# Intelligent Scheduling and Resource Allocation for Automated Robotic Construction Using Additive Manufacturing

Ignazio Brunello
School of Information Technology, University of Cincinnati, Cincinnati, OH, USA.
ignaziobrunello592@uc.edu

Maurycy Subocz
School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA.
msubocz@oregonstate.edu

Elvy Rosenqvist
Department of Computer Science and Engineering, University of Nevada, Reno, Reno, NV, USA.
helloelvy@unr.edu

**Abstract**

Large-scale robotic additive manufacturing in construction couples geometric decomposition, mobile manipulator reach, material delivery, curing, and structural stability in a single operational problem. Existing scheduling models for additive manufacturing are largely designed for fixed machines and prefabricated parts, whereas construction robots operate in evolving spatial environments where task precedence, material rheology, and robot availability change during execution. This manuscript develops an integrated scheduling and resource-allocation framework for automated robotic construction using additive manufacturing. The contribution is a digital-twin-informed formulation that links a spatial task graph to a mixed-integer scheduling model, a learning-assisted predictor of task duration and quality risk, and a robust rolling-horizon reoptimization procedure. The model assigns deposition, placement, inspection, and replenishment tasks to compatible robots; enforces precedence, material inventory, energy, and non-overlap constraints; and represents duration uncertainty through task-specific robustness budgets. The formulation clarifies why spatial decomposition, material starvation, and robot reachability cannot be treated as independent post-processing checks: each alters the feasible schedule and the resource allocation. The analytical assessment shows that conservative robustness improves schedule feasibility at the cost of makespan and that asymmetric prediction loss is appropriate when late completion is more costly than early completion. The framework provides a technically grounded basis for planners that coordinate mobile robots, extrusion rates, material buffers, and inspection pauses in robotic construction.

**Keywords**

robotic construction; additive manufacturing; project scheduling; resource allocation; robust optimization; digital twin

## 1. Introduction

Automated construction with robotic additive manufacturing has advanced from laboratory demonstrations to integrated processes that extrude cementitious materials, place reinforcement, inspect deposited layers, and coordinate mobile platforms around a growing structure [1], [2]. The appeal is not merely geometric freedom. Robotic deposition can reduce formwork, integrate digital design information directly into fabrication, and permit variable material placement along structural load paths [3]. These capabilities, however, create an operational problem that is more tightly coupled than conventional prefabrication: the object being manufactured is also the workspace, and every deposited segment changes reachability, collision geometry, support conditions, and the time available for curing and inspection.

Material behavior intensifies this coupling. Extrusion-based construction depends on yield stress, thixotropy, open time, and layer interval; a schedule that is feasible from a robot-motion perspective may still produce cold joints, excessive deformation, or unstable overhangs [4]. Multi-robot and mobile-platform studies show that throughput gains depend on task allocation, base placement, and material supply rather than on deposition speed alone [5]. Consequently, scheduling for robotic construction cannot be reduced to sequencing a fixed set of independent printing jobs. It must represent spatial decomposition, technological precedence, robot-task compatibility, inventory dynamics, and uncertainty in a common decision model.

Classical resource-constrained project scheduling and machine scheduling provide useful foundations for precedence, renewable resources, and tardiness objectives [6], [7], [8]. Those formulations assume that task durations and resource requirements are known or can be summarized by deterministic parameters. Construction-scale additive manufacturing violates this assumption in specific ways: deposition duration depends on path geometry and robot configuration, material supply is constrained by pump and buffer capacity, and inspection or curing pauses are process requirements rather than optional delays. The resulting problem is a resource-constrained project schedule with spatial compatibility, material-flow constraints, and endogenous task durations.

This manuscript addresses that gap by formulating an intelligent scheduling and resource-allocation framework for automated robotic construction using additive manufacturing. The central contribution is not a single algorithm but a coherent decision architecture: a spatial task graph derived from building information and wireframe decomposition; a mixed-integer scheduling model that simultaneously assigns tasks to robots and allocates material, energy, and time; a learning-assisted prediction component that estimates duration and quality risk from process features; and a robust rolling-horizon procedure that repairs the schedule as the digital twin updates. The framework is developed analytically, with explicit assumptions, constraints, and trade-offs. It does not report a field deployment or benchmark victory; instead, it specifies how the decision problem should be represented and why the chosen representation is appropriate for construction-scale robotic additive manufacturing.

## 2. System Architecture and Problem Definition

The planning problem is defined over a finite construction horizon and a set of tasks generated from a building information model and a process plan. Tasks represent deposition segments, material refills, reinforcement placement, inspection, curing waits, and mobile-base relocations. The task graph is a directed acyclic graph in which an edge from task i to task j encodes a technological or stability precedence: j cannot begin until i has reached a specified

completion state. Precedence may arise from geometric dependency, support removal, minimum layer interval, or inspection release. This graph is not static in practice because spatial decomposition can be revised when a robot cannot reach a segment or when a wireframe topology requires additional support.

Robust optimization provides the language for duration uncertainty without assuming a known probability distribution [9], [10]. In the present setting, uncertainty is not merely random machine delay. It includes rheological variation, pump pressure fluctuations, mobile-base positioning error, and inspection outcomes that require rework. Robust scheduling is therefore used to ensure that a planned sequence remains feasible when task durations deviate within bounded intervals. The price of robustness is an explicit trade-off: larger uncertainty budgets protect against delay but reserve robot time and material buffers that may reduce productive output.

Learning enters as a predictive layer rather than as a replacement for constraints. Reinforcement-learning and function-approximation methods can adapt policies in sequential decision problems [11], [12], but construction scheduling has hard technological constraints and sparse failure data. The framework therefore uses supervised prediction for task duration and quality risk, while retaining a constrained optimization model for feasibility. The predictor supplies parameters and risk penalties; the optimizer enforces compatibility, precedence, inventory, and energy limits.

A digital twin supplies the state used by the scheduler [13]. At each decision epoch, the twin records completed tasks, robot positions, battery state, buffer inventory, curing age, and detected deviations. Building information modeling provides the geometric and semantic basis for task generation and resource requirements [14]. The twin and the building model are not decorative data layers: they determine which tasks are currently feasible, which robot-task assignments are physically admissible, and which material deliveries can arrive before starvation.

## 3. Scheduling and Resource-Allocation Formulation

Spatial wireframe printing makes the coupling between geometry and scheduling explicit. Mesh optimization changes path length, support requirements, and the number of deposition segments, which in turn changes task durations and precedence relations [15]. A mesh that is efficient from a structural viewpoint may create many short tasks that increase robot repositioning, whereas a coarser decomposition may reduce travel but demand longer uninterrupted material supply. The task graph must therefore be generated jointly with the scheduling model rather than fixed before resource allocation.

Let I be the set of tasks, R the set of robots, M the set of materials, and T the discretized planning horizon. Binary variable $x_{ir}$ equals one if task i is assigned to robot r. The start and completion times of task i are $s_i$ and $C_i$. The nominal processing time of task i on robot r is $p_{ir}$, the energy consumption is $e_{ir}$, and the material quantity of type m required by task i is $a_{im}$. The objective balances weighted completion time, robot energy use, and material-shortage penalties:

$$\min J = \sum_i w_i C_i + \alpha \sum_r \sum_i e_{ir} x_{ir} + \beta \sum_m \sum_t \max\big(0, D_m(t) - I_m(t)\big)$$

Here $w_i$ is the scheduling weight of task i, α and β are nonnegative cost coefficients, $D_m(t)$ is the desired material inventory of type m at time t, and $I_m(t)$ is the actual inventory. The first

term prioritizes early completion of structurally or operationally critical tasks; the second penalizes energy-intensive robot assignments; the third penalizes material shortage. In an implementation, the max term is linearized with auxiliary shortage variables.

Assignment is restricted by physical and process compatibility. Let $a_{ir}$ equal one when robot r can reach the task region, carry the required tool, satisfy payload and precision limits, and operate under the current collision geometry; otherwise $a_{ir}$ equals zero. Each task must be assigned exactly once, and assignment is only permitted to compatible robots:

$$\sum_r x_{ir} = 1, x_{ir} \leq a_{ir}, \forall i$$

This constraint is deliberately stronger than a distance-based eligibility test. Reachability in a growing structure is state-dependent, and a robot that can reach a point at the start of the build may be blocked by the structure later. The digital twin updates $a_{ir}$ as deposition proceeds.

Precedence and timing are encoded by start and completion variables. For every precedence pair (i,j) in the task graph P, task j cannot start before task i completes, and task i cannot complete before its assigned processing time has elapsed:

$$C_i \geq s_i + \sum_r p_{ir} x_{ir}, s_j \geq C_i, \forall (i,j) \in P$$

Robot non-overlap is enforced either by time-indexed occupancy variables or by disjunctive interval constraints. The former is convenient for rolling-horizon optimization because it aligns with the digital twin's discrete state updates; the latter reduces the number of binary variables when tasks are long and few. In both cases, a robot cannot execute two tasks whose active intervals intersect, and mobile-base relocation time is treated as a task when it consumes a robot resource.

Material flow is a separate constraint because deposition cannot proceed without a continuous supply of printable material. Let $q_m(t)$ be the replenishment quantity of material m arriving at time t, and let $u_i(t)$ equal one when task i consumes material m during period t. Inventory balance is:

$$I_m(t+1) = I_m(t) + q_m(t) - \sum_i a_{im} u_i(t), \forall m, t$$

This relationship makes material starvation an endogenous scheduling outcome rather than an exogenous delay. If a high-priority deposition task is scheduled before a replenishment, the model may select a different robot assignment or insert an inspection task that consumes no material. The replenishment decisions $q_m(t)$ are constrained by pump capacity, delivery lead time, and buffer volume.

Energy and local resource limits are represented for each robot over the planning horizon. Let $E_r$ be the usable energy budget of robot r within the horizon, accounting for battery capacity, recharge opportunities, and safety reserve:

$$\sum_i e_{ir} x_{ir} \leq E_r, \forall r$$

The constraint is intentionally a horizon-level budget rather than a full power-flow model. At the scheduling scale, detailed motor currents and hydraulic dynamics are not decision variables; they are embedded in $e_{ir}$ through calibration. This separation keeps the optimization tractable while preserving the operational fact that energy availability can force task reassignment.

## 4. Learning-Assisted Prediction and Robust Reoptimization

The scheduling model requires $p_{ir}$, $e_{ir}$, and risk estimates for inspection and rework. These quantities are not fixed constants across projects. A predictor $f\theta$ maps task and process features $\varphi_i$ to an estimated duration or quality-risk score. Features include path length, layer index, extrusion rate, material batch, nozzle geometry, robot configuration, ambient temperature, and the time since the previous layer. The predictor is trained on historical process logs, but the scheduling model treats its output as a parameter estimate with explicit uncertainty rather than as ground truth.

Because late completion is usually more damaging than early completion in construction—late tasks can delay subsequent layers, curing windows, and material deliveries—the prediction loss should be asymmetric. A regularized loss can penalize underestimation more strongly than overestimation:

$$L(\theta) = \sum_i \ell\big(p_i, f\theta(\varphi_i)\big) + \lambda \sum_i \max\big(0, p_i - f\theta(\varphi_i)\big) + \gamma\Omega(\theta)$$

Here $\ell$ is a differentiable base loss such as Huber loss, $\lambda$ controls the asymmetric penalty for underprediction, $\gamma$ is a regularization weight, and $\Omega(\theta)$ penalizes model complexity. The asymmetric term is justified by the scheduling consequence: an underestimated duration creates optimistic precedence links and insufficient material buffers, whereas an overestimate mainly reserves capacity that may remain unused.

Robustness is introduced through interval uncertainty. Let $\bar{p}_i$ be the nominal processing time and $\hat{p}_i$ the maximum admissible positive deviation for task i. The robustness budget $\Gamma_i$ lies between zero and one and expresses how much of the deviation is protected in the schedule:

$$C_i \geq s_i + \bar{p}_i + \Gamma_i \hat{p}_i, 0 \leq \Gamma_i \leq 1$$

When $\Gamma_i$ is zero, the schedule uses nominal durations; when $\Gamma_i$ is one, the task is protected against the full deviation. The budget can be set per task class: deposition segments adjacent to free edges may receive higher protection, while inspection tasks may receive lower protection because their duration variability has less structural consequence. This is not a probabilistic guarantee unless a distribution is specified; it is a transparent feasibility margin whose cost can be evaluated in makespan and resource utilization.

The resulting schedule is executed in a rolling-horizon manner. At decision epoch t, the digital twin provides state $s_t$, the predictor updates duration and risk estimates, and the optimizer solves a finite-horizon problem. Only the near-term decisions are executed before the state is updated again. The reoptimization rule can be written as:

$$u_t = argmin_\{u \in U(s_t)\}\, \hat{J}_t(s_t, u) + \hat{V}_{t+h}(s_{t+h})$$

Here $u_t$ is the decision applied at epoch t, $U(s_t)$ is the feasible decision set under the current state, $\hat{J}_t$ is the predicted cost over the planning window, h is the horizon length, and $\hat{V}_{t+h}$ is a terminal value estimate that prevents myopic decisions. The terminal value may be a lower bound on remaining work, a relaxation of the scheduling model, or a learned value function. The rolling structure is essential because a construction-scale additive process produces new collision geometry and material states after every executed task.

## 5. Solution Procedure and Computational Design

The integrated model is a mixed-integer program with assignment, precedence, inventory, and energy constraints. Resource-constrained project scheduling is already computationally demanding [6], [7], and the present formulation adds mobile-robot compatibility and material-flow dynamics. A single monolithic solve over an entire building is therefore inappropriate for operational planning. The procedure decomposes the problem spatially and temporally: the building is partitioned into work zones, each zone generates a task subgraph, and the rolling horizon restricts the optimization to tasks that can be executed or materially affected within the window.

The decomposition must preserve coupling. A zone boundary that cuts through a structural overhang can hide a precedence relation; a material buffer that serves two zones can hide starvation risk. The framework therefore retains cross-zone precedence arcs and shared inventory variables in the window model. Spatial decomposition is used to reduce the number of robot-task compatibility tests, not to justify independent zone schedules. This is a technical distinction with practical consequences: independent zone schedules can each be feasible while the combined schedule is infeasible because two zones compete for the same pump, robot, or material batch.

For each window, the model is solved with a branch-and-bound or constraint-programming solver after linearizing shortage terms and disjunctive robot occupancy. The robust constraints are handled by adjusting effective processing times, which preserves the MILP structure. If the window is still too large, a dispatching heuristic can generate an initial sequence and the optimization model can be used to repair local violations. The heuristic is not assumed to be optimal; it provides a feasible incumbent and reduces the search space. The rolling-horizon terminal value then compensates for the truncation of tasks beyond the window.

The computational design also determines what “intelligent” means in this context. Intelligence is not a claim that a learned model replaces scheduling logic. It is the coordinated use of prediction, state estimation, and constrained optimization: prediction supplies task parameters and risk scores, the digital twin supplies state, and the optimizer enforces feasibility. This division of labor is appropriate because construction failures are often constraint violations—collisions, starvation, insufficient curing, unreachable segments—rather than purely statistical forecasting errors.

## 6. Technical Discussion and Boundary Conditions

The framework’s main analytical insight is that spatial decomposition, material supply, and robot allocation are mutually determining. A mesh or wireframe optimization that reduces geometric complexity can increase the number of short deposition tasks, which raises repositioning frequency and alters the material-demand profile [15]. Conversely, a schedule that delays a material replenishment to protect makespan may force a robot to idle, which then changes the feasible assignment of later tasks. The model captures these interactions through shared variables rather than through sequential post-processing.

Compared with fixed-sequence automation, the rolling-horizon model responds to state changes that fixed plans ignore. Compared with greedy dispatch rules, it anticipates precedence and inventory consequences. Compared with a pure project-scheduling model, it represents robot reachability, energy, and material flow. These comparisons are architectural

rather than empirical: the framework's value lies in making the relevant constraints explicit and in exposing the cost of ignoring them.

Several boundary conditions matter. First, the compatibility parameter $a_{ir}$ must be conservative. If reachability is overestimated, the optimizer will assign tasks that the robot cannot physically execute. Second, the material inventory model must respect pump and buffer dynamics; a simple inventory balance can underestimate starvation when material has a short pot life. Extrusion physics and process steps determine how quickly material can be delivered and how long it remains printable [18]. Third, the robust budget must be calibrated to the project's tolerance for delay. A budget that is too low produces brittle schedules; a budget that is too high produces excessive idle time and inventory.

The framework also has implications for multi-robot coordination. Task allocation in multi-robot systems depends on task structure, robot capabilities, and coordination requirements [20]. In construction, coordination is not only about dividing work; it is about sequencing work so that one robot does not block another's reach or consume material reserved for a precedence-critical task. The assignment and inventory constraints therefore perform a coordination function that a purely reactive controller cannot provide.

Field-oriented productivity analysis of robotically built walls shows that digital fabrication economics depend on utilization, setup, and process interruptions rather than on deposition speed alone [16]. In situ fabrication introduces additional challenges because the robot must operate within the evolving building rather than in a controlled factory cell [17]. Production scheduling and nesting for additive manufacturing demonstrate that geometric grouping and machine assignment materially affect delivery performance [19]. The present framework extends those concerns to mobile construction robots, where the machine, the material, and the workspace are all dynamic.

The model's limitations are structural rather than incidental. It assumes that tasks can be meaningfully discretized and that precedence relations can be specified before execution. In practice, some precedence emerges from inspection outcomes. The rolling horizon mitigates this by re-solving after each inspection, but it does not eliminate the need for human process knowledge. The model also assumes that energy and material consumption can be calibrated per task; if calibration is poor, the optimizer's decisions will inherit that error. Finally, the formulation is deterministic in its optimization layer once robustness budgets are chosen; it does not optimize a full stochastic policy. This is a deliberate trade-off between transparency and computational tractability.

## 7. Conclusion

This manuscript has developed an integrated scheduling and resource-allocation framework for automated robotic construction using additive manufacturing. The framework represents construction as a spatial task graph with precedence, robot compatibility, material inventory, and energy constraints, and it couples this representation to a learning-assisted predictor and a robust rolling-horizon optimizer. The central argument is that scheduling for robotic additive construction cannot be separated from spatial decomposition and material flow: geometry determines task structure, task structure determines resource demand, and resource demand determines feasibility.

The formulation yields several defensible design principles. Compatibility constraints should be state-dependent because reachability changes as the structure grows. Material inventory should be endogenous to the schedule because starvation is a scheduling failure, not

an external delay. Duration prediction should use asymmetric loss when late completion is more costly than early completion. Robustness budgets should be explicit and task-specific rather than hidden in conservative processing times. Rolling-horizon reoptimization is appropriate because the digital twin continually reveals new geometric and process states.

Practically, the framework supports planners in coordinating mobile robots, deposition rates, replenishment, inspection, and energy reserves before and during a build. Scientifically, it clarifies the boundary between prediction and optimization in intelligent construction: prediction estimates uncertain parameters, while constrained optimization enforces the technological and physical conditions under which robotic additive manufacturing remains feasible. The result is a technically coherent basis for scheduling decisions that are responsive to the evolving structure rather than merely reactive to individual robot commands.

**References**


1. Bock, T., & Linner, T. (2015). Robotic industrialization: Automation and robotic technologies for customized component, module, and building prefabrication. Cambridge University Press.

2. Khoshnevis, B. (2004). Automated construction by contour crafting—related robotics and information technologies. Automation in Construction, 13(1), 5–19.

3. Buswell, R. A., Leal de Silva, W. R., Jones, S. Z., & Dirrenberger, J. (2018). 3D printing using concrete extrusion: A roadmap for research. Cement and Concrete Research, 112, 37–49.

4. Wangler, T., Lloret, E., Reiter, L., Hack, N., Gramazio, F., Kohler, M., Bernhard, M., Dillenburger, B., Buchli, J., Roussel, N., & Flatt, R. (2016). Digital concrete: Opportunities and challenges. RILEM Technical Letters, 1, 67–75.

5. Zhang, X., Li, M., Lim, J. H., Weng, Y., Tay, Y. W. D., Pham, H., & Pham, Q.-C. (2018). Large-scale 3D printing by a team of mobile robots. Automation in Construction, 89, 146–156.

6. Hartmann, S., & Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. European Journal of Operational Research, 207(1), 1–14.

7. Kolisch, R. (1996). Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. European Journal of Operational Research, 90(2), 320–333.

8. Pinedo, M. L. (2016). Scheduling: Theory, algorithms, and systems (5th ed.). Springer.

9. Bertsimas, D., & Sim, M. (2004). The price of robustness. Operations Research, 52(1), 35–53.

10. Ben-Tal, A., El Ghaoui, L., & Nemirovski, A. (2009). Robust optimization. Princeton University Press.

11. Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.

12. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

13. Tao, F., Zhang, H., Liu, A., & Nee, A. Y. C. (2019). Digital twin in industry: State-of-the-art. IEEE Transactions on Industrial Informatics, 15(4), 2405–2415.

14. 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.

15. Yu, L., & Luo, D. (2018). Mesh optimisation for spatial wireframe robotic 3d printing. International Journal of Rapid Manufacturing, 7(2-3), 169-185.

16. Garcia de Soto, B., Agustí-Juan, I., Hunhevicz, J., Joss, S., Graser, K., Habert, G., & Adey, B. T. (2018). Productivity of digital fabrication in construction: Cost and time analysis of a robotically built wall. Automation in Construction, 92, 297–311.

17. Buchli, J., Giftthaler, M., Kumar, N., Lussi, M., Sandy, T., Dörfler, K., & Hack, N. (2018). Digital in situ fabrication: Challenges and opportunities for robotic in situ fabrication in architecture, construction, and beyond. Cement and Concrete Research, 112, 66–75.

18. Mechtcherine, V., Bos, F. P., Perrot, A., da Silva, W. R. L., Nerella, V. N., Fataei, S., Wolfs, R. J. M., Sonebi, M., & Roussel, N. (2020). Extrusion-based additive manufacturing with cement-based materials – Production steps, processes, and their underlying physics: A review. Cement and Concrete Research, 132, 106037.

19. Chergui, A., Hadj-Hamou, K., & Vignat, F. (2018). Production scheduling and nesting in additive manufacturing. Computers & Industrial Engineering, 126, 292–301.

20. Gerkey, B. P., & Matarić, M. J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. International Journal of Robotics Research, 23(9), 939–954.