A Decision-Oriented Synthesis of AI-Based Assembly Sequence Planning for Construction Automation

Anas Itani[1], Mohamed Al-Hussein[1], and Simaan Abourizk[1]

[1]Department of Civil and Environmental Engineering, University of Alberta, Canada
aitani@ualberta.ca, malhussein@ualberta.ca, abourizk@ualberta.ca

Abstract

Assembly Sequence Planning (ASP) is a critical decision-making problem in construction automation, affecting productivity, cost, constructability, and system robustness. ASP is NP-hard (Non-deterministic Polynomial-time hard) due to the factorial growth in feasible assembly sequences as product complexity increases. While artificial intelligence (AI) and soft computing methods have been widely applied to ASP, existing research remains fragmented, algorithm-centric, and largely rooted in manufacturing contexts, limiting its transferability to construction automation. This paper presents a construction-oriented synthesis of AI-based ASP approaches, emphasizing decision-level insights rather than algorithmic mechanics. This paper reviews ASP modelling strategies, constraints, assembly scales, and objectives from a construction perspective. AI-based optimization methods are organized into major algorithm families and evaluated according to scalability, computational effort, and construction suitability. The paper further discusses emerging considerations such as parallel assembly, BIM-enabled digital twins, and data-driven learning approaches within a construction automation context. Key research gaps are prioritized to guide future research and development. The synthesis provides guidance for researchers and practitioners seeking to design scalable, realistic, and implementable ASP solutions for automated and industrialized construction.

Keywords –

Assembly Sequence Planning; Construction Automation; Artificial Intelligence; Prefabricated and Modular Construction; Optimization Methods

1 Introduction

The increasing adoption of automation, prefabrication, and modular construction is transforming how construction systems are designed, planned, and executed. As construction adopts manufacturing-inspired production paradigms, assembly-related decisions, such as component ordering, assembly directions, and tool usage, have become critical determinants of productivity, cost efficiency, and constructability [1,2]. Assembly Sequence Planning (ASP) addresses these challenges by determining feasible and efficient assembly sequences under technical and operational constraints.

ASP is NP-hard combinatorial optimization problem, as the number of feasible sequences grows factorially with the number of components [3,4]. In construction applications, this complexity is amplified by large-scale components, heterogeneous materials, diverse connection types, and varying degrees of automation [5]. Poor sequencing can result in excessive tool changes, frequent reorientation, unbalanced workloads, and increased human or robotic effort, reducing overall system efficiency [6,7].

Figure 1 positions ASP within the construction automation decision framework, highlighting its role as a coordinating function linking product design, production planning, and execution. Unlike manufacturing ASP (rigid, sequential, controlled), construction involves parallelism, variability, and human–robot interaction [8,9]. These characteristics increase planning complexity and limit the transferability of manufacturing-oriented solutions.

This distinction is illustrated in Figure 2, which contrasts manufacturing- and construction-oriented assembly systems in terms of component scale, assembly parallelism, and operational variability, clarifying the need for construction-specific modelling assumptions and decision criteria.

To address ASP's combinatorial complexity, AI and soft computing methods, such as genetic algorithms, ant colony optimization, particle swarm optimization, simulated annealing, and hybrid approaches, have been widely applied [12–14]. However, much of the literature remains algorithm-centric, emphasizing method-specific implementations rather than decision-level applicability. In addition, several construction-critical considerations remain underrepresented, including flexible or deformable components, ergonomic and human–machine interaction constraints, sustainability objectives, and explicit treatment of parallel or collaborative assembly processes [13]. These limitations are synthesized in Figure 3.

In this context, this paper presents a focused synthesis of AI-based ASP approaches from a construction automation perspective, emphasizing decision-level insights rather than algorithmic mechanics. The paper: (i) reframes ASP as a construction decision problem spanning design, planning, and execution; (ii) introduces assembly scale as a key determinant of modelling and algorithm suitability; (iii) synthesizes ASP modelling approaches and constraint types relevant to construction systems; (iv) compares AI-based optimization methods according to their suitability across different assembly scales; and (v) identifies prioritized research gaps to guide future development of practical ASP solutions for construction automation.

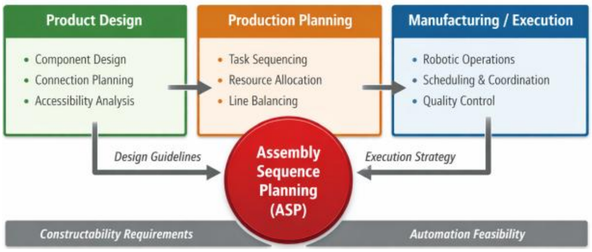

Figure 1. Role of assembly sequence planning in construction automation

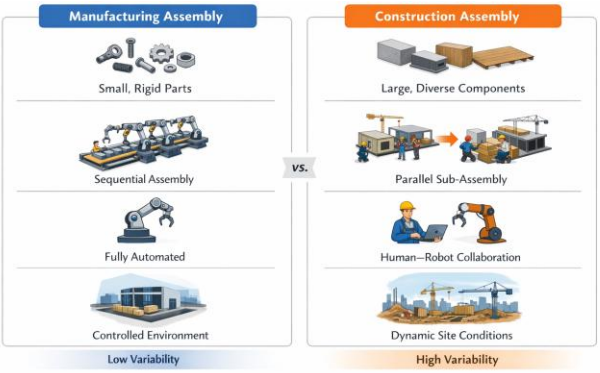

Figure 2. Comparison of manufacturing- and construction-oriented assembly systems

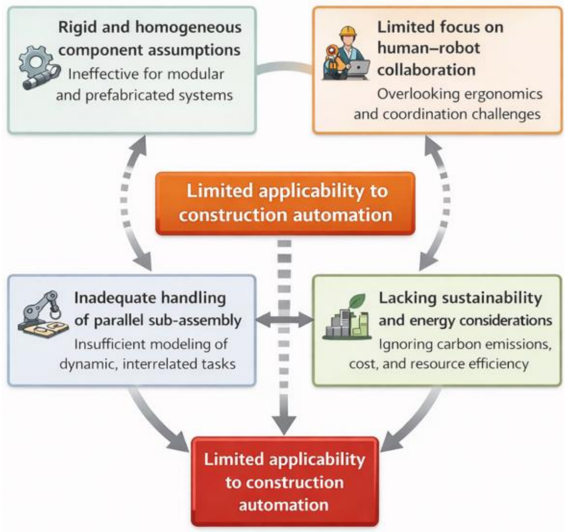

Figure 3. ASP limitation in construction automation research

This study adopts a structured narrative review approach. Literature was identified from Scopus, Web of Science, and Google Scholar using keywords such as 'assembly sequence planning' and 'construction automation'. The review focuses on studies (2000–2024) relevant to ASP modelling, constraints, and optimization, covering approximately 120 papers.

2 Assembly Sequence Planning as a Construction Decision Problem

ASP has traditionally been formulated as a combinatorial optimization problem within manufacturing-oriented research, where it is treated as a technical sequencing task aimed at generating feasible or near-optimal assembly orders under predefined precedence constraints [3,4,8]. While this formulation has enabled significant algorithmic advances, it captures only part of ASP's role in construction automation.

In construction-oriented systems, ASP is more appropriately viewed as a system-level decision problem spanning product design, production planning, and execution. Construction assemblies differ fundamentally from manufacturing assemblies due to larger components, heterogeneous materials, variable environments, and differing levels of automation [1,2,5]. Consequently, ASP decisions influence not only feasibility but also constructability, system efficiency, and automation readiness. Across the construction lifecycle, ASP acts as a coordinating function, shaping component segmentation and accessibility at the design stage [6,10], interacting with resource allocation and line balancing during production planning [2,11], and guiding task scheduling and human–robot coordination during execution [13].

The limited transferability of existing ASP research largely stems from manufacturing-oriented assumptions, including rigid components, sequential processes, and controlled environments [8,9,12]. These assumptions rarely hold in construction, where parallel sub-assembly, distributed production, and semi-automated operations are common, and where human–robot collaboration introduces additional ergonomic and safety constraints [2,13]. In prefabricated and modular construction, parallel assembly further increases ASP complexity by introducing concurrency, synchronization, and resource-sharing constraints [11,12], requiring ASP to address not only component sequencing but also coordination across parallel sub-assemblies.

Reframing ASP as a construction decision problem has direct implications for modelling and optimization. Models must capture system-level characteristics such as parallelism, resource coupling, and human–robot interaction, while optimization objectives must extend beyond local efficiency to include constructability, safety, and robustness [6,7]. Algorithm selection should therefore be guided by assembly characteristics and system requirements rather than algorithm popularity alone. This reframing provides the foundation for the next section, which introduces assembly scale and system characteristics as key determinants of modelling strategies and AI-based optimization suitability in construction-oriented ASP.

3 Assembly Scale and System Characteristics in Construction-Oriented ASP

Building on the decision-oriented framing introduced in Section 2, this section operationalizes that perspective by formalizing assembly scale as a determinant of modelling and optimization suitability. Assembly scale remains an underexplored dimension in ASP research, where modelling and optimization strategies are often assumed to scale uniformly with problem size. In construction automation, however, assembly scale fundamentally influences solution space growth, constraint dominance, modelling feasibility, and algorithm suitability, making its explicit consideration essential for practical ASP solutions.

3.1 Definition of Assembly Scale in Construction Contexts

In manufacturing-oriented ASP research, assembly scale is often defined solely by the number of components. While component count remains relevant, it is insufficient to characterize construction assemblies. In construction automation, assembly scale is more appropriately defined by the combined effects of component count, presence of sub-assemblies, degree of parallelism, and level of system coordination required.

Based on the literature and construction practice [2,5,11], construction-oriented ASP problems can be classified according to assembly scale, as summarized in Table 1, providing a decision-oriented basis for selecting modelling and optimization strategies.

While assembly scale is presented conceptually, future work should define quantitative thresholds based on component count, degree of parallelism, and interaction density to enable more objective classification across construction scenarios.

Table 1. Assembly scale classification for construction-oriented ASP

Assembly Scale	Typical Characteristics	Implications for ASP
Small	Few components; mostly sequential	Exact or simple meta-heuristics feasible
Medium	Multiple sub-assemblies; mixed sequencing	Evolutionary or swarm methods effective
Large	Parallel sub-assemblies; system coupling	Hierarchical, hybrid, or memetic methods

3.2 ASP Solution Space Growth and Assembly Scale

Sequential assembly represents the baseline ASP case, where components are assembled one at a time under precedence constraints; parallel assembly generalizes this formulation by allowing concurrent sub-assemblies subject to synchronization and resource constraints

For a purely sequential assembly of n components, the theoretical upper bound of feasible sequences is:

$$S_{seq} = n! \qquad (1)$$

For a system composed of k independent sub-assemblies with component counts $n_1, n_2, \dots, n_k$, the number of possible interleavings can be approximated as:

$$S_{par} = \frac{(n_1+n_2+\cdots+n_k)!}{n_1!+n_2!+\cdots+n_{k!}} \qquad (2)$$

Parallel sub-assembly, common in prefabricated and modular construction, therefore, expands the solution space dramatically relative to sequential assemblies, explaining the limited scalability of many manufacturing-oriented ASP approaches [12,14].

3.3 Scale-Dependent Dominance of ASP Constraints

Assembly scale strongly influences which constraints dominate ASP outcomes. In small-scale assemblies, feasibility constraints, such as precedence, geometric, and stability constraints, are dominant, as ensuring feasible assembly orders constitutes the primary planning challenge [20,21]. In medium-scale assemblies, optimization-oriented constraints, including assembly direction changes, tool changeovers, and operation-type variation, increasingly influence sequence quality and cycle time [6,7].

In large-scale construction assemblies, typical of modular and prefabricated systems, system-level constraints often dominate. These include workstation coordination, resource sharing, ergonomic limits in human–robot collaboration, and synchronization of parallel tasks [13,15]. At this scale, local sequencing decisions can significantly affect overall system performance. This transition in constraint dominance is illustrated in Figure 4.

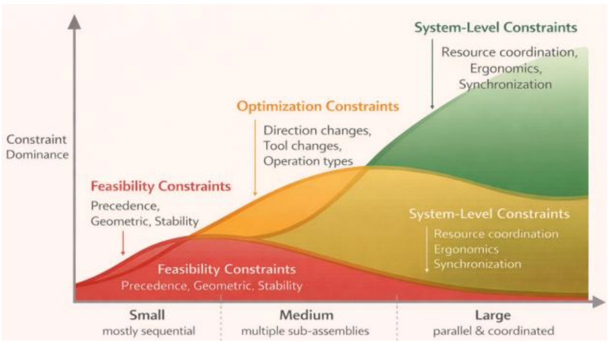

Figure 4. Dominant ASP constraints across assembly scales

3.4 Implications for Modelling and Algorithm Selection

Assembly scale directly informs the suitability of ASP modelling approaches. Part-based modelling offers high fidelity and is effective for small-scale assemblies but becomes computationally prohibitive as scale increases. Task-based modelling reduces problem dimensionality through operation aggregation [15], while connector-based modelling is well suited to large-scale modular systems, where standardized interfaces support hierarchical planning [18,19].

The interaction between assembly scale, modelling complexity, and constraint dominance also guides algorithm selection: small-scale assemblies can be addressed using evolutionary or physics-inspired methods, medium-scale assemblies benefit from swarm intelligence approaches, and large-scale assemblies typically require hybrid or memetic strategies that combine global search with hierarchical decomposition. By explicitly incorporating assembly scale into ASP formulation, algorithm selection becomes a decision-driven process rather than a methodological preference, providing a foundation for the comparative analysis of AI-based ASP methods in the next section.

4 ASP Modelling Approaches and Constraint Formulation

ASP performance and applicability in construction automation are strongly influenced by how the assembly problem is modelled and how constraints are formulated. Modelling choices define the structure of the search space, the types of constraints that can be represented, and the computational feasibility of optimization. In construction-oriented ASP, models must account for large-scale components, heterogeneous connections, and parallel assembly processes, which are often inadequately captured by manufacturing-focused formulations.

4.1 Overview of ASP Modelling Approaches

Existing ASP research typically adopts one of three modelling approaches: part-based, task-based, or connector-based representations [15,16,18]. These approaches differ in abstraction level and exhibit distinct advantages and limitations when applied to construction assemblies.

4.1.1 Part-Based Modelling

Part-based modelling treats individual components as the fundamental assembly units. Precedence relationships are derived from geometric interference, mechanical feasibility, and stability conditions [16,17]. This approach offers high modelling fidelity and is well suited to small-scale assemblies where component interactions remain manageable.

In construction automation, part-based models are commonly used for detailed prefabrication tasks or small modular assemblies. However, as assembly scale increases, the precedence graph grows rapidly, leading to significant computational challenges. As a result, part-based modelling becomes impractical for large-scale construction assemblies with extensive parallelism.

4.1.2 Task-Based Modelling

Task-based modelling abstracts the assembly process into sequences of operations rather than individual components. Tasks may represent fastening, positioning, or tool-specific actions [15]. This abstraction reduces problem dimensionality and enables direct incorporation of operational constraints such as tool usage and assembly direction.

Task-based models are particularly suitable for medium-scale assemblies, where multiple sub-assemblies exist but full part-level modelling is computationally prohibitive. However, this abstraction may obscure detailed geometric constraints, which can be critical for large or irregular construction components.

4.1.3 Connector-Based Modelling

Connector-based modelling organizes the assembly process around connection mechanisms such as bolts, welds, or standardized interfaces [18,19]. Components associated with a common connector are grouped, enabling hierarchical or multi-stage planning.

This approach aligns well with modular and prefabricated construction systems, where standardized connections are prevalent. Connector-based modelling supports scalable ASP by reducing effective problem size and enabling decomposition into manageable sub-problems, although it often requires more complex model formulation and multi-level optimization.

4.2 Constraint Categories in Construction-Oriented ASP

ASP constraints can be broadly classified into feasibility constraints and optimization-oriented constraints [6,15], with their relative importance varying by assembly scale, as discussed in Section 3.

4.2.1 Feasibility Constraints

Feasibility constraints determine whether an assembly sequence is physically and functionally admissible and typically include precedence constraints that enforce required assembly orderings [20], geometric constraints that prevent collisions and ensure accessibility [21], and stability constraints that maintain structural integrity during intermediate assembly states [22]. These constraints dominate small-scale assemblies and must be satisfied before optimization objectives are considered.

4.2.2 Optimization-Oriented Constraints

Optimization-oriented constraints influence the efficiency and quality of feasible sequences. In construction automation, commonly considered constraints include assembly direction changes, tool changeovers, operation-type variation, and task sequencing consistency. These factors directly affect cycle time, energy consumption, and operational complexity, particularly in robotic and semi-automated systems [6,7].

4.2.3 System-Level Constraints

In large-scale construction assemblies, system-level constraints become increasingly important. These include workstation coordination, resource sharing across parallel tasks, ergonomic and safety constraints in human–robot collaboration, and synchronization of parallel sub-assemblies. Such constraints link ASP with higher-level planning functions and require models capable of representing system-wide interactions [13,15].

System-level constraints such as ergonomic limits, resource sharing, and workstation coordination can be integrated through multi-layered modelling approaches, where local sequencing decisions are coupled with higher-level scheduling and resource allocation models, enabling synchronized optimization across parallel sub-assemblies.

4.3 Relationship Between Assembly Scale, Modelling, and Constraints

The interaction between assembly scale, modelling approach, and constraint formulation is central to construction-oriented ASP. Part-based models are most effective when feasibility constraints dominate, task-based models balance feasibility and optimization concerns, and connector-based models support system-level constraint representation in large-scale assemblies. Table 2 summarizes the suitability of ASP modelling approaches as a function of assembly scale and dominant constraint types.

Table 2. Relationship between assembly scale, modelling approach, and dominant constraints

Assembly Scale	Dominant Constraints	Suitable Modelling Approach
Small	Feasibility	Part-based
Medium	Optimization-oriented	Task-based
Large	System-level	Connector-based/Hierarchy

4.4 Implications for Construction Automation

The analysis highlights that no single ASP modelling approach is universally applicable across construction scenarios. Effective ASP solutions must align modelling fidelity with assembly scale and dominant constraints. Overly detailed models risk computational infeasibility, while overly abstract models may fail to capture critical construction constraints.

This reinforces the need to treat modelling and constraint formulation as decision problems, guided by construction system characteristics rather than algorithmic convenience. The next section builds on this foundation by examining AI-based optimization methods under varying modelling and constraint conditions.

5 AI-Based Optimization Methods for Construction-Oriented ASP

The combinatorial nature of Assembly Sequence Planning has led to extensive adoption of AI-based and soft computing optimization methods. However, much of the literature evaluates these methods in isolation, emphasizing algorithmic performance rather than decision-level suitability for construction systems. As established in Sections 3 and 4, the effectiveness of an optimization method in construction-oriented ASP depends strongly on assembly scale, modelling abstraction, and dominant constraint types.

Accordingly, this section synthesizes AI-based ASP methods by algorithm family, focusing on their strengths, limitations, and applicability to construction automation rather than their internal mechanics.

While quantitative benchmarking across ASP studies remains challenging due to differences in problem formulation and evaluation metrics, a comparative summary of reported scalability characteristics and typical problem sizes has been incorporated to enhance analytical depth

5.1 Evolutionary Algorithms

Evolutionary algorithms, particularly Genetic Algorithms (GA), are among the most widely applied methods for ASP [27–30]. GA-based approaches encode assembly sequences as chromosomes and evolve solutions through selection, crossover, and mutation. Their flexibility enables straightforward incorporation of multiple objectives and constraint penalties.

In construction contexts, evolutionary algorithms perform well for small- to medium-scale assemblies, particularly when part-based or task-based models are used. However, their performance degrades as assembly scale increases due to premature convergence and reduced exploration efficiency [23,24]. GA-based methods are therefore poorly suited to large-scale construction assemblies involving parallel sub-assemblies and system-level constraints.

5.2 Swarm Intelligence Methods

Swarm intelligence methods, including Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO), exploit collective search mechanisms to explore large solution spaces [25–27]. ACO is especially effective for sequencing problems, as pheromone trails provide structured guidance through discrete search spaces.

In construction-oriented ASP, swarm intelligence methods are well suited to medium-scale assemblies, where optimization-oriented constraints dominate. ACO-based approaches have demonstrated greater robustness than GA in problems with complex precedence relationships [25,26]. PSO offers computational efficiency but is less effective for highly discrete or heavily constrained ASP formulations [27].

5.3 Physics- and Bio-Inspired Optimization Methods

Physics-inspired methods, such as Simulated Annealing, have been applied to ASP due to their ability to escape local optima via probabilistic acceptance of inferior solutions [25–27]. These methods are simple to implement and require fewer parameters than population-based approaches.

In construction applications, Simulated Annealing is most effective for small-scale assemblies or as a local refinement mechanism. Its slow convergence and limited scalability restrict its applicability to large-scale assemblies with extensive parallelism and system-level constraints.

Bio-inspired methods, including Artificial Immune Systems and Shuffled Frog Leaping Algorithms, emphasize exploration and adaptability in complex search spaces [27]. While capable of handling complex constraint landscapes, these methods often incur high computational cost and require careful parameter tuning.

Hybrid and memetic algorithms combine complementary strategies, such as global search with local refinement or hierarchical decomposition [25–27]. In construction-oriented ASP, hybrid approaches show strong potential for large-scale assemblies where system-level constraints dominate. These methods align naturally with connector-based and hierarchical modelling approaches, although they require greater implementation effort and domain knowledge.

5.4 Learning-Based and Data-Driven ASP Approaches

Recent advances in machine learning have stimulated interest in learning-based ASP approaches, particularly reinforcement and imitation learning, which aim to learn sequencing policies that generalize across similar assembly configurations [28–30]. In RL-based ASP, assembly is formulated as a sequential decision-making problem, where an agent selects actions based on the current assembly state to maximize cumulative reward [28,30].

Although conceptually attractive for construction automation, current learning-based ASP research remains largely experimental. Most studies focus on simplified environments or small assemblies, with limited consideration of construction-specific challenges such as large components, parallel sub-assembly, and human–robot collaboration [29,31]. In addition, learning-based methods typically require substantial training data and careful reward design, which are difficult to obtain in construction contexts.

Related approaches, including case-based reasoning and graph-based learning, have been proposed to exploit prior solutions or structural patterns [28–30], but their applicability to large-scale construction systems remains constrained by data availability and interpretability concerns. From a decision-level perspective, learning-based methods are therefore best viewed as complementary tools, particularly suited to repetitive or data-rich contexts such as standardized modular systems or digital twin environments, rather than as replacements for established optimization methods.

Recent advancements in synthetic data generation and sim-to-real transfer techniques are increasingly addressing data scarcity and generalization challenges, suggesting that learning-based ASP approaches may become more practically viable in construction contexts than previously assumed.

5.5 Decision-Level Suitability of AI Methods for Construction ASP

The synthesis above confirms that no single AI-based optimization method is universally optimal for ASP. Algorithm suitability depends on assembly scale, modelling abstraction, and dominant constraint types. Treating algorithm selection as a decision problem is therefore essential for construction automation.

While quantitative benchmarking across ASP studies remains challenging due to differences in problem formulation and evaluation metrics, a comparative summary of reported scalability characteristics and typical problem sizes has been incorporated to enhance analytical depth.

5.6 Implications for Construction Automation

The prevalence of algorithm-centric ASP research has obscured the importance of decision-level alignment in construction automation. Algorithms that perform well on benchmark problems may be unsuitable for real construction systems if assembly scale, parallelism, and system-level constraints are ignored. Future ASP development should therefore prioritize integration, scalability, and decision transparency over incremental algorithmic novelty.

The next section synthesizes insights from Sections 3–5 to provide a comparative, construction-oriented perspective on ASP decision-making.

6 Decision-Oriented Synthesis and Research Directions

The preceding sections examined assembly scale, modelling approaches, constraint formulation, and AI-based optimization methods individually. This section synthesizes these elements to provide decision-oriented insights for construction-oriented ASP. Rather than comparing algorithms based solely on performance metrics, the synthesis emphasizes *when* and *why* specific modelling and optimization strategies are appropriate given construction system characteristics.

6.1 Alignment Between Assembly Scale, Modelling, and Optimization

A central insight from the literature is that ASP effectiveness depends on alignment between assembly scale, modelling abstraction, and optimization strategy. Mismatches between these elements are a common cause of poor scalability and limited practical applicability in construction automation.

For example, applying part-based modelling with population-based optimization to large-scale modular assemblies often leads to prohibitively large search spaces and excessive computation time [16,18]. Conversely, overly abstract models applied to small-scale assemblies may fail to capture critical geometric or stability constraints, resulting in infeasible or unsafe sequences [20,21].

Effective ASP therefore requires balancing modelling fidelity with computational tractability. Assembly scale provides a unifying lens for achieving this balance by guiding both modelling choices and algorithm selection.

6.2 Trade-offs Between Feasibility, Optimization, and System-Level Objectives

ASP objectives shift systematically with assembly scale. In small-scale assemblies, feasibility dominates, and performance differences among feasible sequences are often marginal. In medium-scale assemblies, optimization objectives, such as minimizing tool changes, direction changes, or operation-type variation, become increasingly important [6,7].

In large-scale construction assemblies, system-level objectives, including resource coordination, ergonomic considerations, and synchronization of parallel sub-assemblies, frequently outweigh local optimization goals [13,15]. ASP solutions that focus solely on local efficiency may therefore degrade overall system performance by creating bottlenecks or underutilized resources.

These observations highlight the need to frame ASP objectives in terms of overall construction system performance rather than isolated task-level efficiency.

6.3 Decision-Oriented Synthesis Framework

Integrating the insights from Sections 3–5, Table 3 presents a decision-oriented framework linking assembly characteristics, dominant constraints, modelling approaches, and suitable AI-based optimization strategies.

Table 3. Decision-oriented synthesis of ASP approaches for construction automation

Assembly Characters	Dominant Constraints	Model Method	Optimize Method
Small, sequential	Feasible	Part-based	GA, SA
Medium, mixed parallelism	Optimize-oriented	Task-based	ACO, PSO
Large, parallel, modular	System-level	Connector-based / hierarchical	Hybrid, memetic
Repetitive, data-rich	Adaptive / dynamic	Hierarchical / learned	RL-assisted, learning-based

Rather than prescribing specific algorithms, this framework supports informed decision-making by aligning ASP strategies with construction system requirements.

6.4 Implications for Practical Deployment

From a practical perspective, the synthesis explains why many ASP approaches reported in the literature have seen limited adoption in real construction automation systems. Algorithm-centric solutions that neglect assembly scale, modelling abstraction, and system-level constraints often fail to scale beyond laboratory examples.

For industrialized construction applications, such as off-site modular fabrication and robotic prefabrication facilities, ASP solutions must integrate with production planning, resource allocation, and execution control. Decision-oriented ASP frameworks are therefore more likely to support practical deployment than isolated optimization algorithms.

6.5 Research Gaps and Future Directions

Another critical gap is the limited integration of human–robot collaboration and ergonomic constraints in ASP models. Construction automation often operates in semi-automated settings where humans and robots interact closely, yet safety limits, ergonomics, and task handovers are rarely embedded explicitly in sequence planning. Incorporating these factors is essential for improving both feasibility and acceptance of automated assembly solutions.

Existing ASP formulations also commonly assume rigid components and deterministic conditions, assumptions that rarely hold in construction. Flexible or semi-deformable components, tolerance variability, and environmental uncertainty remain difficult to represent, highlighting the need for advanced modelling approaches, including hybrid geometric and data-driven representations.

While learning-based ASP methods, such as reinforcement and imitation learning, have gained interest, their practical applicability remains limited by data scarcity, training cost, and interpretability challenges. In the near term, these methods are more likely to augment traditional optimization, through solution initialization, adaptation, or digital twin integration, rather than replace established approaches.

Finally, scalability and the lack of construction-specific benchmarks remain major obstacles. Many studies rely on small or idealized examples, limiting insight into industrial-scale performance. Developing standardized datasets and benchmark problems (e.g., modular frames or panelized façade systems) would significantly improve comparability and accelerate translation to practice.

Addressing these gaps requires a shift from algorithm-centric experimentation toward integrated, system-aware, and decision-oriented ASP frameworks capable of supporting real-world automated and industrialized construction.

Future work should focus on validating the proposed decision-oriented framework through pilot case studies and real-world construction scenarios, particularly in modular and prefabrication contexts, to assess its practical applicability and scalability.

7 Conclusions

This paper presented a focused synthesis of AI-based Assembly Sequence Planning from a construction automation perspective, reframing ASP as a decision problem spanning design, production planning, and execution rather than an algorithm-centric optimization task.

A central contribution is the introduction of assembly scale as a unifying concept for understanding ASP complexity in construction. Distinguishing between small-, medium-, and large-scale assemblies clarifies how solution space growth, constraint dominance, and algorithm suitability vary systematically, explaining the limited scalability of many manufacturing-oriented ASP approaches in construction contexts involving parallel sub-assembly and system-level coordination.

The synthesis further demonstrated that ASP modelling approaches and constraint formulations are inherently coupled. Part-based, task-based, and connector-based models involve distinct trade-offs in fidelity and scalability, and no single AI-based optimization method is universally optimal; effective ASP solutions instead arise from decision-level alignment between assembly scale, modelling abstraction, and optimization strategy.

By positioning learning-based methods alongside classical meta-heuristics, the paper identified reinforcement learning and related approaches as complementary tools rather than replacements for established frameworks. Overall, this work advances ASP research toward practical applicability, scalability, and decision support, providing a structured foundation for future construction-oriented ASP development.

References

[1] Bock, T., "The future of construction automation: Technological disruption and the upcoming ubiquity of robotics," *Automation in Construction*, vol. 59, pp. 113–121, 2015.

[2] Pan, W., and Gibb, A. G. F., "Decision-making for industrialized housing systems," *Journal of Construction Engineering and Management*, vol. 138, no. 11, pp. 123–134, 2012.

[3] De Fazio, T. L., and Whitney, D. E., "Simplified generation of all mechanical assembly sequences," *IEEE Journal of Robotics and Automation*, vol. 3, no. 6, pp. 640–658, 1987.

[4] Baldwin, D. F., Abell, T. E., Lui, M. C., De Fazio, T. L., and Whitney, D. E., "An integrated computer aid for generating and evaluating assembly sequences," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 1, pp. 78–94, 1991.

[5] Yuan, Z., Sun, C., and Wang, Y., "Design for assembly in prefabricated construction," *Automation in Construction*, vol. 88, pp. 13–22, 2018.

[6] Lu, Y., Wang, X., and Chen, K., "Multi-objective assembly sequence planning using genetic algorithms," *International Journal of Advanced Manufacturing Technology*, vol. 28, no. 7–8, pp. 721–730, 2006.

[7] Pan, Z., Polden, J., Larkin, N., Van Duin, S., and Norrish, J., "Recent progress on programming methods for industrial robots," *International Journal of Production Research*, vol. 50, no. 13, pp. 3603–3619, 2012.

[8] Homem de Mello, L. S., and Sanderson, A. C., "AND/OR graph representation of assembly plans," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 2, pp. 188–199, 1990.

[9] Wolter, J. D., Chakrabarty, S., and Tsao, T. C., "Automated assembly planning: A survey," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 3, pp. 453–464, 1994.

[10] Chen, L., and Liu, Y., "Assembly sequence planning based on CAD models," *Computer-Aided Design*, vol. 33, no. 6, pp. 403–417, 2001.

[11] Lu, N., "Design for manufacture and assembly for prefabricated housing," *Automation in Construction*, vol. 18, no. 2, pp. 211–219, 2009.

[12] Rashid, M. M., Fattah, A., and Ahmed, S., "Assembly sequence planning using ant colony optimization," *International Journal of Production Research*, vol. 49, no. 18, pp. 5377–5406, 2011.

[13] Villani, V., Pini, F., Leali, F., and Secchi, C., "Survey on human–robot collaboration in industrial settings," *Mechatronics*, vol. 55, pp. 248–266, 2018.

[14] Yu, J., and Wang, X., "Assembly sequence planning using particle swarm optimization," *International Journal of Advanced Manufacturing Technology*, vol. 64, no. 9–12, pp. 1437–1450, 2013.

[15] Mok, H. S., Kim, Y. S., and Lee, J. H., "Constraint-based assembly sequence planning," *International Journal of Production Research*, vol. 39, no. 17, pp. 4007–4023, 2001.

[16] Tseng, M. M., and Su, C. J., "Assembly planning using part-based modelling," *International Journal of Production Research*, vol. 45, no. 21, pp. 4827–4849, 2007.

[17] Shan, H., Wang, Y., and Liu, J., "Memetic algorithms for assembly sequence planning," *Applied Soft Computing*, vol. 9, no. 2, pp. 651–660, 2009.

[18] Lu, Y., and Wang, X., "Hierarchical assembly planning for modular products," *International Journal of Advanced Manufacturing Technology*, vol. 83, no. 5–8, pp. 1001–1016, 2016.

[19] Sacks, R., Girolami, M., and Brilakis, I., "Building information modelling, digital twins and smart construction," *Data-Centric Engineering*, vol. 1, e14, 2020.

[20] Dul, J., and Neumann, W. P., "Ergonomics contributions to company strategies," *Applied Ergonomics*, vol. 40, no. 4, pp. 745–752, 2009.

[21] Chen, Y., Xu, Y., and Wang, Z., "Reinforcement learning for robotic assembly planning," *Robotics and Computer-Integrated Manufacturing*, vol. 68, 2021.

[22] Mnih, V., Kavukcuoglu, K., Silver, D., et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[23] Goldberg, D. E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

[24] Holland, J. H., Adaptation in Natural and Artificial Systems, MIT Press, 1992.

[25] Dorigo, M., and Stützle, T., *Ant Colony Optimization*, MIT Press, 2004.

[26] Dorigo, M., Maniezzo, V., and Colorni, A., "Ant system: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.

[27] Kennedy, J., and Eberhart, R., "Particle swarm optimization," *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.

[28] Sutton, R. S., and Barto, A. G., *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, 2018.

[29] Silver, D., Huang, A., Maddison, C. J., et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, 2016.

[30] Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A., "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[31] Zhang, J., Teizer, J., and Pradhananga, N., "Digital twin–enabled construction automation," *Automation in Construction*, vol. 136, 2022.