Algorithms for Dynamic Scheduling in Manufacturing Towards Digital Factories

Ioan Hedea , i.m.hedea@student.tudelft.nl , Delft University of Technology
Responsible Professor: Mathijs de Weerdt, Supervisors: Léon Planken, Kim van den Houten

1. Research Question & Aim

Gap in PyJobShop: PyJobShop's solver handles flexible job-shop makespan, earliness and tardiness, but it does not natively model or enforce *hard deadlines* under uncertainty.

Objective: Extend the Flexible Job Scheduling Problem pipeline with dummy-task hard deadlines and STNU–RTE* real-time control to *guarantee* that jobs finish before their deadlines despite bounded duration variability.

Research Sub-Questions

1. **Slack bound**: Is $\Delta^\star = \max \sum_t (\bar{d}_t - \underline{d}_t)$ both necessary & sufficient?
2. **Weight tuning**: How do (w_e, w_t) shift the average earliness–tardiness Pareto front?
3. **Noise limit**: Up to what α can one policy keep $P_{tardy} < 0.32$?
4. **Runtime growth**: How does end-to-end wall-time scale with $|T|$?

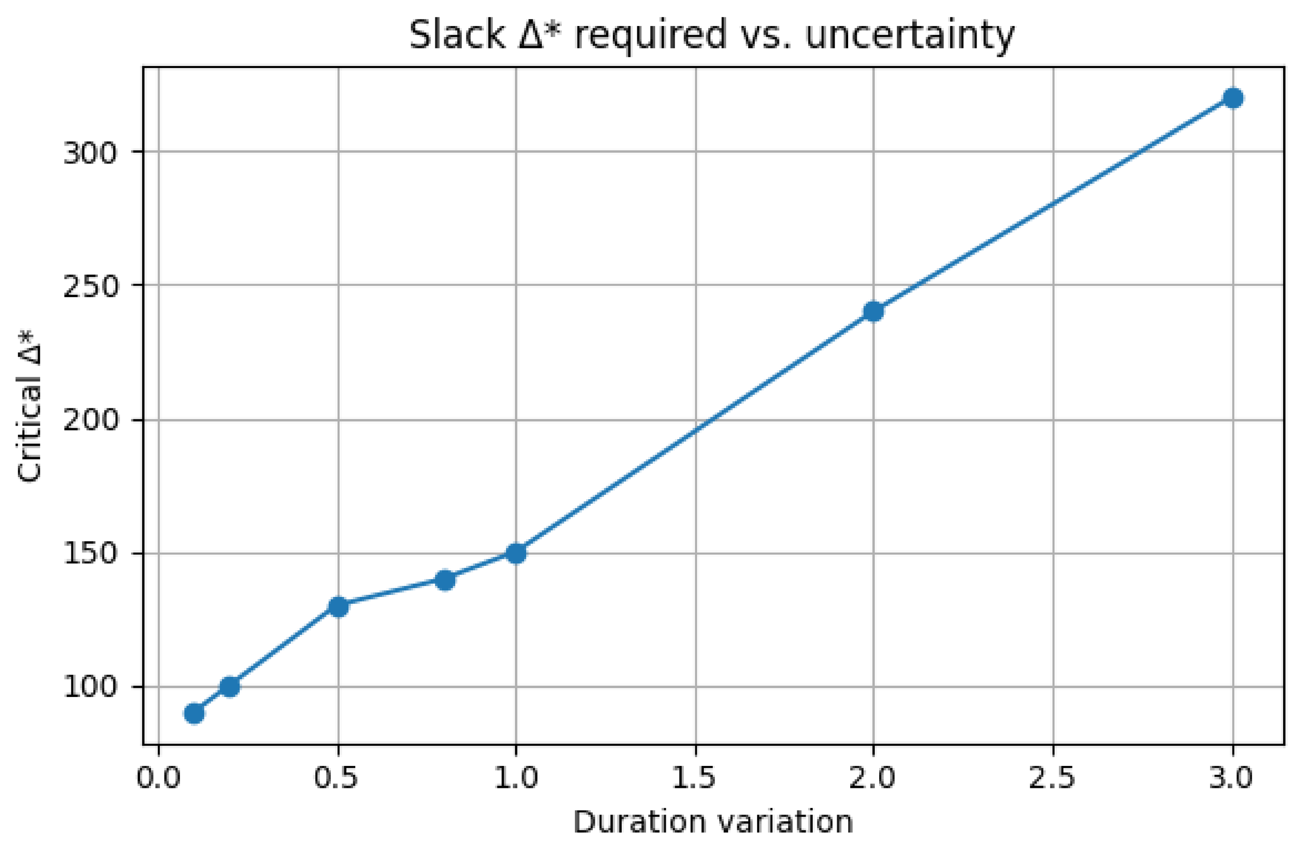
2. Methodology in Four Steps

1. **Offline CP design**
   - Flexible Job-Shop model with alternative machines.
   - Add one *dummy deadline task* per job.
   - Search a single slack $\Delta^\star$ that makes the CP model feasible.
   - Grid–sweep soft weights (w_e, w_t) for the earliness-tardiness Pareto.
2. **STNU build**
   - Map every task to *start* / *finish* nodes; add resource-chain edges.
   - Encode duration noise $d_t \sim \mathrm{U}[(1-\alpha)\underline{d}_t, (1+\alpha)\bar{d}_t]$, with $\alpha \in \{0.1, 0.2, 0.5, 0.8, 1.0, 2.0, 3.0\}$.
3. **Guarantee phase**
   - Check *dynamic controllability* (Java CSTNU tool).
   - If DC holds, hand schedule to the real-time dispatcher RTE*.
   - Run 500 Monte Carlo simulations for the STNU considering d_t.
4. **Evaluation**
   - *Quality metrics:* $C_{\max}$, mean earliness E, tardy prob. P_{tardy}.
   - *Runtime metrics:* CP solve time, DC check time, RTE* latency.
   - *Benchmarks:* public Kacem 1–4 suite (4–10 jobs, 12–55 ops, 5–10 machines).

Key References

[1] K. van den Houten *et al.*, "Proactive and Reactive Constraint Programming for Stochastic Project Scheduling with Maximal Time-Lags," 2024.
[2] P.H. Morris, N. Muscettola, T. Vidal, "Dynamic Controllability of STNUs," *IJCAI 2001*.
[3] I. Kacem, S. Hammadi, P. Borne, "Multi-objective Optimisation for the Flexible JSSP," *IEEE T-SMC C 32*, 2002.
[4] L. Hunsberger, R. Posenato, "The RTE* Dispatcher for STNUs," *ICAPS 2024*.
[5] R. Reijnen *et al.*, "Job-Shop Benchmark Environments and Instances," arXiv 2308.12794, 2025.

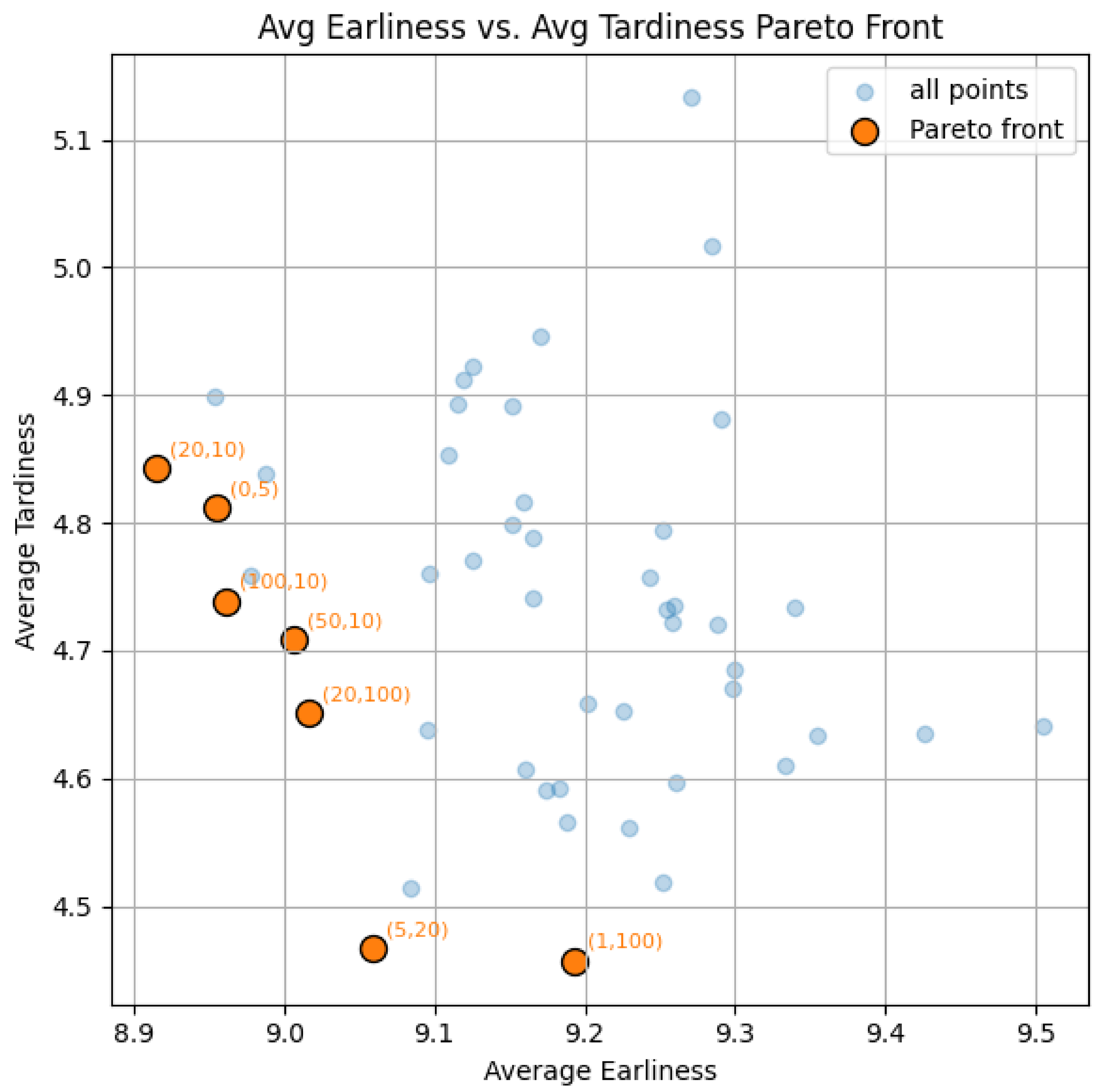
4. Hard-Deadline Slack Calibration (RQ1)



Duration Variations (x) vs. slack Δ (y).

- One global slack value per duration variation α guarantees *both* offline feasibility and online dynamic controllability.
- Closed-form bound $\Delta^\star = \max \sum_t (\bar{d}_t - \underline{d}_t)$ is tight to ± 10 tu on all instances.

5. Soft-Deadline Trade-off (RQ2)



Pareto front (w_e, w_t). Elbow w_e=5/w_t=20 achieves $P_{tardy} < 0.32$ for $< 2\%$ makespan hit at $\alpha = 0.6$.

- **Early bonus**: w_e=1 cuts P_{tardy} by 10 % and trims $C_{\max}$ 1 tu.
- **Sweet spot**: w_e=5 drops another 9 pp for +1 tu; gains flatten beyond.

6. Robustness vs Uncertainty (RQ3)

- Mean makespan rises 15 % from $\alpha = 0$ to 1.0 (linear degradation).
- Residual earliness drops below 50 % at $\alpha = 1$; P_{tardy} then climbs steeply (exceeds the RQ3 target of $P_{tardy} < 0.32$).
- *Heuristic trigger*: when shop-floor earliness < 0.5 of nominal slack, re-optimise with larger Δ.

7. Pipeline Scalability (RQ4)

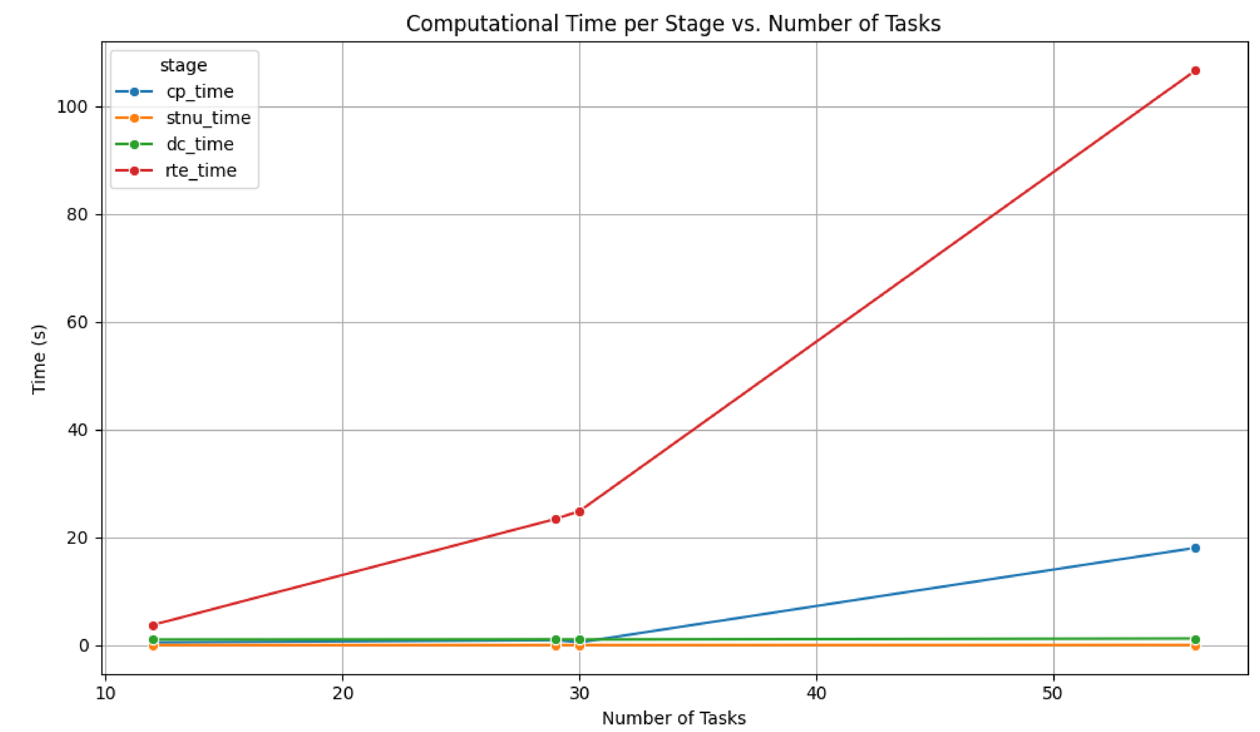


Fig. 3 — Near-linear wall-time vs. task count; DC check $< 1\%$ of total.

- CP: $\approx 0.33\,s \times |T|$; 500-run RTE*: $\approx 1.9\,s \times |T|$.
- 55-task Kacem-4 solved & simulated in 126 s on an M1-Pro laptop.

8. Limitations & Future Work

- **Uncertainty model** — Uniform i.i.d. bounds ignore correlation and heavy tails; log-normal draws already break DC on Kacem-4; move toward Gamma / log-normal fits and probabilistic STNUs.
- **Slack granularity** — Same $\Delta^\star$ for every job is safe but wasteful; per-job slack budgeting plus a "distance-to-DC" surrogate could trim margins 15–20 %.
- **Auto-tuning** — Current (w_e, w_t) grid search is brute-force; Bayesian or RL tuning could track drift on the shop floor in real time.
- **Industrial validation** — Replay the pipeline on real industrial data
- **Reactive benchmark** — Compare against rolling-horizon CP and rule-based dispatch on the same α-grid.

9. Conclusions and Recipe for Practitioners

- **Soft deadlines**: pick $w_e \in [5, 20]$, $w_t \in [0, 20]$ $P_{tardy} < 0.32$ and $< 5\%$ makespan loss on Kacem-3/4-size shops.
- **Hard deadlines**: set $D_j = \sum_t \min d_{jt} + \Delta^\star$ with $\Delta^\star = \max \sum(\bar{d} - \underline{d})$; guarantees DC.
- **Health trigger**: when on-line earliness falls below 50 % of nominal slack, rerun the CP+STNU loop with a larger Δ (empirically catches the $\alpha > 1$ failure mode).