefined dependency relationships, constraint rules, or planning heuristics embedded within the scheduling or BIM environment, rather than by explicit physics-based evaluation of construction feasibility. In these settings, geometry-based reasonings are used to derive logical structural and spatial dependencies through rule-based geometric computations involving element coordinates, spatial orientation, adjacency, clearance, and topological relationships within 3D space. Nevertheless, such methods cannot evaluate physical behaviors such as load transfer, deformation, structural stability, time-dependent kinematic evolution, or the effects of temporary support conditions as construction progresses. Consequently, 4D BIM-based methods primarily support visualization and coordination and, on their own, are insufficient for generating construction sequences that satisfy real-world physical constructability constraints (Amer et al., 2019; Abioye et al., 2021).

In parallel, a separate stream of research applies physics-based simulation to construction operations, using engines such as PyBullet, Chrono, and Gazebo/ROS. Physics simulation has been used to analyze crane dynamics , motion planning for heavy equipment collision-risk analysis, cooperative lifting, and operator training(Naboni et al., 2021; Osorio-sandoval et al., 2022). These studies demonstrate that rigid-body and multibody simulation can accurately represent equipment kinematics, load swing, cable behavior, and spatial interference. However, this body of work primarily focuses on equipment-level analysis and operational safety, with physics-based simulation typically applied to evaluate or visualize predefined operations rather than to generate construction activity sequences.

In contrast, robotics and automated assembly research provides compelling examples of physics-based reasoning embedded directly within the sequencing process. Tian et al. (2022) introduced Assemble-Them-All, which uses physics-based disassembly to infer

http://creativecommons.org/licenses/by-nc-nd/4.0/

feasible assembly sequences by evaluating stability and contact constraints. Subsequent work of the authors integrates collision checking, gravitational stability analysis, and support constraints into a search algorithm to compute provably feasible assembly sequences for complex CAD models (Tian et al., 2023). Related research demonstrates how assembly constraints can be inferred from simulated work-cell variations, while task and motion planning frameworks incorporate multi-contact dynamics, friction, and force feasibility directly into planning algorithms. Physics-based reinforcement learning has further shown the ability to generate physically plausible action sequences through interaction with simulated environments (Liu & Negrut, 2021; Liu et al., 2025; Serrano-Ruiz et al., 2021). Collectively, these advances illustrate that embedding physics-based reasoning within the planning loop enables systematic elimination of infeasible actions and generation of sequences compatible with real-world constraints.

Across traditional automatic scheduling methods, rule-based logic, ML–driven inference, BIM-integrated workflows, and construction-oriented physics simulation, no existing construction sequencing approach explicitly incorporates real-world physical behavior to determine and enforce execution constraints during sequence generation. Instead, current methods encode logical rules, infer statistical dependencies, or apply geometrical reasoning only after activity sequences have been produced. As a result, construction sequences are rarely derived from ground-truth physical constraints encountered during execution, such as stability, load transfer, temporary support requirements, or equipment–structure interactions. By contrast, robotics research demonstrates that embedding physics-based reasoning directly within planning and search processes yields sequences that are constructible, safe, and physically valid.

Cumulatively, these findings demonstrate that real-world physical behavior is predominantly represented at a semantic or abstract level in existing approaches to construction sequence derivation. This limitation underscores the necessity of integrating virtual, physics-aware environments with suitable transformations of BIM representations to enable direct reasoning about physical feasibility during construction. Accordingly, there is a clear need for a physics-informed sequencing framework that: (1) transforms BIM components into simulation-ready entities endowed with appropriate rigid-body, material, and contact properties; (2) dynamically evaluates activity behavior within a physics-aware environment, including stability under gravity, adequacy of support conditions, collision avoidance, load-transfer behavior, and installation clearances; and (3) applies algorithmic reasoning to generate feasible sequences through incremental, constraint-driven exploration, ultimately yielding constructible activity sequences.

Objective

The objective of this study is to develop and demonstrate a physics-based simulation framework for construction activity sequence planning in which activity orders are generated by explicitly evaluating physical constructability through a simulation-based constraint-satisfaction process using a baseline search algorithm.

The framework represents BIM-derived construction components as simulation-ready entities within a virtual physical environment and evaluates activity feasibility incrementally based on stability under gravity, structural support conditions, and spatial installation constraints. Through iterative sequence evaluation, the approach systematically filters infeasible actions and retains only constructible sequences. The

http://creativecommons.org/licenses/by-nc-nd/4.0/

applicability of the framework is demonstrated using an industrial module case study, and the computational implications of physics-driven sequence generation are examined to inform future development of scalable and more efficient or intelligent sequencing methods.

Implementation of the Model

The implementation of the proposed framework conceptualizes construction sequencing as a physics-aware process (Yang et al., 2021; Liu et al., 2025) in which construction activities are explicitly associated with their corresponding structural components. Each activity is mapped to a specific structural element derived from a 3D BIM model developed in Autodesk Navisworks. To enable physics-based reasoning, the BIM representation is transformed into a simulation-ready rigid-body model. Geometric information is exported in *.obj* format and enriched with construction-related metadata, after which these data are encoded into the Universal Robot Description Format (URDF)(Zhu et al., 2023).

A physics-aware virtual environment is established using the PyBullet physics engine and configured to model real-world physical constraints including gravity, collision interactions, spatial limitations, rigidity, base support conditions, and inter-component connectivity behavior(Zhu et al., 2023). Within this environment, construction activities are evaluated dynamically by simulating the placement and interaction of URDF-defined components, enabling direct assessment of structural stability, installation feasibility, and spatial compatibility under realistic construction conditions.

An initial activity sequence, either arbitrary or is introduced into the physics-enabled environment and evaluated through iterative simulation. A brute-force search strategy is employed to explore feasible activity orderings. Sequences that violate physical constraints are systematically reordered and re-evaluated until a complete and physically viable construction sequence is identified. This integrated process combines BIM-based data transformation, URDF-based physical representation, and physics-driven computational sequencing to generate constructible activity sequences consistent with the modeled physical specifications of the structure.

Construction Algorithm

A construction algorithm encapsulates construction knowledge, which can be categorized into structural behaviour knowledge and construction method knowledge (de Vries & Harink, 2007). Structural behaviour is dictated by physical laws such as gravity, load distribution, force interactions, joint configuration, and material properties, while construction methods are influenced by material type, assembly techniques, construction equipment, regulatory codes, and safety considerations.

In this research, two fundamental considerations are adopted to address structural behaviour and construction methods. For structural behaviour, when considering gravitation, it is essential that any construction activity depicting the installation or construction of a physical component of the structure must be supported by a base or another component representing a construction activity within the sequencing process. Support may be provided in various forms including simple supports, which allow for the structure to rest on a surface, fixed joints, which require a single support point, and hinge supports, which necessitate two support points (Echeverry et al., 1991). In the case of

http://creativecommons.org/licenses/by-nc-nd/4.0/

construction methods, it is assumed that components are prefabricated according to their specifications and must be placed by equipment in their designated positions, following a predefined path. Afterward, connections are made by tradesmen. Therefore, each construction activity must have a clear installation path to be executed and, when tradesmen from two different disciplines are required to work in the same area, a time gap must be provided to avoid conflicts. In practical applications, construction methods can vary widely, leading to multiple valid construction sequences. However, all these sequences must adhere to structural stability, physical dependencies, and spatial constraints.

Methodology

In this study, a simplified methodology is developed, as shown in Figure 1, to generate an automated construction sequence from a 3D BIM model using the PyBullet physics simulation engine (Yang et al., 2021). The simulation engine requires that component entities be in URDF. Therefore, module elements such as steel structures, columns, beams, and pipe spools need to be converted into URDF before integration into the simulation environment. URDF provides a standardized XML-based representation of rigid-body geometry, material properties, joint definitions, and connectivity relationships required for physics simulation and is widely used in robotics and physics engines (Zhu et al., 2023). In this model, geometric and metadata information for structural components are extracted from a 3D BIM model available in *.nwd* format. The geometric representation of each component is exported as a triangulated mesh in *.obj* format. While geometric information is readily available from the BIM model, several physics-relevant attributes, such as mass, are not explicitly defined. These properties are therefore estimated using standard material unit weights and component volumes following common engineering practice and inserted into external datasets stored in *.csv* files. When available, equivalent information may alternatively be obtained from Industry Foundation Classes (IFC) (Tauscher et al., 2009; Zhu et al., 2023) models and integrated directly into the physics simulation. The complete set of geometric, physical, and metadata attributes used for URDF generation is summarized in Table 1.

Table 1: URDF XML Parameters Used for BIM-to-Physics Simulation

Attribute	Purpose in simulation	URDF element / field	Source in this study
Component identifier	Unique identification of each simulated entity	<robot name>, <link name>	BIM element ID with material type
Geometry mesh	Visual representation and collision detection	<visual><geometry><mesh>, <collision><geometry><mesh> <scale>	BIM export(.obj)
Position and orientation	Correct placement in simulation space	<origin xyz rpy>	BIM placement data
Mass	Gravity and stability evaluation	<inertial><mass>	Computed from volume (bounding box of .obj file x scale) × unit weight (simplified)

http://creativecommons.org/licenses/by-nc-nd/4.0/

Inertia tensor	Rigid-body rotational behavior	<inertial><inertia>	-
Joint type	Representation of support/connection behavior	<joint type>	Assumed by component type (fixed/revolute/null)
Joint axis	Defines allowable rotation direction	<axis xyz>	Derived from component orientation
Parent-child link	Support and connectivity definition	<parent>, <child>	Structural design logic from BIM

As joint behavior and explicit connectivity are not directly modeled in standard BIM geometry exports, simplified joint and parent–child link assumptions are adopted to support physics-based feasibility evaluation. Structural columns are modeled using fixed joints to represent rigid connections to foundations or supporting elements, while beams are connected using revolute joints in PyBullet to approximate hinged end conditions with rotational freedom about a single axis. Pipe spools and electrical cable trays are modeled as simply supported elements. Parent–child relationships between components are derived from the structural design logic embedded in the BIM model, defining support and load-transfer relationships within the simulation. These assumptions are consistent with typical construction-stage support conditions and allow evaluation of stability and support dependency without introducing unnecessary articulation complexity. Parameters not directly required for the current rigid-body simulation are assigned default values to ensure compatibility with the physics engine.

The extracted geometric, physical, joint, and parent–child connectivity data for each component are then converted into URDF files, which serve as the primary input for physics-based simulation and constraint evaluation during construction sequence generation.

http://creativecommons.org/licenses/by-nc-nd/4.0/

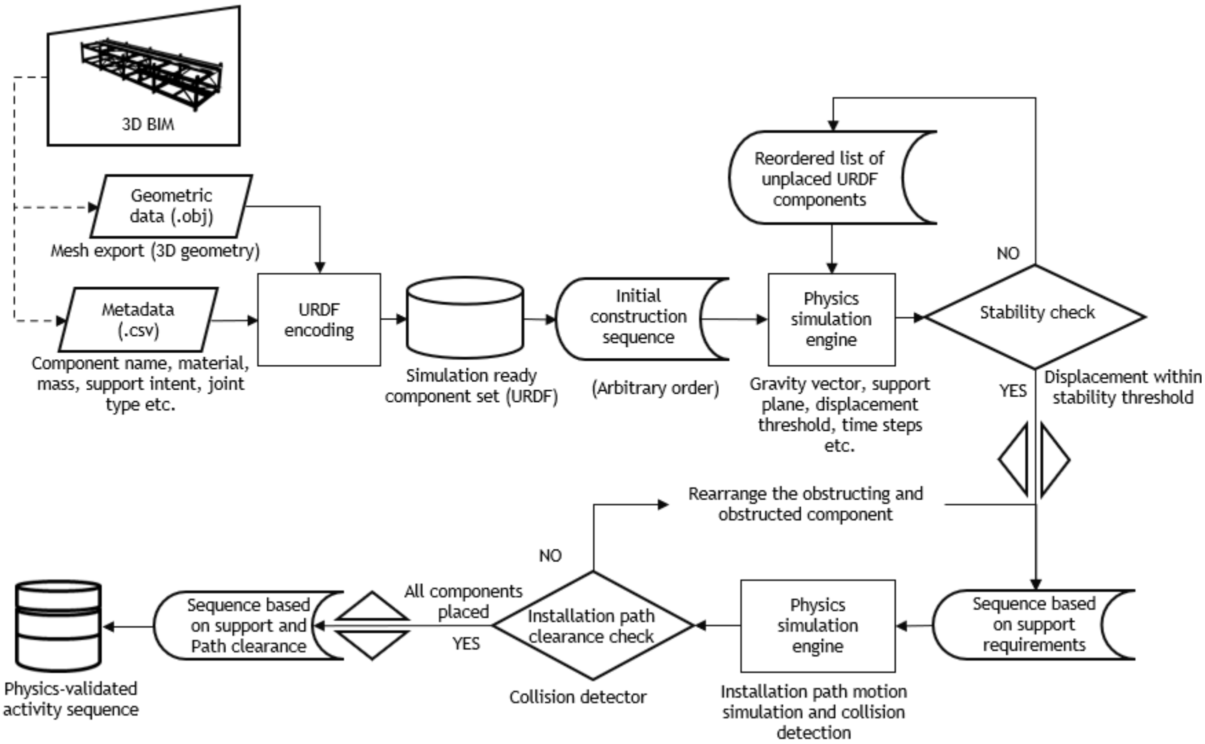


Figure 1. Physics-based Framework for Construction Activity Sequencing

If a component satisfies the prescribed stability criteria following placement, the sequencing process proceeds to evaluate the next component in the list. When a stability violation occurs, the corresponding URDF-defined component is removed from its current position and repositioned within the remaining unplaced component list, and the sequence is re-evaluated. This iterative procedure continues until all components satisfy structural support requirements. Through this support-driven sequencing mechanism, load-bearing dependencies are enforced implicitly, preventing components from being placed before adequate physical support is available. Once a structurally stable ordering is obtained, each component placement is further evaluated for installation feasibility through an installation path clearance assessment. This assessment simulates the motion of a component from its loading position to its designated installation location within the physics-enabled environment. If physical interference with previously placed components is detected, the attempted placement is deemed infeasible. The construction sequence is then modified by repositioning the obstructed component ahead of the obstructing component within the remaining unplaced list, thereby enforcing an explicit precedence relationship that resolves the detected interference. The revised sequence is subsequently subjected to repeated path motion simulation and collision checking. This iterative process continues until a collision-free installation path is identified for all components.

Upon satisfying both structural stability and installation path clearance constraints, the resulting sequence of URDF-defined components is classified as a physically admissible, physical-constraint-satisfied construction sequence. The generation of detailed construction schedules is outside the scope of this study; however, the validated sequence

http://creativecommons.org/licenses/by-nc-nd/4.0/

may be mapped to conventional planning representations to demonstrate compatibility with traditional scheduling workflows.

Throughout the sequencing process, computational performance indicators including execution time, number of sequencing iterations, and feasibility evaluation frequency are recorded. These metrics provide quantitative insight into the computational behavior of the framework and its scalability as structural complexity increases.

Brute-Force Search Performance Modeling

Brute-force search represents the most fundamental and assumption-free approach to construction sequence generation (Anantharaman et al., 1990; Schaeffer et al., 1993; Berliner, 1981). In this study, it is adopted as a baseline sequencing strategy to explicitly evaluate physical feasibility without relying on predefined precedence relationships, heuristics, or learned policies. The approach systematically enumerates all possible permutations of construction activities and evaluates each candidate sequence against governing feasibility constraints. Although computationally expensive, brute-force search provides completeness guarantees and serves as a transparent reference framework for characterizing the feasible sequencing space and assessing scalability limitations.

Consider a structure consisting of N construction activities, each corresponding to a component or a group of components that must be placed in a specific order to achieve a stable and constructible intermediate state. The sequencing process is governed by two key constraints: support requirement and installation path clearance. Support requirement warrants that a component can only be placed after its necessary supporting components are constructed to maintain structural stability. Here, $S\left(z_i\right) \geq S_{min}$, where $S\left(z_i\right)$ represents the total support provided by completed components beneath z_i, and S_{min} is the minimum required support threshold. If this condition is not met, component z_i cannot be placed.

The installation path clearance constraint ensures that there is an unobstructed space for the placement of a component. This condition is represented as $C\left(z_i\right) = 1$, where $C\left(z_i\right)$ is a binary function indicating whether a valid installation path exists (1 for clear, 0 for obstructed). If $C\left(z_i\right)$ = 0, the sequence must be iterated again, placing the obstructed component before the obstructing component, and requires reordering and re-evaluation of all remaining components to establish a feasible sequence. A valid construction sequence $S = \left(S_1, S_2, \ldots, S_N\right)$ must satisfy both constraints for all components, ensuring that each component is placed in a physically feasible and interference-free arrangement (Bernstein, 2005). In an unconstrained scenario, the number of possible sequences follows factorial growth represented by (Eq. 1):

$$[1]\ \Omega_{unconstrained} = N!$$

This expression accounts for all possible permutations of the *N* components. However, in practical construction scenarios, each component z_i is subject to structural stability requirements and installation path clearance conditions. These constraints significantly reduce the number of feasible sequences, which can be expressed as (Eq. 2):

http://creativecommons.org/licenses/by-nc-nd/4.0/

[2] $$\Omega_{constrained} = \frac{N!}{\prod_{i=1}^{N} k_i! + \prod_{i=1}^{N} c_i!}$$

The parameters k_i and c_i represent the number of immediate predecessor components required for structural stability before placing z_i and the number of components obstructing its path clearance, respectively.

Since both support and clearance constraints impose order dependencies on component placements, the number of feasible sequencing options decreases as the values of k_i and c_i increase. Accordingly, the reduction in valid sequences follows an inverse proportionality (Eq. 3):

[3] $$\Omega_{constrained} \propto \frac{1}{\prod_{i=1}^{N} k_i! + \prod_{i=1}^{N} c_i!}$$

The reduction in valid sequences significantly impacts computational search methods, particularly for brute-force. In a constrained scenario, the worst-case computational effort required to obtain a valid sequence through brute-force search is (Eq. 4):

[4] $$T_{brute-force} = N! - \Omega_{constrained}$$

In extreme cases where constraints are highly restrictive, such as when every component has a predetermined order of placement dictated by both structural and clearance constraints, the number of valid sequences may be reduced to a single feasible sequence, i.e., $\Omega_{constrained} = 1$.

From Eq. 4, it is evident that the computational effort increases factorially with *N*. Since brute-force search exhaustively tries all possible options, the constrained search space can lead to substantial computational inefficiencies. When constraints can be grouped into subgroups based on support dependencies and clearance requirements, the number of valid sequences is not merely adjusted by a factorial term but is further reduced by redefining the number of independent elements requiring sequencing. If *p* represents the number of constraint pairings or groups of interdependent components, then the effective number of independent elements to be sequenced is reduced from N to N' as shown in Eq. 5:

[5] $$N' = N - \sum_{j=1}^{p} \left(g_j - 1\right)$$

Here, g_j denotes the size of the j^{th} grouped set of interdependent components, and the subtraction reflects that, within each group, components follow a fixed internal sequence and do not contribute to independent permutations. Finally, the worst-case effort required to find a valid sequence is reduced to (Eq. 6):

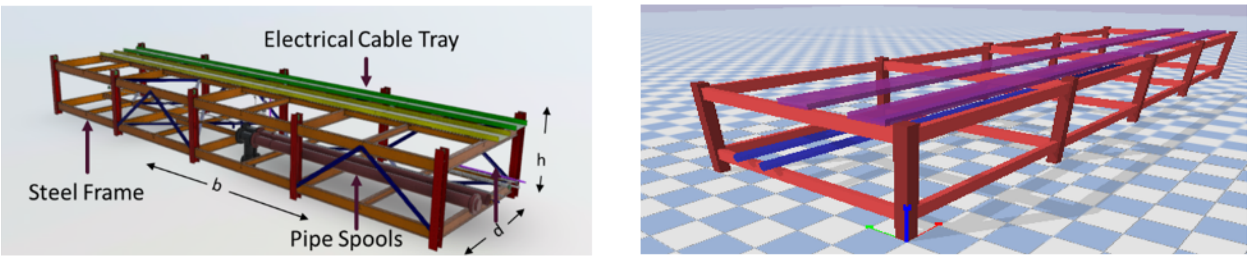
[6] $$T_{grouped} = N'! - \Omega_{constrained}$$

http://creativecommons.org/licenses/by-nc-nd/4.0/

Replacing $N!$ with $N^{'}!$ (in Eq. 4 to obtain Eq. 6) in the brute-force search space reduces factorial growth, lowers unsuccessful search attempts, and decreases computational effort. Since factorial growth is super-exponential, even a small reduction in N saves significant computation.

Case Studies and Results

The test case for applying the model involves generating a construction activity sequence as a prerequisite step toward schedule development for an industrial module, as illustrated in Figure 2, from *.nwd* format (Navisworks) file to be fabricated at a module fabrication yard. This module consists of three key structural components, steel frames, pipe spools, and electrical cable trays, each requiring installation by different trades.



(a) **(b)**

Figure 2. (a) A 3D BIM of Industrial Module, and (b) 3D URDF Representation of it in Physics Engine

In practice, the construction process follows a hierarchical assembly sequence to ensure structural stability and feasible installation. Assembling steel structures, pipe spools, and cable trays adheres to a well-defined sequence to maintain constructability and efficiency. The process begins with erecting the steel frame, which serves as the primary support for subsequent installations. Columns are first placed on prepared foundations or fixed support points, ensuring vertical alignment and securing them in place. These columns act as the primary load-bearing elements and must be fully stabilized before proceeding. Once the columns are set, horizontal beams are installed, connecting the columns to form a rigid frame. The connections between columns and beams are secured using either bolted or welded joints, depending on the design. The steel frame assembly follows a bottom-up approach, completing each level before advancing to the next. Bracing elements may also be added to enhance stability, particularly in taller or more complex structures.

After constructing the steel frame, pipe spools are installed. Their placement depends on the steel frame, as pipes require support from structural beams or dedicated pipe racks. Hangers and brackets are attached to the steel frame to secure the pipes. Following pipe installation, cable trays are mounted to create structured pathways for electrical wiring and instrumentation cables. These trays are typically supported by the steel frame, either attached directly to beams or secured using secondary support brackets. Since pipe spools and cable trays are assumed to be placed from above using crane support, their installation paths must remain unobstructed. Therefore, any

http://creativecommons.org/licenses/by-nc-nd/4.0/

upper-level steel beam should be installed only after the lower-level pipes or cable trays are positioned.

In a physics-based simulation model, the initial construction sequence is arbitrary. From this starting point, a viable sequence is iteratively generated based on stability constraints, support dependencies, and path clearance requirements. The model dynamically evaluates various placement possibilities and determines a feasible sequence where all components remain stable under gravity and structural forces while ensuring unobstructed installation paths. Each component, for example, column, beam, pipe spool, etc., is represented as a discrete entity using URDF files. These files define the component's geometry, mass connection behavior etc., enabling the simulation to accurately model physical interactions and constraints.

The simulation begins by placing components in their designated positions. If a component lacks the necessary structural support after placement, such as a beam being placed before its supporting columns, it is deemed unstable and moved to the end of the queue for repositioning in the sequence. Similarly, if the installation of a component is obstructed by an already placed element, the obstructed component is placed ahead of the obstructing component; the process continues until a clear installation path for all components is available. This iterative process refines the sequence until all components are positioned in a structurally sound manner without interference. In the first iteration of the simulation, columns are accepted first as the physical environment provides the necessary base support for their erection. Once stable columns are in place, beams are accepted next, forming a structural framework that supports additional elements. After the steel frame is sufficiently developed, the simulation evaluates when pipe spools can be installed, ensuring they are properly supported by beams or designated pipe racks. Likewise, cable trays are placed only when appropriate structural supports are available. Since pipe spools and cable trays are installed from above, the simulation verifies path clearance to prevent obstructions during placement. When higher-level steel beams block the positioning of pipe spools and cable trays, the installation of these beams is deferred until the pipe spools and cable trays are securely in place. Throughout the process, the model autonomously adjusts the sequence based on real-time stability validation and path clearance, preventing unsupported, unstable, or obstructed placements.

http://creativecommons.org/licenses/by-nc-nd/4.0/

Table 2: Computational Time for Sequence Generation Across Module Sizes

Module Name	Size (d x b x h)	Total number of components	Time to generate structurally stable seq. (sec)	Time to generate installation path cleared seq. (sec)	Total time (sec)
Module_1	4x1x1	41	0.36	0.69	1.05
Module_2	4x1x2	82	2.27	2.81	5.08
Module_3	4x2x2	138	6.66	10	16.66
Module_4	4x2x3	207	21.63	23.71	45.34
Module_5	4x3x3	291	44.09	49.33	93.42
Module_6	4x3x4	388	90.14	92.31	182.45
Module_7	4x4x4	500	152.17	160.42	312.59

Seven module sizes varying in complexity based on number of horizontal bays (d), longitudinal bays (b), and vertical levels (h), are simulated, as shown in Table 2. As the number of components of the modules increases, sequencing becomes more complex, revealing the limitations of brute-force search. The results in Table 2 illustrate how growing complexity affects sequencing efficiency and computational feasibility. A higher component count leads to a significant rise in computational time, highlighting the challenges of applying brute-force searches to large-scale construction scheduling.

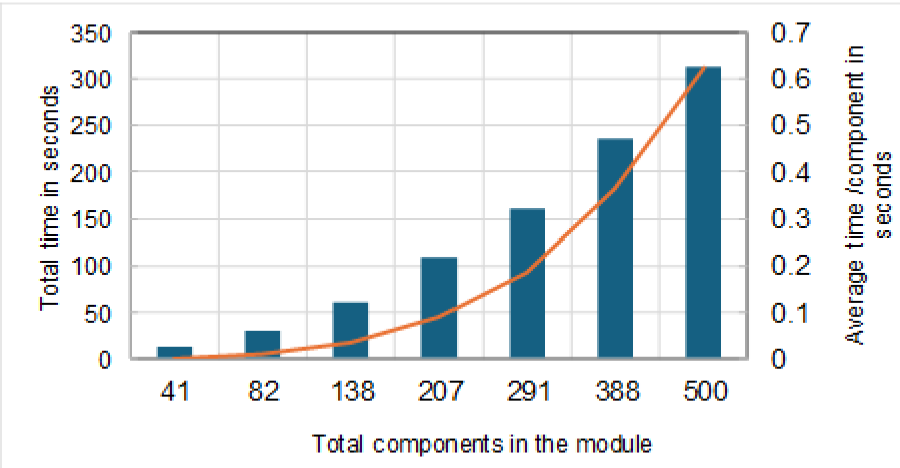


Figure 3. Computation Time vs. Increased Complexity (Number of Components)

Furthermore, Figure 3 illustrates the timing variation in the module structure as the number of components increases. As the number of components increases, the computational time per component grows significantly, demonstrating the exponential nature of brute-force search used for sequencing. When the component count in the structure is low (41 components), the average time required per component is 0.0256 seconds. However, as the structure becomes more complex with 500 components, the time per component increases drastically to 0.6252 seconds. This indicates that, as complexity

http://creativecommons.org/licenses/by-nc-nd/4.0/

rises, the sequencing process becomes more computationally demanding. In fact, the time per component in the high-complexity scenario is approximately 24 times higher than in the low-complexity case, highlighting the limitations of brute-force search methods. This sharp increase in computational demand confirms the necessity for more efficient sequencing algorithms, such as artificial Intelligent (AI)-driven optimization techniques to improve scalability and performance in large-scale construction sequencing.

Discussion

The results demonstrate that physics-based simulation can function as an effective feasibility reasoning mechanism for resolving physical constraints in construction activity sequencing. In the case study, by explicitly modeling structural stability and installation path clearance within a simulated physical environment, the proposed approach enables construction sequences to be evaluated based on their physical admissibility. This approach moves beyond predefined precedence rules by adopting constraint-driven feasibility reasoning, in which valid sequences are derived from the physical interactions between components and their environment, without the need to explicitly encode all constraints in advance.

In this context, the physics engine effectively embeds construction knowledge by encoding fundamental physical principles, such as gravity, collision avoidance, and stability directly into the sequencing process. As a result, feasibility is determined implicitly through simulation outcomes rather than through explicit rule enumeration. This behavior reinforces the suitability of physics-based simulation as a constructability-aware sequencing mechanism, particularly in modular and industrialized construction settings where physical interdependencies among components are not well pronounced and difficult to capture using traditional rule-based approaches.

Nonetheless, the computational performance analysis reveals a critical limitation inherent to brute-force search strategies. Although feasible sequences were successfully identified for all tested module sizes, which were relatively simple in nature, computation time increased exponentially with increasing structural complexity. These results indicate that brute-force search is not scalable for large-scale construction projects and should therefore be regarded primarily as a baseline approach. However, the mathematical formulation introduced in this study provides valuable insight into how constraint grouping and partial ordering can substantially reduce the effective problem size. This observation highlights a promising role for physics-based simulation as a ground-truth generator within hybrid sequencing frameworks, in which the physics engine ensures physical correctness and constructability compliance, while learning-based models or heuristic-guided algorithms efficiently explore the solution space. Such a hybrid integration presents a viable pathway toward scalable, constructability-aware sequencing solutions. At its current stage, the method is best positioned as a validation and benchmarking tool capable of generating high-fidelity feasibility data, rather than as a direct substitute for conventional project planning workflows.

In summary, several key limitations of the proposed approach should be acknowledged. First, computational scalability remains a major challenge due to the factorial growth associated with brute-force search as the number of components increases. Although physical constraints reduce the number of feasible sequences, identifying them through exhaustive iteration remains impractical for large assemblies.

http://creativecommons.org/licenses/by-nc-nd/4.0/

Second, the current implementation relies on simplified construction assumptions, including prefabricated components, idealized installation paths, and simplified connection behaviors. Key aspects of real construction processes, such as temporary support, partial assemblies, equipment kinematics, and detailed erection procedures—are not explicitly modeled. Third, gaps between BIM representations and physics-based simulation requirements pose challenges, as standard BIM models lack explicit information on load paths, joint behavior, or certain construction-relevant physical attributes essential for simulation, necessitating additional assumptions and preprocessing. Finally, the scope of constraints considered is limited to physical feasibility constraint satisfaction, while other influential factors such as trade coordination, productivity variability, safety requirements, and regulatory sequencing rules remain important directions for future research.

Conclusions

This study presents a physics-based framework for generating construction activity sequences through direct interaction with a simulated physical environment. Rather than relying on predefined precedence rules, the framework allows feasible sequences to emerge based on whether components can be physically placed, supported, and installed without interference. By representing BIM-derived components as simulation-ready physical entities, the approach embeds constructability directly into the sequencing process. The developed formulation, implementation, and case study demonstrate that physics-based simulation can serve as a systematic and reproducible mechanism for generating physically feasible construction sequences, providing a foundation for more advanced constructability-aware planning methods.

Future research should focus on extending the framework toward greater scalability and practical use. One important direction is the development of more efficient sequence generation strategies that exploit feedback from the simulation environment, reducing reliance on exhaustive search while retaining physical correctness. Another avenue involves enhancing the realism of the simulated environment by incorporating construction equipment behavior, staged assembly conditions, and evolving site constraints. Strengthening interoperability between BIM models and physics-based simulations is also essential to improve automation and reduce modeling effort. Finally, expanding the framework to integrate physical feasibility with broader planning considerations such as scheduling performance, resource coordination, and safety objectives would increase its applicability to real-world construction projects.

Acknowledgments

The authors gratefully acknowledge the support of PCL Industrial Management Inc., the Construction Innovation Center (CIC) at the University of Alberta, and the Natural Sciences and Engineering Research Council of Canada (NSERC) for providing data access and financial support for this research.

References

Abioye, S. O., Oyedele, L. O., Akanbi, L., Ajayi, A., Davila Delgado, J. M., Bilal, M., Akinade, O. O., & Ahmed, A. (2021). Artificial intelligence in the construction

http://creativecommons.org/licenses/by-nc-nd/4.0/

industry: A review of present status, opportunities and future challenges. *Journal of Building Engineering, 44*, 103299.
Al-Sinan, M. A., Bubshait, A. A., & Aljaroudi, Z. (2024). Generation of construction scheduling through machine learning and BIM: A blueprint. *Buildings, 14*(4), 934.
Alshawi, M., & Jagger, D. (1991). An expert system to assist in generating and scheduling construction activities. *Computers & Structures, 40*(1), 53–58.
Amer, F., & Golparvar-Fard, M. (2019a). Formalizing construction sequencing knowledge and mining company-specific best practices from past project schedules. *Journal of Construction Engineering and Management, 145*(7).
Amer, F., & Golparvar-Fard, M. (2019b). Automatic understanding of construction schedules: Part-of-activity tagging. *Proceedings of the European Conference on Computing in Construction*, 190–197.
Amer, F., Koh, H. Y., & Golparvar-Fard, M. (2021). Automated methods and systems for construction planning and scheduling: Critical review of three decades of research. *Journal of Construction Engineering and Management, 147*(7).
Anantharaman, T. S., Campbell, M. S., & Hsu, F.-H. (1990). Singular extensions: Adding selectivity to brute-force searching. *Artificial Intelligence, 43*(1), 99–110.
Bagshaw, K. B. (2021). New PERT and CPM in project management with practical examples. *American Journal of Operations Research, 11*(4), 215–226.
Baker, K. R., & Trietsch, D. (2009). *Principles of sequencing and scheduling*. John Wiley & Sons.
Berliner, H. J. (1981). An examination of brute force intelligence. *Proceedings of the International Joint Conference on Artificial Intelligence*, 581–587.
Bernstein, D. J. (2005). Understanding brute force. In *Proceedings of the ECRYPT Workshop on Symmetric Key Encryption*.
de Vries, B., & Harink, J. M. J. (2007). Generation of a construction planning from a 3D CAD model. *Automation in Construction, 16*(1), 13–18.
Desgagné-Lebeuf, A., Lehoux, N., Beauregard, R., & Desgagné-Lebeuf, G. (2019). Computer-assisted scheduling tools in the construction industry: A systematic literature review. *IFAC-PapersOnLine, 52*(13), 1843–1848.
Echeverry, D., Ibbs, W., & Kim, S. (1991). Sequencing knowledge for construction scheduling. *Journal of Construction Engineering and Management, 117*(1), 118–130.
Faghihi, V. (2014). *Automated and optimized project scheduling using BIM* (Doctoral dissertation). University of Illinois at Urbana-Champaign.
Faghihi, V., Nejat, A., Reinschmidt, K. F., & Kang, J. H. (2015). Automation in construction scheduling: A review of the literature. *International Journal of Advanced Manufacturing Technology, 81*(9–12), 1845–1856.
Fischer, M., & Aalami, F. (1996). Scheduling with computer-interpretable construction method models. *Journal of Construction Engineering and Management, 122*(4), 337–347.
Hong, Y., Xie, H., Agapaki, E., & Brilakis, I. (2023). Graph-based automated construction scheduling without the use of BIM. *Journal of Construction Engineering and Management, 149*(2).
Kim, H., Anderson, K., Lee, S., & Hildreth, J. (2013). Generating construction schedules through automatic data extraction using open BIM technology. *Automation in Construction, 35*, 285–295.
Kartam, N., & Levitt, R. E. (1990). Intelligent planning of construction projects. *Journal of Construction Engineering and Management, 116*(4), 596–615.

http://creativecommons.org/licenses/by-nc-nd/4.0/

Koo, B., & Fischer, M. (2000). Feasibility study of 4D CAD in commercial construction. *Journal of Construction Engineering and Management, 126*(4), 251–260.
Koo, B. (2004). Formalizing construction sequencing constraints for rapid generation of schedule alternatives (Doctoral dissertation). Stanford University.
Liu, C. K., & Negrut, D. (2021). The role of physics-based simulators in robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 4, 35–58.
Liu, R., Chen, A., & Zhao, W. (2025). Physics-aware combinatorial assembly sequence planning using data-free action masking. *IEEE Robotics and Automation Letters, 10*(5), 4882–4889.
Liu, Y., & Luo, H. (2026). An efficient machine learning-based model for duration prediction of construction tasks with large-scale datasets. Advanced Engineering Informatics, 69(Part A), 103820.
Naboni, R., Kunic, A., Kramberger, A., & Schlette, C. (2021). Design, simulation, and robotic assembly of reversible timber structures. *Construction Robotics, 5*(1), 13-22.
Osorio-Sandoval, C. A., Tizani, W., Pereira, E., Ninić, J., & Koch, C. (2022). Framework for BIM-Based Simulation of Construction Operations Implemented in a Game Engine. *Buildings, 12*(8), 1199.
Project Management Institute (PMI). (2021). *A guide to the project management body of knowledge (PMBOK® Guide)* (7th ed.). Project Management Institute.
Russell, A., Staub-French, S., Tran, N., & Wong, W. (2009). Visualizing high-rise building construction strategies using linear scheduling and 4D CAD. *Automation in Construction, 18*(2), 219–236.
Schaeffer, J., Lu, P., Szafron, D., & Lake, R. (1993). A re-examination of brute-force search. *AAAI Technical Report FS-93-02*.
Serrano-Ruiz, J. C., Mula, J., & Poler, R. (2021). Smart manufacturing scheduling: A literature review. *Journal of Manufacturing Systems*, 61, 265–287.
Staub-French, S., Russell, A., & Tran, N. (2008). Linear scheduling and 4D visualization. *Journal of Computing in Civil Engineering, 22*(3), 192–205.
Tauscher, E., Mikulakova, E., Beucke, K., & König, M. (2009). Automated generation of construction schedules based on the IFC object model. In *Proceedings of the Construction Research Congress* (pp. 719–728). ASCE.
Tian, Y., Willis, K. D. D., Al Omari, B., Luo, J., Ma, P., Li, Y., Javid, F., Gu, E., Jacob, J., Sueda, S., Li, H., Chitta, S., & Matusik, W. (2024). ASAP: Automated sequence planning for complex robotic assembly with physical feasibility. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 4380–4386). IEEE.
Tian, Y., Xu, J., Li, Y., Luo, J., Sueda, S., Li, H., Willis, K. D. D., & Matusik, W. (2022). *Assemble them all: Physics-based planning for generalizable assembly by disassembly*. ACM Transactions on Graphics, 41(6), Article 278.
Wang, H. W., Lin, J. R., & Zhang, J. P. (2020). Work package-based information modeling for resource-constrained scheduling of construction projects. *Automation in Construction, 109*, 102958.
Wang, J., Li, Y., Gao, R. X., & Zhang, F. (2022). Hybrid physics-based and data-driven models for smart manufacturing: Modelling, simulation, and explainability. *Journal of Manufacturing Systems, 63*, 381–391.
Wang, Z., & Rezazadeh Azar, E. (2019). BIM-based draft schedule generation in reinforced concrete-framed buildings. *Construction Innovation, 19*(2), 280–294.

http://creativecommons.org/licenses/by-nc-nd/4.0/

Weber, J., Stolipin, J., Jessen, U., König, M., & Wenzel, S. (2020). Rule-based generation of assembly sequences for simulation in large-scale plant construction. In *Proceedings of the 37th International Symposium on Automation and Robotics in Construction (ISARC)* (pp. 155–162). Kitakyushu, Japan.

Weldu, Y. W. (2016). *Automated generation and visualization of initial construction schedules from building information models* (Doctoral dissertation). Louisiana State University.

Xu, L., Veeramani, D., & Zhu, Z. (2024). Automated physics-based modeling of construction equipment through data fusion. *Automation in Construction, 168(1)*, 105880.

Yang, X., Ji, Z., Wu, J., & Lai, Y.-K. (2021). An open-source multi-goal reinforcement learning environment for robotic manipulation with PyBullet. In *Towards Autonomous Robotic Systems* (pp. 14-24). Springer.

Zhu, A., Pauwels, P., & de Vries, B. (2023). Component-based robot prefabricated construction simulation using IFC-based building information models. *Automation in Construction, 152*, 104899.

http://creativecommons.org/licenses/by-nc-nd/4.0/