Knowledge-Driven Workflow Optimization for Integrating Industrial Robotics into Conventional Construction Management

Brigid Hegarty
Department of Computer Science, University of New Hampshire, Durham, NH, USA.
hegartybrigid@unh.edu

Ferdinando Lollobrigida
Department of Computer Science, Binghamton University, Binghamton, NY, USA.
helloferdinando@binghamton.edu

Abstract

Industrial robots are increasingly capable of executing construction tasks that were previously organized around human crews, yet conventional construction management systems represent work at a granularity that is too coarse for robotic execution. Work breakdown structures, critical path schedules, and Last Planner routines coordinate trades and buffers but do not encode robot reach, payload, tolerance, calibration, safety-zone, and handoff requirements. This paper develops a knowledge-driven workflow optimization framework that links construction-domain knowledge to executable robotic work packages and to robust scheduling decisions. The framework has three components. First, an ontology and knowledge graph represent tasks, work packages, robot capabilities, site states, constraints, and evidence, allowing capability matching and constraint generation to be traced to explicit knowledge rather than ad hoc rules. Second, a capability-aware eligibility model filters task-robot assignments, and a precedence model represents setup, calibration, and safety handoffs as first-class workflow dependencies. Third, a mixed-integer optimization model balances makespan, cost, risk, and schedule deviation under bounded duration uncertainty, with execution feedback updating capability estimates. The contribution is a formal integration logic for embedding industrial robotics into conventional construction management without replacing existing planning artifacts. The framework clarifies how robotic constraints alter workflow precedence, why semantic knowledge representation is necessary for reliable constraint generation, and how robust optimization can support managerial trade-offs. Its main implications concern task decomposition, handoff design, knowledge maintenance, and the boundary conditions under which robotic construction schedules remain both feasible and auditable.

Keywords

construction robotics; workflow optimization; knowledge representation; robust scheduling; construction management; human-robot handoff

1. Introduction

Industrial robotics in construction has advanced from isolated single-task machines toward configurable systems for bricklaying, drilling, welding, surface treatment, material handling, and prefabrication support [1], [2], [3]. This technical progress has not been matched by an equivalent transformation in construction management practice. Conventional

management represents work through work breakdown structures, activity-on-node networks, trade coordination, and production-control meetings. Building information modeling improves geometric and semantic coordination [4], [5], while lean construction and Last Planner practices improve commitment reliability and flow [6], [7]. Project management standards provide control structures for scope, schedule, cost, and risk [8]. These instruments are effective for human-centered operations, but they treat execution capability as an implicit property of crews, subcontractors, and site experience.

Robotic execution exposes that implicitness. A robotic work package must satisfy reach, payload, stability, tolerance, calibration, material-presentation, safety-separation, and commissioning conditions that are not normally represented in a construction schedule. Scheduling theory provides mature models for precedence, resource capacity, and multi-mode execution [9], [10], but those models assume that the feasible modes and their durations are already known. In robotic construction, the knowledge that determines feasible modes is distributed across robot specifications, site geometry, temporary works, safety plans, operator experience, and commissioning records. The result is a workflow-design gap: managers can schedule a robotic task as an activity, but they cannot reliably determine whether the activity is executable, which handoffs it requires, how sensitive it is to site variation, or how robotic constraints should alter the sequence of conventional work.

This paper addresses that gap by proposing a knowledge-driven workflow optimization framework for integrating industrial robotics into conventional construction management. The central contribution is a formal decision architecture that connects explicit construction knowledge to capability-aware task eligibility, precedence and handoff modeling, and robust multi-objective scheduling. The framework is methodological and analytical. It does not assume that a particular robot, site, or project dataset is available; instead, it specifies how domain knowledge should be represented, how robotic constraints should enter scheduling formulations, and how execution feedback should revise the knowledge base. The argument is that integration fails neither because optimization is absent nor because robots are insufficiently capable, but because the knowledge that determines feasible robotic work is not represented in a form that management systems can use.

2. Conventional Workflows and Robotic Integration Constraints

2.1 Workflow representation in construction management

Conventional construction workflows are organized around deliverable-oriented decomposition. A work package groups activities that can be planned, procured, executed, inspected, and handed over under managerial responsibility. Critical path methods then impose precedence and duration assumptions, while resource-leveling procedures resolve shared equipment and labor constraints. Lean production control adds short-interval planning and commitment management, which reduces schedule uncertainty by making readiness and constraints explicit at the crew level [6], [7]. BIM-based coordination adds spatial and product information, and digital construction platforms increasingly connect model objects to schedule activities [4], [5].

These representations are not designed to express robotic execution knowledge. A schedule activity such as “install curtain wall panel level three” may be perfectly adequate for a crew-based operation, but it does not specify whether a mobile manipulator can reach the fixing points, whether the panel must be presented in a particular orientation, whether the robot base requires a stabilized floor zone, or whether a human spotter must control an

exclusion area. When such conditions remain tacit, the schedule can be logically valid but operationally infeasible. The problem is especially acute at interfaces between prefabrication, logistics, temporary works, and on-site assembly, where robotic execution depends on upstream decisions that conventional schedules treat as separate trade responsibilities.

Scheduling formulations can represent alternative execution modes and resource constraints [9], [10], but they require a prior enumeration of modes and durations. In conventional practice, that enumeration is supplied by cost libraries, historical productivity rates, or expert judgment. For robotic work, the enumeration depends on a matching relation between task requirements and robot capabilities. Without an explicit knowledge layer, the optimizer may select an assignment that violates a capability threshold or may exclude a feasible assignment because the relevant knowledge was never entered into the model.

2.2 Robotic execution knowledge and integration failures

Robotic construction introduces three classes of constraint that are poorly handled by conventional workflow models. The first is capability constraint. Reach, payload, dexterity, positioning accuracy, sensor performance, and tool compatibility determine whether a robot can execute a task at a given location and tolerance. The second is readiness constraint. Robotic execution often requires stabilized bases, calibrated reference frames, clean power, material staging, and verified safety zones. These readiness conditions create precedence dependencies that are not product-based but execution-based. The third is uncertainty constraint. Site conditions, weather, material fit-up, and human-robot coordination can cause duration and quality deviations that are not well represented by deterministic activity durations.

Robust optimization provides a disciplined way to represent bounded uncertainty without collapsing the schedule to a single deterministic scenario [11], [12]. Multi-criteria optimization provides a way to expose trade-offs among makespan, cost, risk, and schedule stability rather than hiding them in a single composite score [13]. These methods are necessary but not sufficient. They operate on a constraint set whose structure must reflect construction and robotic knowledge. If the knowledge layer is weak, the optimization model becomes numerically sophisticated but semantically unreliable.

3. Knowledge-Driven Workflow Model

3.1 Ontological representation and knowledge base

The framework begins with an ontology that defines the entities and relations needed to reason about robotic construction workflows. Core entities include task, work package, robot, tool, site state, constraint, evidence, and handoff. Core relations include requires capability, precedes, incompatible with, validated by, requires readiness, and assigned to. The ontology does not attempt to encode all construction knowledge. It defines the minimal vocabulary needed to translate conventional work packages into robotic execution conditions and to trace each condition to a source, such as a robot specification, a method statement, a commissioning record, a safety assessment, or a past execution report.

Ontology engineering principles support this choice [14], [15]. A formal vocabulary allows the same constraint to be reused across projects and checked for consistency. It also allows the optimization model to be generated from queries rather than hard-coded rules. For example, a query can retrieve all robots whose payload and reach envelopes satisfy a task requirement, then retrieve the readiness conditions that must precede their deployment. The

knowledge base therefore acts as a boundary object between construction management, robotics engineering, safety planning, and site operations. It also provides an audit trail: if a task is excluded from robotic assignment, the exclusion can be traced to a capability threshold, a risk limit, or a missing readiness condition rather than to an unexplained planning judgment.

3.2 Capability matching and eligibility

Capability matching compares task requirements with robot capabilities across K relevant dimensions. Let q_{tk} be the requirement of task t on capability dimension k, and let c_{rk} be the corresponding capability of robot r. The weights w_{tk} express the relative importance of each dimension for the task, with nonnegative weights summing to one. The capability score is defined as follows.

$$\kappa_{tr} = \sum_{k} w_{tk} \min\left(1, \frac{c_{rk}}{q_{tk}}\right)$$

The ratio $\min(1, c_{rk}/q_{tk})$ rewards capability that meets or exceeds the requirement and penalizes shortfall proportionally. The score is not a productivity prediction. It is a transparent matching index that supports eligibility screening and constraint generation. A task with a high requirement for positioning accuracy, for example, will receive a low score from a robot whose accuracy is below that requirement even if its payload is large. The weights allow project-specific priorities to be represented explicitly, but they should be documented in the knowledge base so that later schedule outcomes can be interpreted.

Eligibility is then defined by threshold and risk conditions. Let θ_t be the minimum acceptable capability score for task t, let ρ_{tr} be the assessed execution risk of assigning robot r to task t, and let ρ_t^{max} be the maximum acceptable risk. The eligible assignment set is given by the following expression.

$$A_t = \{r \in R : \kappa_{tr} \geq \theta_t, \rho_{tr} \leq \rho_t{}^{max}\}$$

The risk term may combine safety-zone conflict, environmental exposure, tolerance sensitivity, and human-robot proximity. The threshold θ_t and the risk limit ρ_t^{max} are managerial parameters, but they are not arbitrary. They should be calibrated against method statements and execution evidence and stored with version control. The eligibility set reduces the combinatorial space of the optimization model and prevents assignments that are semantically infeasible even if they appear attractive in cost or duration terms.

3.3 Precedence, handoffs, and execution sequences

Robotic work packages introduce handoffs that are not captured by product-level precedence alone. A handoff may include logistical preparation, calibration, safety-zone establishment, material presentation, and post-execution inspection. Let S_t be the start time of task t, d_{ur} the duration of task u when assigned to robot r, x_{ur} a binary assignment variable, and h_{uv} the handoff time between tasks u and v. For every precedence relation (u, v) in the workflow graph P, the schedule must satisfy the following condition.

$$S_v \geq S_u + \sum_{r} d_{ur} x_{ur} + h_{uv}, (u, v) \in P$$

The handoff term h_{uv} is not a generic buffer. It is derived from knowledge-base relations such as calibration dependency, safety-zone transition, material presentation, and inspection release. This distinction matters because buffers can hide unreadiness, whereas explicit handoff constraints expose the operational work required to make a robotic task executable. If

a calibration step is missing, the optimizer should not be allowed to absorb the omission as a probabilistic delay. The missing readiness condition should appear as a constraint or as a knowledge-base defect.

4. Optimization Formulation

4.1 Objective and constraints

The workflow optimization model assigns eligible tasks to robots and sequences them subject to precedence, resource, and handoff constraints. The objective balances four managerial concerns: makespan, direct execution cost, execution risk, and deviation from the baseline construction schedule. Let T_{max} be the project makespan, c_{tr} the cost of assigning robot r to task t, ρ_{tr} the corresponding risk, S_t^0 the planned start time from the conventional baseline schedule, and α, β, γ, δ nonnegative scalar weights. A representative objective is the following.

$$min\, Z = \alpha T_{max} + \beta \sum c_{tr}\, x_{tr} + \gamma \sum \rho_{tr}\, x_{tr} + \delta \sum |S_t - S_t{}^0|$$

The first term compresses project duration. The second and third terms penalize costly and risky assignments. The fourth term discourages unnecessary displacement of conventional work, which matters because robotic tasks are embedded in a larger schedule with subcontractors, inspections, and material deliveries. The weights should reflect project priorities and can be varied in a Pareto analysis rather than fixed once for all projects. The model must also enforce single assignment for each task, start-time nonnegativity, precedence constraints, and resource capacity constraints for robots, operators, lifting equipment, and shared work areas.

The objective is deliberately multi-criteria rather than a single productivity measure. A schedule that minimizes robot time may create costly temporary works or destabilize subsequent trades. A schedule that minimizes risk may be unnecessarily long and expensive. The optimization does not resolve these trade-offs normatively; it makes them explicit and auditable.

4.2 Uncertainty and robust feasibility

Robotic construction schedules are exposed to bounded uncertainty in task duration, calibration time, material fit-up, and human-robot coordination. A deterministic model can produce a schedule that is optimal for one assumed duration vector but fragile under small deviations. The framework therefore represents uncertain duration as a nominal value plus a bounded deviation. Let d_{tr}^0 be the nominal duration of task t assigned to robot r, $\hat{d}_{tr}$ the maximum positive deviation, ξ_{tr} a deviation variable, and Γ an uncertainty budget controlling how many deviations may simultaneously reach their maximum. The uncertain duration is expressed as follows.

$$d_{tr}(\xi) = d_{tr}{}^0 + \xi_{tr}\, \hat{d}_{tr}, 0 \le \xi_{tr} \le 1, \sum \xi_{tr} \le \Gamma$$

The budget Γ is a managerial parameter that controls conservatism. A small Γ protects against a few simultaneous delays; a large Γ produces a more robust but potentially more expensive schedule. The robust counterpart of the precedence and resource constraints can be generated using standard reformulation techniques [11], [12]. The important point for construction management is that uncertainty is not treated as a generic contingency percentage. It is attached to specific tasks and robot assignments and bounded by an explicit robustness budget.

4.3 Why optimization rather than rule execution

A rule-based system could encode simple conditions such as “use robot type A when reach exceeds ten meters.” Such rules are transparent but brittle because they cannot trade off cost, risk, and schedule disruption across a network of interdependent tasks. Simulation alone can evaluate scenarios but does not search systematically for a best assignment and sequence. A pure optimization model without a knowledge layer can compute an optimum over an incorrectly specified feasible set. The proposed coupling of ontology-based knowledge, eligibility filtering, handoff constraints, and robust multi-objective optimization addresses each weakness. The knowledge layer defines what is feasible; the optimizer determines what is preferable within that feasible set.

5. Solution Procedure and Implementation Logic

The solution procedure begins with conventional construction artifacts: work breakdown structure, baseline schedule, BIM objects, method statements, safety plan, and resource plan. These artifacts are mapped into work packages and tasks. Each task is annotated with capability requirements, readiness conditions, and potential handoff dependencies. The knowledge base then queries robot specifications and site-state information to generate eligibility sets and handoff constraints.

The next step is constraint generation. Capability thresholds produce assignment restrictions. Readiness conditions produce additional precedence arcs. Safety-zone requirements produce mutual-exclusion constraints for shared work areas. Resource plans produce capacity constraints for robots, operators, and lifting equipment. These constraints are passed to a mixed-integer solver. The solver searches over eligible assignments and start times, and the robust formulation ensures that the schedule remains feasible under the specified uncertainty budget. The result is not merely a schedule but an assignment rationale: each selected robot-task pair is linked to a capability score, each excluded pair is linked to a violated condition, and each handoff is linked to a readiness requirement.

Execution feedback closes the loop. After each work package, observed duration, quality deviation, and handoff performance are recorded. Capability estimates are updated rather than replaced. Let $c_{rk}^{(n)}$ be the current estimate of robot capability dimension k after n execution cycles, let $\tilde{c}_{rk}^{(n)}$ be the observed value from the latest cycle, and let η be a learning rate between zero and one. The update rule is as follows.

$$c_{rk}^{(n+1)} = (1 - \eta) c_{rk}^{(n)} + \eta\, \tilde{c}_{rk}^{(n)}, 0 < \eta \leq 1$$

The update rule is a knowledge-maintenance mechanism, not an autonomous change to safety limits. Thresholds and risk limits should remain under formal governance. The rule simply prevents the knowledge base from freezing at commissioning values when site conditions and operational learning change.

The factory-to-site perspective is important at this point. Design decisions that determine access, temporary works, tolerance strategy, and material presentation can make a robotic task easy or impossible before scheduling begins [16]. The workflow optimizer cannot compensate for a product or site design that excludes robotic execution. It can, however, make the consequence of such design choices visible by showing which tasks become ineligible, which handoffs expand, and which schedule alternatives disappear. Knowledge engineering principles support this integration by separating domain vocabulary, inference rules, and

execution data [17]. Linked-data and knowledge-graph architectures further support the querying and reuse of heterogeneous construction information across project parties [18], [19].

6. Technical Discussion and Boundary Conditions

The framework's main strength is semantic traceability. Conventional schedule optimization often produces results whose feasibility depends on assumptions that are not visible to reviewers. In the proposed model, a robotic assignment is feasible because a capability score exceeds a threshold, a risk limit is satisfied, and a set of readiness conditions is represented in the precedence graph. This traceability matters for safety review, contractual coordination, and post-project learning. It also supports incremental adoption: a contractor can begin with a small knowledge base for one robot class and one work package type, then expand coverage as evidence accumulates.

The approach also has boundary conditions. The ontology must be maintained. If robot capabilities, tool configurations, or site-readiness rules change and the knowledge base is not updated, the optimizer will produce confident but stale recommendations. The capability score is a screening device, not a substitute for engineering verification. Thresholds and risk limits require calibration against method statements, safety cases, and execution records. The uncertainty budget must be chosen with awareness of its effect on cost and schedule. A high robustness budget can lead to over-conservative schedules that reserve excessive time and space for robotic tasks, while a low budget can produce schedules that are theoretically optimal but operationally fragile.

A second boundary condition concerns handoff design. Robotic construction is not simply a matter of replacing a human activity with a machine activity. It often redistributes work to calibration, material preparation, inspection, and safety control. If those handoffs are underestimated, the optimizer may appear to improve productivity while shifting workload to unmodeled activities. The explicit handoff term in the precedence model is intended to prevent this form of accounting drift. The lean and BIM integration literature similarly emphasizes that digital information improves production only when it is connected to actual flow and commitment management [20]. Robot-oriented design extends this point by treating automation requirements as design and management inputs rather than downstream adjustments [21]. Virtual design and construction provides a related precedent for integrating product, organization, and process information before execution [22].

A third boundary condition concerns human-robot shared work. The model represents safety zones and eligibility limits, but it does not simulate human behavior or certify safe operation. Those functions remain with safety engineering, site supervision, and regulatory approval. The optimizer's role is to avoid scheduling assignments that violate explicitly represented constraints and to expose trade-offs that managers would otherwise resolve informally. In this sense, the framework is a decision-support architecture rather than an autonomous planning system.

7. Conclusion

This paper has developed a knowledge-driven workflow optimization framework for integrating industrial robotics into conventional construction management. The starting problem is not a lack of scheduling algorithms or robot capability, but a mismatch between the knowledge required for robotic execution and the granularity at which construction workflows are represented. The framework responds with an ontology-based knowledge layer, capability-aware eligibility sets, explicit handoff and readiness constraints, a robust multi-

objective optimization model, and an execution-feedback mechanism for knowledge maintenance.

The main analytical insight is that robotic constraints should enter construction management as structured knowledge before they enter as schedule constraints. When capability, risk, readiness, and handoff conditions are represented explicitly, the optimizer can search a semantically valid feasible set and managers can audit why an assignment was selected or rejected. When those conditions remain tacit, optimization may improve a schedule that should not have been considered feasible. The framework therefore contributes a formal integration logic rather than a claim of field performance. Its practical value lies in making robotic construction workflows traceable, comparable, and adaptable within existing construction management structures, while preserving engineering verification and safety governance as non-negotiable boundary conditions.

References

1. Bock, T. (2007). Construction robotics. Autonomous Robots, 22(3), 201–209.
2. Bock, T., & Linner, T. (2016). Construction robots: Elementary technologies and single-task construction robots. Cambridge University Press.
3. Saidi, K. S., O'Brien, J. B., & Lytle, A. M. (2008). Robotics in construction. In B. Siciliano & O. Khatib (Eds.), Springer handbook of robotics (pp. 1079–1098). Springer.
4. Sacks, R., Eastman, C., Lee, G., & Teicholz, P. (2018). BIM handbook: A guide to building information modeling for owners, designers, engineers, contractors, and facility managers (3rd ed.). Wiley.
5. Eastman, C., Teicholz, P., Sacks, R., & Liston, K. (2011). BIM handbook: A guide to building information modeling for owners, managers, designers, engineers, and contractors (2nd ed.). Wiley.
6. Ballard, G., & Howell, G. (2003). Lean project management. Building Research & Information, 31(2), 119–133.
7. Koskela, L. (1992). Application of the new production philosophy to construction (CIFE Technical Report No. 72). Stanford University.
8. Project Management Institute. (2017). A guide to the project management body of knowledge (PMBOK guide) (6th ed.). Project Management Institute.
9. Pinedo, M. L. (2016). Scheduling: Theory, algorithms, and systems (5th ed.). Springer.
10. Brucker, P., & Knust, S. (2012). Complex scheduling. Springer.
11. Bertsimas, D., & Sim, M. (2004). The price of robustness. Operations Research, 52(1), 35–53.
12. Ben-Tal, A., El Ghaoui, L., & Nemirovski, A. (2009). Robust optimization. Princeton University Press.
13. Ehrgott, M. (2005). Multicriteria optimization (2nd ed.). Springer.
14. Noy, N. F., & McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology (Technical Report KSL-01-05). Stanford Knowledge Systems Laboratory.

15. Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2), 199–220.

16. Luo, D., & Yu, L. (2021). From factory to site—designing for industrial robots used in on-site construction. In Automating cities: design, construction, operation and future impact (pp. 87-109). Singapore: Springer Singapore.

17. Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: Principles and methods. Data & Knowledge Engineering, 25(1–2), 161–197.

18. Pan, J. Z., Vetere, G., Gomez-Perez, J. M., & Wu, H. (Eds.). (2017). Exploiting linked data and knowledge graphs in large organisations. Springer.

19. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutierrez, C., Kirrane, S., Labra Gayo, J. E., Navigli, R., Neumaier, S., Ngomo, A.-C. N., Polleres, A., Rashid, S. M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., & Zimmermann, A. (2021). Knowledge graphs. ACM Computing Surveys, 54(4), 1–37.

20. Sacks, R., Koskela, L., Dave, B. A., & Owen, R. (2010). Interaction of Lean and building information modeling in construction. Journal of Construction Engineering and Management, 136(9), 968–980.

21. Bock, T., & Linner, T. (2015). Robot-oriented design: Design and management tools for the deployment of automation and robotics in construction. Cambridge University Press.

22. Fischer, M., & Kunz, J. (2013). Virtual design and construction: Themes, case studies and implementation suggestions (CIFE Working Paper No. 097). Stanford University.