Dynamic Single Additive Manufacturing Machine Scheduling Considering Postponement Decisions: A Lookahead Approximate Dynamic Programming

Hao Wu Chunlong Yu*

School of Mechanical Engineering, Tongji University

IE Forum 2025.12.14

Outline

Introduction: Economies of Scale in Additive Manufacturing

- AM Cost Bottlenecks: AM is hindered by high capital investment and significant per-batch operational expenses.
- Economies of Scale: Batching is essential as Fixed costs (e.g., recoating, inert gas, setup) are independent of batch size.

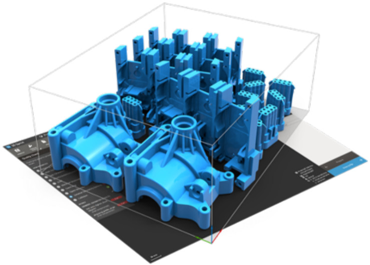


Figure: AM batching.

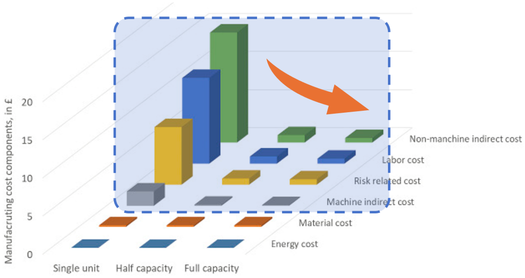


Figure: Cost breakdown showing impact of utilization.

Ding, J., Baumers, M., Clark, E. A., & Wildman, R. D. (2021). The economics of additive manufacturing: Towards a general cost model including process failure. *International Journal of Production Economics, 237*, 108087.

Introduction: Social Manufacturing as a Solution

- Cloud-based AM Platforms: Connect small service providers within a broader community.
- Solution to Cost Challenges: Pool geographically dispersed orders to form cost-effective batches.

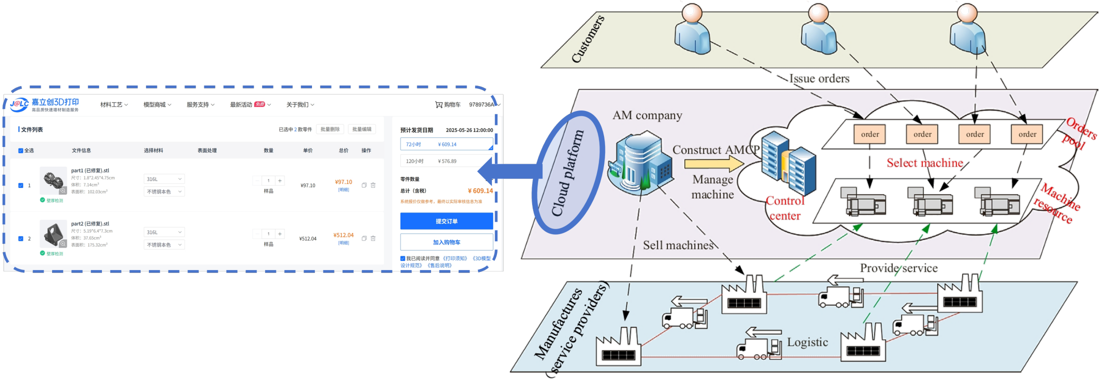


Figure: Social Additive Manufacturing Paradigm.

Wu, Q., Xie, N., Zheng, S., & Bernard, A. (2022). Online order scheduling of multi 3D printing tasks based on the additive manufacturing cloud platform. *Journal of Manufacturing Systems, 63.*

Operational Dilemma: To Process or To Postpone?

In a dynamic environment with stochastic order arrivals, a single AM machine operator faces a critical trade-off:

Process Immediately

Mitigates tardiness penalties and ensures on-time delivery.

Postpone Production

Consolidates with future orders for cost savings (higher density, fewer builds).

Research Question

How to optimally decide *when* to postpone and *what* to produce to maximize long-term profit?

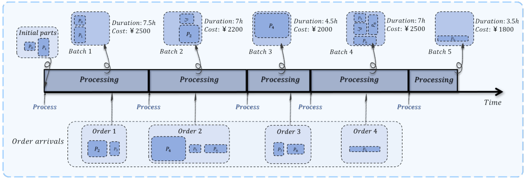
Van Mieghem, J. A., & Dada, M. (1998). Price versus production postponement: capacity and competition. *Management Science, 45*(12), 1631-1649.
Yang, B., & Burns, N. D. (2003). Implications of postponement for the supply chain. *International Journal of Production Research, 41*(9), 2075-2090.
Prataviera, L. B., Jazairy, A., & Abushaikha, I. (2024). Navigating the intersection between postponement strategies and additive manufacturing: insights and research agenda. *International Journal of Production Research*, 1-23.

Our Contribution: Addressing the Gap by Proactive Postponement

Research Gap: Existing studies employ reactive scheduling while overlook the benefits of strategic postponement, leading to suboptimal profitability.

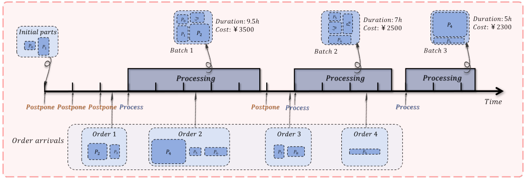
Reactive Scheduling

Triggered by machine idle or arrival.



Proactive Postponement

Deliberately waiting to gather information.



Li, Q., Zhang, D., Wang, S., & Kucukkoc, I. (2019). A dynamic order acceptance and scheduling approach for additive manufacturing on-demand production. *The International Journal of Advanced Manufacturing Technology*, 105, 3711-3729.

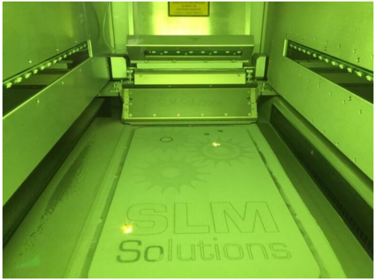
Wu, Q., Xie, N., Zheng, S., & Bernard, A. (2022). Online order scheduling of multi 3D printing tasks based on the additive manufacturing cloud platform. *Journal of Manufacturing Systems, 63.*

Sun, M., Ding, J., Zhao, Z., Chen, J., Huang, G. Q., & Wang, L. (2025). Out-of-order execution enabled deep reinforcement learning for dynamic additive manufacturing scheduling. *Robotics and Computer-Integrated Manufacturing*, 91, 102841.

Problem Description: AM Machine

- A single AM machine using *Selective Laser Melting* technique.
- Capable of processing a batch of parts simultaneously.
- Build chamber dimensions: Length $\mathcal{L}$, Width $\mathcal{W}$, Height $\mathcal{H}$.
- Parts not fitting the chamber are rejected in advance.

SLM®280 2.0

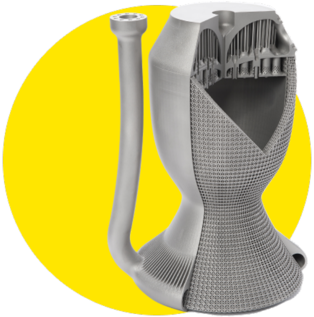


Build chamber.

Problem Description: AM Parts

Each part p is characterized by:

- Volume v_p.
- Support structure volume s_p.
- Height h_p.
- Minimum bounding box dimensions of its projection (l_p, w_p).
- Requested due-date d_p.



A monolithic thrust chamber manufactured by SLM.

Problem Description: AM Batching

- Nesting requirements must be satisfied:
  - Parts' bounding boxes must not overlap and fit platform boundaries.
  - Parts cannot be stacked.
  - Bounding boxes aligned parallel to platform axes.
- Processing time of a batch consists of:
  - Pre-build time t^{pre} (fixed).
  - Build time t^{bld} (variable, based on batch geometry and machine parameters).
  - Post-build time t^{post} (fixed).
- Non-preemptive:
  - Once a batch starts, it cannot be interrupted.
  - No parts can be added to or removed from an ongoing batch.
  - All parts in the same batch share the same completion time.

Lv, J., Peng, T., Zhang, Y., & Wang, Y. (2021). A novel method to forecast energy consumption of selective laser melting processes. *International Journal of Production Research*, 59(8), 2375-2391.

Yu, C., Matta, A., Semeraro, Q., & Lin, J. (2022). Mathematical models for minimizing total tardiness on parallel additive manufacturing machines. *IFAC-PapersOnLine*, 55(10), 1521-1526.

Problem Description: Time Horizon & Decision Points

- Entire time horizon divided into K discrete time slots, each of length u.
  - u is shorter than fixed batch times to ensure optimality.

- Decision points: $\{0, 1, \ldots, K\}$. Decisions only at these points.
  1. Should processing of the next batch be postponed (re-decided at $k+1$) or initiated immediately within current time slot?
  2. Which available parts should be selected for the next batch?

- Parts can only be scheduled after arrival; information unavailable until then.

Problem Description: Objective

- Maximize: Total Profit $= \sum$ Revenue - $\sum$ Processing Costs - $\sum$ Tardiness Fines.

1. Price ϱ_p of a part is estimated based on geometry and machine capability.
2. Cost of processing a batch:
   - Energy cost ($\propto$ build time t^{bld}).
   - Metal powder ($\propto$ total volume of build).
   - Inert gas ($\propto$ build time t^{bld}).
   - Operator cost (fixed for each batch, covers CAD prep, setup, part removal, post-processing).
3. Fine of tardiness:
   - A part p is tardy if not completed before its due-date d_p.
   - Each tardy part incurs a fine $\propto$ its tardiness T_p.

MDP Formulation: State S_k

The problem can be modeled as a Markov Decision Process (MDP).

State: the state of the system at decision epoch k is $S_k = (\mathbb{P}_k^{\text{ava}}, \mathbb{P}_k^{\text{pro}}, \tau_k, \mathbf{A}_k)$:

- $\mathbb{P}_k^{\text{ava}}$: Set of available parts waiting in queue.
- $\mathbb{P}_k^{\text{pro}}$: Set of parts currently being processed (if machine busy).
- τ_k: Machine availability time ($\tau_k \geq k \cdot u$).
- $\mathbf{A}_k$: Attributes of all parts.

MDP Formulation: Decision $\mathbf{X}_k$

Decision $\mathbf{X}_k = (Y_k, \{X_{k,p}\}_{p \in \mathbb{P}_k^{\mathsf{ava}}})$ at epoch k:

- $Y_k \in \{0, 1\}$: $\mathbf{1}$ to process immediately, $\mathbf{0}$ to postpone.
- $X_{k,p} \in \{0, 1\}$: $\mathbf{1}$ if part p is selected, $\mathbf{0}$ otherwise.

MDP Formulation: Decision space $\mathcal{X}_k$

- Machine Availability: If unavailable within current time slot, must postpone. $\tau_k - (k+1)u \leq M(1 - Y_k)$
- Batch Validity: If process ($Y_k = 1$), batch must be non-empty. $Y_k \leq \sum_{p \in \mathbb{P}_k} X_{kp}$

Two-dimensional nesting constraints

$$x_p + l_p(1 - o_p) + w_p o_p \leq \mathcal{L} + M(1 - X_{k,p}), \forall p \in \mathbb{P}_k^{\text{ava}}$$
$$y_p + w_p(1 - o_p) + l_p o_p \leq \mathcal{W} + M(1 - X_{k,p}), \forall p \in \mathbb{P}_k^{\text{ava}}$$
$$x_p + l_p(1 - o_p) + w_p o_p \leq x_{p'} + M(3 - X_{k,p} - X_{k,p'} - PL_{pp'}),$$
$$\forall p, p' \in \mathbb{P}_k^{\text{ava}}, p \neq p'$$
$$y_p + w_p(1 - o_p) + l_p o_p \leq y_{p'} + M(3 - X_{k,p} - X_{k,p'} - PB_{pp'}),$$
$$\forall p, p' \in \mathbb{P}_k^{\text{ava}}, p \neq p'$$
$$PL_{pp'} + PB_{pp'} + PL_{p'p} + PB_{p'p} \geq X_{k,p} + X_{k,p'} - 1,$$
$$\forall p, p' \in \mathbb{P}_k^{\text{ava}}, p < p'$$

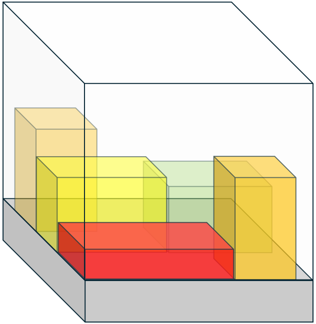


Figure: 2D Nesting with orthogonal rotation.

MDP Formulation: Transitions

Transition Function $S_{k+1} = S^M(S_k, \mathbf{X}_k, W_{k+1})$:

- Processing Set Update:

$$\mathbb{P}^{\text{pro}}_{k+1} = \begin{cases} \mathbb{P}^{\text{pro}}_{k}, & \text{if } \tau_k \geq (k+1)u \\ \{p \mid Y_k X_{k,p} = 1\}, & \text{if } \tau_k < (k+1)u \end{cases}$$

- Available Set Update:

$$\mathbb{P}^{\text{ava}}_{k+1} = (\mathbb{P}^{\text{ava}}_{k} \setminus \mathbb{P}^{\text{pro}}_{k+1}) \cup \mathbb{P}^{\text{arv}}_{k+1}$$

- Time Update:

$$\tau_{k+1} = \begin{cases} \tau_k + t^{\text{fixed}} + t^{\text{bld}}_k, & \text{if } Y_k = 1 \\ \max((k+1)u, \tau_k), & \text{if } Y_k = 0 \end{cases}$$

Exogenous Information W_{k+1}: Stochastic arrival of new orders $\mathbb{P}^{arv}_{k+1}$.

Objective Function: $\max \mathbb{E}\left\{\sum_{k=0}^{K} \mathcal{V}_k \mid S_0\right\}$, $\mathcal{V}_k$ = profit in slot k.

Challenges of Exact MDP Solution

The optimal policy is defined by the optimality equations:

Bellman's Equation

$$V_k(S_k) = \max_{\mathbf{X}_k \in \mathcal{X}_k} \left(\mathcal{V}(S_k, \mathbf{X}_k) + \mathbb{E}\{V_{k+1}(S_{k+1}) \mid S_k\} \right).$$

Solving this is computationally intractable due to:

"Three Curses of Dimensionality"

1. Explosion of the state space dimensionality (S_k).
2. Explosion of the decision space dimensionality $(\mathbf{X}_k)$.
3. Explosion of the outcome space dimensionality (W_{k+1}).

Bellman, R. (1954). The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6), 503-515.
Powell, W. B. (2007). *Approximate Dynamic Programming: Solving the curses of dimensionality.*

Why Direct Lookahead Approximation (DLA)?

Two classes of Approximate Dynamic Programming methods to approximate the expectation in Bellman's equation:

1. Value Function Approximations (VFAs): Effective when the value function has an exploitable structure.
2. Direct Lookahead Approximations (DLAs): Suitable when:
    - Few decisions per state.
    - Large or infinite set of random outcomes.

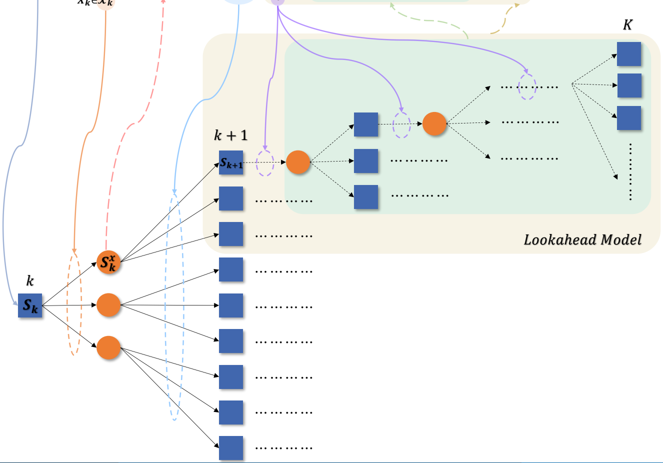
Our problem involves multiple layers (part selection, postponement), making functional approximation of state value difficult.
$\Longrightarrow$ Direct Lookahead Approximations

Powell, W. B. (2019). A unified framework for stochastic optimization. *European Journal of operational Research*, 275(3), 795-821.

Direct Lookahead Approximation Policy

- DLA rolls out using a lookahead model over the time horizon to capture the impact of current decisions on future activities.

$$X_k^*(S_k) = \underset{X_k \in \mathcal{X}_k}{\operatorname{argmax}} \left(\mathcal{V}_k(S_k, X_k) + \mathbb{E}_{W_{k+1}} \left\{ \max_{\pi} \mathbb{E}\left[\sum_{k'=k+1}^{K} \mathcal{V}_{k'}(S_{k'}, X^{\pi}(S_{k'})) \mid S_{k+1} \right] \middle| S_k^x \right\} \right)$$



DLA Policy Implementation Challenges

Implementing DLA presents computational challenges:

Lookahead Tree Estimation

- Enumerating the entire state-decision tree is intractable due to infinite random outcomes.

Solution: Monte Carlo Tree Search as a heuristic search.

Optimal Lookahead Policy

- Searching over all policies for optimal lookahead policy π is impossible.

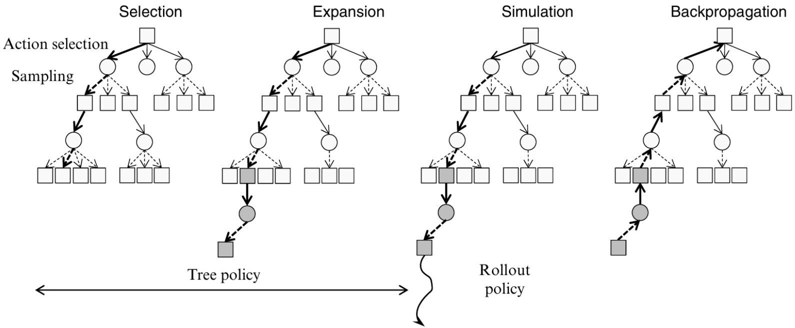
Solution: Design a reasonable lookahead policy $\tilde{\pi}$.

Decision Space Management

- Adding the full set of feasible decisions as branches is computationally prohibitive.

Solution: High-quality feasible decision space generation heuristic.

Heuristic Search: Monte Carlo Tree Search



- `Selection`: Traverse the tree to a promising *leaf* node using UCT for post-decision nodes and progressive sampling for next pre-decision nodes.
- `Expansion`: Add child post-decision nodes for feasible decisions.
- `Rollout`: From a *leaf* node, simulate a trajectory to the horizon using a fast lookahead policy ($\tilde{\pi}$).
- `Backup`: Propagate the simulation result back up the tree.

Kocsis, L., & Szepesvári, C. (2006). Bandit based Monte-Carlo planning. *Proceedings of the 17th European Conference on Machine Learning*, 282-293.

Design of Lookahead Policy $\tilde{\pi}$

To ensure efficiency during rollouts, we use a composite heuristic policy:

1. Temporal Decision (Process or Postpone?)

- **One-Step Lookahead Greedy:** Explicitly compares:
  - Reward of immediate production ($\hat{R}^{prod}$).
  - Expected reward of postponing one step and then producing ($\hat{R}^{post}$).
- If $\hat{R}^{post} > \hat{R}^{prod}$, postpone; else, process.

2. Combinatorial Decision (Which parts?)

- **Generate-then-Repair:**
  - Greedily add parts in non-decreasing order of *profit density* () until area limit.
  - If checked nesting infeasible, remove lowest profit density part until feasible.

Decision Space Approximation: Feasibility Generation

Challenge: Identifying all feasible batches is an NP-hard 2D Bin Packing problem.

1. **Fast Feasibility Check:** Skyline-based Best-Fit heuristic to verify if a set of parts fits in.
2. **Top-Down Search:** Starting from the largest possible batch size, exploits Downward Closure: If a batch is feasible, all its subsets are feasible.

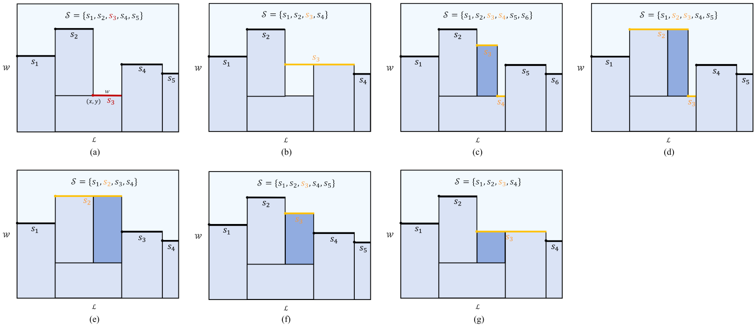


Figure: Skyline Best-Fit heuristic operations.

Decision Space Pruning: Pareto Filtering

"Curse of Width": Too many feasible decisions of producing batch dilutes MCTS search budget.

Evaluate every feasible batch on two criteria:

1. **Maximize:** Immediate Net Profit $f_1(\mathbf{X}_k) = \mathcal{V}_k(S_k, \mathbf{X}_k)$.
2. **Minimize:** Average Cost per Part $f_2(\mathbf{X}_k) = \frac{\mathcal{C}_k^{\text{pro}}}{|\{p|X_{k,p}=1\}|}$.

Only add Pareto-optimal batches + "Postpone" to MCTS tree.

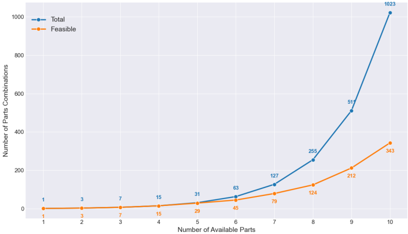


Figure: Number of feasible batches grows superlinearly.

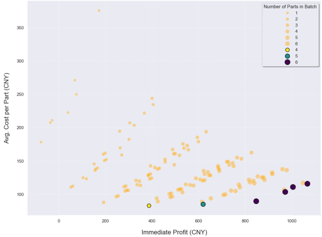


Figure: Non-dominated batches identified.

Numerical Results: Value of Strategic Postponement

We compared our ADP approach against a purely reactive **Process while Available (PwA)** policy.

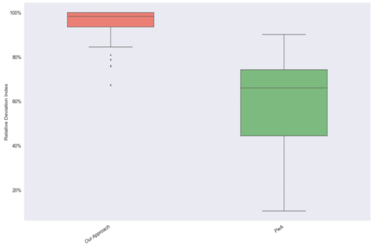


Figure: Relative Deviation Index Comparison.

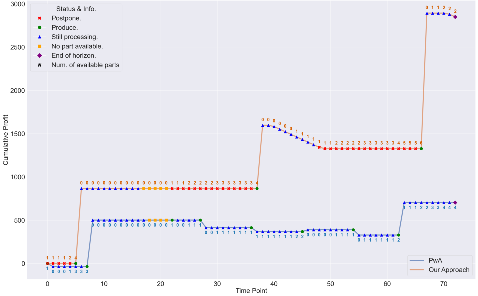


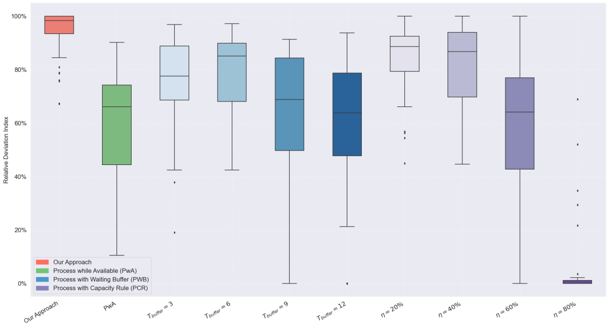
Figure: Cumulative profit trajectory comparison.

- **Instability of Reactive scheduling:** Coupling decisions with stochastic arrivals leads to low-fill, high-cost batches.
- **Superiority by Decoupling:** Postponement acts as a buffer, filtering short-term noise to form high-density batches.

Numerical Results: Value of Lookahead Mechanism

Comparison against static postponement rules:

- **PWB:** Process with Waiting Buffer (Wait fixed time T_{buffer}).
- **PCR:** Process with Capacity Rule (Wait until area $> \eta$).



- Lookahead Robustness: Our approach consistently outperforms best static rules without instance-specific tuning.
- Dynamic lookahead evaluates the *economic* opportunity cost, not just time or area.

Numerical Results: Benchmarking against Offline Optimum

To evaluate absolute quality, we formulated a deterministic MILP model with *Perfect Information* as an upper bound.

Instance Type	Horizon	Avg. Empirical Comp. Ratio (ECR)	Optimality Gap
Short	36h	1.23	Near Optimal
Medium	72h	1.35	Competitive
Long	144h	– (MILP Timeout)	Superior Scalability

Table: Comparison of Online ADP vs. Offline MILP.

- Average ECR $\approx$ 1.30 across solvable instances.
- Our approach provides high-quality solutions in real-time, whereas exact MILP fails for longer horizons.

Conclusions

Our Contributions

1. **Explicit Formulation:** Modeled single AM machine scheduling problem as a finite-horizon Markov Decision Process where postponement is an active strategic choice.
2. **Lookahead Approximation Policy:** Developed a Direct Lookahead Approximation policy utilizing Monte Carlo Tree Search for online decision-making.

Key Findings

- The Value of Postponement: Postponement decouples decisions with stochastic arrivals, filtering short-term noise to form high-density batches.
- The Value of Lookahead: The approach effectively identifies "windows of opportunity" to consolidate orders without incurring excessive delays.

Future Works

- Extend the framework to a parallel machine environment.
- Investigate time-dependent or non-stationary arrival.

Questions? Thank you!