AI and Data Science Journal
ISSN: 3078-7394
DOI: https://doi.org/10.61784/adsj4040

A DATA-DRIVEN MULTI-OBJECTIVE PRODUCTION SCHEDULING FRAMEWORK FOR BATTERY PACK MANUFACTURING UNDER MATERIAL ARRIVAL UNCERTAINTY

ChangHao Zhang*, YiNuo Wang
The University of Chicago, 5801 S. Ellis Ave., Chicago, IL 60637, United States.
**Corresponding Author: ChangHao Zhang*

Abstract: Material-arrival uncertainty can invalidate otherwise efficient battery-pack production schedules because cells, battery-management systems, enclosures, and cooling components become available at different assembly stages. This paper presents a reproducible data-to-decision framework that connects leakage-controlled arrival-delay learning with four-objective flexible-flow-shop scheduling. Median and 0.90-quantile histogram gradient-boosting models are trained on 180,519 public DataCo supply-chain records using a chronological 70/15/15 split. A split-conformal correction and stratified residual resampling generate scenario-dependent material availability without using post-arrival variables. The manufacturing layer is a transparent synthetic battery-pack benchmark with six assembly stages, two alternative machines per stage, three product families, sequence-dependent setups, due-date weights, and machine-energy tradeoffs. A calibrated quantile-residual NSGA-II (CQR-RNSGA-II) minimizes expected makespan, expected weighted tardiness, energy, and the 0.90 conditional value-at-risk of material-induced waiting. On the held-out logistics period, the learned 0.90 quantile reduces pinball loss by 32.1% relative to a global 90th-percentile buffer and attains 95.9% empirical upper coverage; the conformal correction is exactly zero because the raw discrete-target quantile is already conservative. Across nine independently seeded scheduling instances, the proposed method is best on the normalized four-objective compromise score for 20- and 30-job cases and essentially tied for 12 jobs. Relative to risk-neutral NSGA-II, it reduces material-wait CVaR by 12.1% at 20 jobs and 26.4% at 30 jobs, while revealing a nontrivial risk–tardiness tradeoff. All numerical claims are generated by the supplied code; no proprietary battery-factory data are asserted.
Keywords: Battery pack manufacturing; Conformalized quantile regression; Flexible flow shop; Material uncertainty; Multi-objective scheduling; NSGA-II; CVaR

1 INTRODUCTION

Battery-pack assembly combines high-value electrochemical components with mechanical, thermal, and electrical integration. Published process descriptions emphasize consecutive operations such as incoming inspection and cell grading, module or cell grouping, interconnection, battery-management-system integration, pack closure, and end-of-line testing [1-4]. These operations do not consume the same materials at the same time. Cell lots constrain early-stage work, battery-management-system hardware constrains electrical integration, and enclosures or cooling components constrain closure. Consequently, the production consequence of a supplier delay depends on both its magnitude and where the affected job is located in the sequence.

Classical scheduling provides mature models for precedence, machine assignment, due dates, and shop congestion [5-6]. Dynamic scheduling and rescheduling research further shows that production plans must be revised or protected when disruptions occur [7-8]. However, a practical gap remains between two decision layers. Supply-chain analytics typically predicts an arrival date or late-delivery probability, whereas production schedulers often assume deterministic release dates or impose a single global safety buffer. A point estimate suppresses tail risk; a global buffer ignores heteroscedasticity across shipping modes, markets, regions, product categories, and order characteristics.

A second gap concerns evaluation. Battery production data are commercially sensitive, and openly downloadable records rarely combine supplier arrival timestamps, detailed pack assembly operations, machine energy, due dates, and product families in one table. Treating an unrelated logistics dataset as if it were a battery-factory log would be misleading. The alternative adopted here is to separate what is observed from what is modeled: real public logistics records calibrate the arrival uncertainty, while shop-floor structure and process parameters are generated transparently from fixed-seed synthetic instances. This hybrid design permits reproducibility without making false claims about proprietary industrial validation.

This paper develops a calibrated quantile-residual scheduling framework with the following contributions. First, it estimates conditional median and upper-tail arrival delays using a chronological split and excludes actual shipping time, delivery status, and other post-arrival leakage variables. Second, it converts prediction residuals into context-stratified scenarios and adds a conformal stress scenario, thereby preserving both conditional location and empirical residual variation. Third, it embeds those scenarios in a six-stage battery-pack flexible flow shop and minimizes four objectives: expected makespan, expected weighted tardiness, energy, and the 0.90 conditional value-at-risk (CVaR) of material-induced waiting. Fourth, it provides an auditable experiment package containing code, raw predictions, raw scheduling observations, summaries, figures, software versions, fixed random seeds, and the SHA-256 digest of the source CSV (Figure 1).

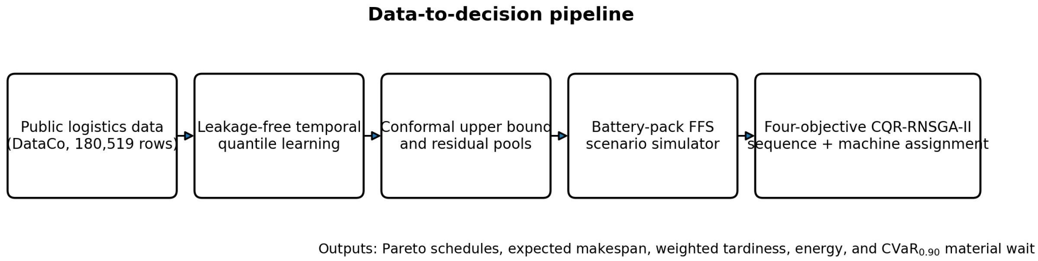

Figure 1 End-to-End Data-to-Decision Architecture

2 RELATED WORK

2.1 Battery-Pack Production Planning

Automotive battery production involves tightly coupled material, process, quality, and equipment decisions [1]. Kölmel et al. argued that quality characteristics should be integrated into battery assembly-system planning rather than appended after equipment selection [2]. The VDMA/PEM process brochure documents the sequencing and quality-control logic of module and pack assembly, while Kampker et al. show that battery architecture choices alter the structural composition and integration paths of the system [3-4]. These studies motivate a staged model with product families, alternative equipment, setup effects, and material gates. They do not, however, provide an open scheduling dataset with stochastic supplier arrivals and energy measurements.

2.2 Robust, Dynamic, and Energy-Aware Scheduling

Robust optimization protects decisions against uncertainty sets or scenario families [9-10]. CVaR offers a coherent tail-risk measure and is attractive when severe but infrequent delays matter more than average performance [11]. In production scheduling, Módos et al. considered robustness under energy-consumption limits, while operational energy reduction and energy key-performance indicators were studied [12-14]. Multi-objective energy-aware flexible-job-shop models explicitly expose the conflict between completion performance and power consumption [15].

Flexible-job-shop and flow-shop metaheuristics encode both sequencing and resource assignment. Representative foundations include tabu search, genetic algorithms, hybrid genetic-variable-neighborhood methods, and widely used scheduling benchmarks [16-19]. NSGA-II remains a practical baseline for multiple conflicting objectives because nondominated sorting and crowding preserve a diverse approximation to the Pareto set [20]. Nevertheless, most such studies assume deterministic material availability or inject uncertainty independently of a predictive model.

2.3 Data-Driven Uncertainty and Learning-Based Scheduling

Conformalized quantile regression combines conditional quantile prediction with a finite-sample calibration adjustment under exchangeability [21]. Calibration methods are important because nominal predictive probabilities can be systematically misaligned with empirical frequencies [22]. The present implementation uses scikit-learn to build histogram gradient-boosting quantile models [23]. Research integrating machine learning and production rescheduling demonstrates the value of combining predictive and optimization layers [24], and supply-chain work has shown that tree-based models can predict availability dates under disruption [25].

Learning-based scheduling is increasingly represented by surveys of integrated flexible-job-shop variants and reinforcement-learning policies for static or dynamic dispatching [27-30]. Such methods can support rapid online decisions, but training often requires large simulator budgets and the objective is commonly makespan or tardiness. The approach here is deliberately lighter: a conventional multi-objective evolutionary optimizer consumes calibrated arrival scenarios, making the risk model inspectable and separating predictive error from scheduling error. The distinct contribution is not a new generic optimizer; it is the calibrated interface from real arrival data to a four-objective battery-pack scheduling model.

3 PROBLEM DESCRIPTION AND FORMULATION

3.1 Manufacturing System

Let J={1,…,n} denote battery-pack jobs, S={1,…,6} assembly stages, and K_s={0,1} two alternative machines at each stage. The modeled stages are: (1) incoming inspection and cell grading; (2) grouping and mechanical positioning; (3) interconnection/welding; (4) battery-management-system and electrical integration; (5) enclosure/cooling integration and closure; and (6) end-of-line electrical and functional testing. Three material classes enter at stages 1, 4, and 5, respectively: cells, BMS/electrical hardware, and enclosure/cooling hardware. This abstraction follows published process logic but does not claim calibrated plant cycle times [1-4].

Each job j belongs to family $g_j \in \{0,1,2\}$, has release time r_j, due date d_j, and tardiness weight $w_j \in \{1,2,3\}$. Processing time p_{jsk} depends on job family, stage, and machine. Machine 0 is faster and higher-power; machine 1 is slower and more energy-efficient. If two consecutive jobs assigned to a machine have different families, a stage-specific setup time and setup-energy increment are charged. A chromosome therefore contains a job permutation π and a binary machine-allocation matrix $X=[x_{js}]$ (Table 1).

Table 1 Synthetic Shop-Floor Design (Fixed-Seed)

Element	Specification
Stages	6 ordered battery-pack assembly stages
Machines	2 alternatives per stage
Product families	3
Material gates	Cells: stage 1; BMS: stage 4; enclosure/cooling: stage 5
Job sizes	12, 20, and 30
Replications	3 independently seeded instances per size
Uncertainty unit	1 logistics delay day mapped to 2 production hours

3.2 Material-Arrival Scenarios

For job j and material m, let $a_{jm}(\xi)$ be the scenario-dependent arrival time and s(m) its consuming stage. The scheduled arrival is based on the job release and nominal cumulative processing before s(m), minus a random planning margin. The stochastic delay is formed from the conditional median prediction plus a residual sampled from a calibration or test residual pool matched by shipping mode and market. When the exact group is sparse, the hierarchy falls back to shipping mode and then to the global pool. Delays are clipped to the historical support [−2,4] days before nonnegative arrival times are constructed. The two-hours-per-day scale maps coarse public logistics days into a compact production horizon; it is a modeling scale, not an empirical conversion rate.

Given a fixed chromosome, the earliest start of operation (j,s) in scenario ξ is the maximum of job precedence, machine availability, release time, and—when s is a material gate—the relevant arrival:

$$B_{js}(\xi) = \max\{C_{j,s-1}(\xi),\ M_{s,x_{js}}(\xi),\ r_j,\ a_{jm}(\xi)\cdot 1[s=s(m)]\}. \tag{1}$$

Completion is $C_{js}(\xi)=B_{js}(\xi)+u_{js}+p_{js,x_{js}}$, where u_{js} is the family-change setup time. The final completion is $C_j(\xi)=C_{j6}(\xi)$. Material-induced waiting accumulates only the positive difference between material readiness and the otherwise feasible operation start, avoiding double counting general machine congestion as supply risk.

3.3 Four Objectives

The objective vector $F=(f_1,f_2,f_3,f_4)$ is minimized. For scenario set Ω, the first two components are expected makespan and expected weighted tardiness:

$$f_1 = |\Omega|^{-1} \Sigma_{\xi\in\Omega} \max_j C_j(\xi), \tag{2}$$

$$f_2 = |\Omega|^{-1} \Sigma_{\xi\in\Omega} \Sigma_{j\in J}\, w_j \max\{0,\ C_j(\xi)-d_j\}. \tag{3}$$

Energy is deterministic for a fixed chromosome because arrival uncertainty changes idle time but not processing duration or selected machine in the present model:

$$f_3 = \Sigma_{j,s}\, p_{js,x_{js}} P_{s,x_{js}} + \Sigma_{\text{family changes}}\, e_s^{setup}. \tag{4}$$

Let W(ξ) denote total material-induced waiting across all jobs and gates. Tail exposure is measured by empirical CVaR at β=0.90 [11]:

$$f_4 = CVaR_{0.9}[W] = \mathrm{mean}\{W(\xi):\ W(\xi) \geq Q_{0.9}(W)\}. \tag{5}$$

This risk measure differs from minimizing average delay. A sequence may perform well under most scenarios but place many jobs behind a shared late component in the upper tail. The fourth objective makes that vulnerability visible rather than burying it in the expected tardiness objective.

4 DATA-DRIVEN SCHEDULING FRAMEWORK

4.1 Leakage-Controlled Quantile Learning

The public DataCo dataset contains 180,519 records [26]. Arrival delay is defined as actual shipping days minus scheduled shipping days. Records are ordered chronologically and divided into 70% training (126,363), 15% calibration (27,078), and 15% testing (27,078). No random shuffle is used. Predictors available before realized delivery are shipping mode, market, order region, category, customer segment, scheduled shipping days, item quantity, product price, order month, and order day of week. Actual shipping days, delivery status, late-delivery risk labels, and other realized-outcome fields are excluded.

Two histogram gradient-boosting regressors estimate $\hat{q}_{0.50}(x)$ and $\hat{q}_{0.90}(x)$. Categorical variables are most-frequent imputed and ordinal encoded with an explicit unknown category; numeric variables are median imputed. Both models use quantile loss, learning rate 0.07, 180 boosting iterations, up to 31 leaves, minimum leaf size 40, and L2 regularization 0.15. These hyperparameters were fixed before the reported test evaluation; no test-period tuning is performed.

For calibration observations $i=1,\dots,n\_c$, one-sided nonconformity scores are $e\_i=y\_i-\hat{q}\_0.90(x\_i)$. At $\alpha=0.10$, the split-conformal correction is

$$\delta = \text{Quantile}\lceil (n\_c+1)(1-\alpha)\rceil /n\_c (\{e\_i\}), \quad \tilde{q}\_0.90(x)=\hat{q}\_0.90(x)+\delta. \tag{6}$$

The resulting upper predictor is used as a stress scenario, while $\hat{q}\_0.50$ and empirical residual pools drive the remaining scenarios. In the actual run, δ=0.0 day. This is not manually forced: the discrete delay target and conservative raw upper quantile already place the required calibration residual quantile at zero. Reporting the zero adjustment is important because it prevents the term “conformal” from being used as an unsupported claim of numerical improvement (Figure 2).

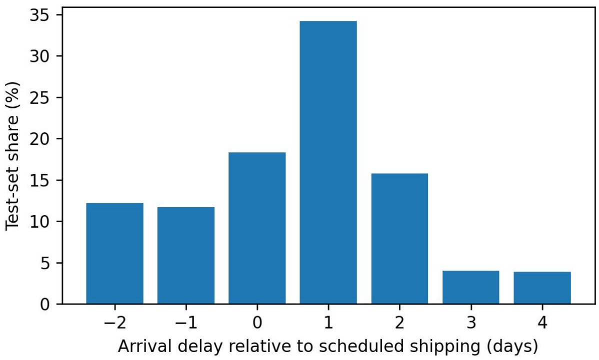

Figure 2 Held-Out Arrival-Delay Distribution (27,078 Records)

4.2 Scenario Construction

For each synthetic job-material pair, a held-out logistics row supplies the median and conformal upper predictions plus a shipping-mode/market group key. Calibration scenarios use $\hat{q}\_0.50(x)$ plus residuals resampled from the calibration period; the last of eight scenarios is replaced by $\tilde{q}\_0.90(x)$, producing a deliberate upper-tail stress case. Evaluation uses 80 independently drawn residual scenarios from the held-out test period. Therefore, optimization and evaluation do not share residual observations. This separation is analogous to training-validation-test discipline in predictive modeling and limits optimistic scenario reuse.

Three optimizer inputs are compared. Risk-neutral NSGA-II (RN-NSGA-II) uses one median-arrival scenario and objectives (f_1,f_2,f_3). Static-buffer NSGA-II (SB-NSGA-II) uses a uniform two-day delay—the empirical training-period 90th percentile—and the same three objectives. The proposed CQR-RNSGA-II uses eight calibration scenarios, including the conformal stress scenario, and all four objectives. EDD assigns jobs by due date to the faster machines, while SPT-E assigns jobs by efficient-machine processing time and uses the energy-efficient machines.

4.3 Multi-Objective Search

NSGA-II encodes the job permutation and the $n\times6$ binary machine matrix [20]. Order crossover preserves a valid permutation; uniform crossover acts on machine assignments. Mutation swaps two jobs with probability 0.35 and flips machine bits with a size-adjusted probability. Binary tournaments use nondomination rank and crowding distance. The reported budget is a population of 20 for 20 generations. The final compromise solution is selected from the nondominated set by the minimum root-mean-square normalized distance to the within-front ideal point. This selection is applied consistently to all NSGA-II variants.

Algorithm 1 CQR-RNSGA-II (Reported Configuration)

Step	Operation
1	Train $\hat{q}0.50$ and $\hat{q}0.90$ on chronological training data.
2	Compute one-sided conformal correction on calibration data.
3	Build context residual pools and generate 7 residual scenarios + 1 upper stress scenario.
4	Initialize 20 sequence/machine chromosomes.
5	Evaluate expected makespan, expected weighted tardiness, energy, and CVaR0.90 waiting.
6	Apply nondominated sorting, crowding, crossover, and mutation for 20 generations.
7	Choose the nondominated schedule nearest the normalized ideal; evaluate on 80 held-out scenarios.

5 EXPERIMENTAL DESIGN

5.1 Reproducibility and Data Provenance

The dataset source is DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS, Version 5 [26]. The supplied run manifest records source-file SHA-256 994b3c8d24049cc46bf20e8161fedece9a9b95376cc0dc929921d76cf6b9bc0d, Python 3.13.5, NumPy 2.3.5, and pandas 2.2.3. The code archive does not redistribute the approximately 96 MB raw

CSV; the README points to the DOI and accepts an environment-variable path. Predictions and calibration residuals are included so that scheduling results remain inspectable.

The shop-floor benchmark is synthetic by design. For each size $n \in \{12,20,30\}$, seeds 101, 203, and 307 produce three instances, for nine total. Each instance has six stages and two machines per stage. Base processing times range from 0.65 to 1.25 hours before family, machine-speed, and lognormal perturbation multipliers. Faster machines draw more power; efficient machines process 16% more slowly. Stage-specific setup times range from 0.08 to 0.18 hour. These values create controlled tradeoffs but are not represented as measured battery-plant parameters.

5.2 Evaluation Protocol

Every selected schedule is reevaluated on the same 80 held-out scenarios for its instance. Reported table entries are means and sample standard deviations across the three independently seeded instances at each job size. Because $n=3$ is insufficient for reliable asymptotic inference, paired one-sided Wilcoxon tests are archived only as exploratory diagnostics. The smallest p-value is 0.125; no result is labeled statistically significant. The main analysis therefore uses effect direction, effect magnitude, and consistency across scales.

A normalized compromise score supports compact comparison. For each job size and each objective, all 15 method-instance observations are min-max normalized; the four normalized values are averaged. Lower is better. This score is not a replacement for the Pareto objectives and can depend on the comparator set, so the unnormalized metrics are reported alongside it.

6 RESULTS

6.1 Arrival-Model Accuracy and Calibration

Table 2 Held-Out Arrival-Model Results

Metric	Value
Test records	27,078
Median MAE (days)	0.983
Median pinball loss	0.492
Learned q0.90 pinball loss	0.177
Global q0.90 pinball loss	0.261
Learned q0.90 coverage	95.9%
Global q0.90 coverage	92.1%
Conformal correction	0.0 day

Table 2 shows that conditional upper-quantile learning materially improves the asymmetric loss. Pinball loss decreases from 0.2608 for a global two-day 90th-percentile buffer to 0.1771, a 32.1% reduction. Empirical upper coverage is 95.9%, above the nominal 90%, whereas the global buffer covers 92.1%. Thus the learned model is sharper according to pinball loss yet conservative in frequency. The calibration curve in Figure 3 confirms overcoverage around the upper quantiles. Since split conformal returns $\delta=0$, the calibrated and raw 0.90 coverage are identical in this experiment.

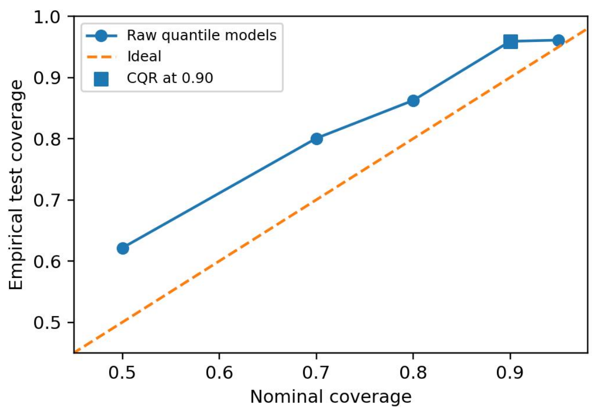

Figure 3 Nominal Versus Empirical Test Coverage. The q0.90 conformal point coincides with the raw model because $\delta=0$.

No method dominates all four criteria (Table 3). SPT-E consistently has the lowest energy but the longest makespan and highest tardiness. EDD improves timeliness relative to SPT-E, yet remains substantially worse than the evolutionary schedules on makespan, tardiness, and energy. The static two-day buffer is also not uniformly robust: SB-NSGA-II has the highest material-wait CVaR among the three evolutionary variants at every scale, demonstrating that a uniform conservative offset does not necessarily protect the production sequence from context-dependent tail delays.

6.2 Scheduling Performance

Table 3 Out-of-Sample Scheduling Results, Mean ± SD Across Three Instances

Jobs	Method	Makespan (h)	Weighted tardiness (h)	Energy (kWh)	CVaR0.90 wait (h)
12	CQR-RNSGAII	20.45±1.01	98.9±30.1	593.1±19.6	9.40±1.19
12	RN-NSGAII	20.32±0.37	91.4±13.2	588.6±15.1	10.91±1.67
12	SB-NSGAII	22.04±1.69	114.1±24.2	588.4±11.1	14.48±3.64
12	EDD	24.98±1.10	135.2±28.5	667.7±12.3	9.39±2.44
12	SPT-E	27.01±2.21	162.7±55.1	536.6±16.8	5.59±2.60
20	CQR-RNSGAII	27.25±0.93	132.9±13.3	1005.5±7.5	10.82±2.21
20	RN-NSGAII	28.18±0.30	151.8±9.3	991.7±10.6	12.42±1.30
20	SB-NSGAII	30.08±2.11	203.1±65.8	986.1±8.5	12.98±3.55
20	EDD	35.35±0.47	273.8±43.3	1120.2±16.0	8.80±2.07
20	SPT-E	41.15±0.42	388.6±41.1	898.5±3.0	8.63±1.15
30	CQR-RNSGAII	40.10±1.39	504.3±73.7	1508.5±35.0	9.40±4.25
30	RN-NSGAII	36.70±2.60	406.0±97.9	1511.7±9.4	12.61±2.22
30	SB-NSGAII	38.96±1.92	443.0±41.3	1503.3±25.6	15.30±0.63
30	EDD	50.48±1.21	775.8±25.9	1701.9±31.6	8.54±0.79
30	SPT-E	56.06±1.31	974.6±129.5	1353.8±25.6	7.76±1.46

The proposed method is most favorable at 20 jobs. Paired by instance against RN-NSGA-II, it reduces expected makespan by 3.33%, weighted tardiness by 12.08%, and material-wait CVaR by 12.09%, with a 1.39% energy increase. At 12 jobs, the proposed method reduces material-wait CVaR by 13.24% but incurs 0.61% more makespan, 11.74% more tardiness, and 0.76% more energy. At 30 jobs, the risk reduction grows to 26.41% and energy decreases by 0.21%, but makespan and tardiness increase by 9.75% and 31.63%, respectively. These results are a genuine tradeoff rather than evidence of universal superiority.

6.3 Compromise Quality and Pareto Structure

The mean normalized compromise scores for CQR-RNSGA-II are 0.293, 0.240, and 0.348 at 12, 20, and 30 jobs. The corresponding RN-NSGA-II values are 0.293, 0.294, and 0.353. Therefore the methods are effectively tied at 12 jobs, while the proposed method ranks first at 20 jobs and narrowly first at 30 jobs. The modest 30-job score difference should be interpreted with the individual objectives: risk improves strongly, whereas timeliness degrades.

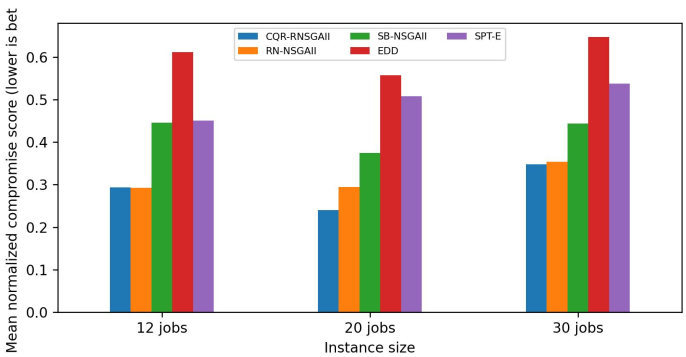

Figure 4 Mean Normalized Four-Objective Compromise Score

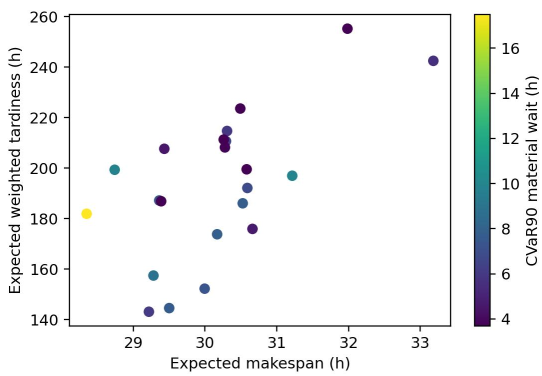

Figure 5 Sample 20-Job Nondominated Neighborhood; Color Encodes CVaR0.90 Material Wait

The sample nondominated neighborhood for a 20-job instance in Figure 4-5 visualizes the source of this behavior. Schedules near 28–30 hours of makespan can have widely different tardiness and material-tail exposure. Several lower-CVaR schedules require higher tardiness or slightly higher energy. This structure supports presenting a Pareto set to a production planner rather than deploying a single opaque scalarized rule.

7 DISCUSSION

7.1 Technical and Managerial Implications

The central technical result is that predictive sharpness and scheduling robustness are related but not interchangeable. The learned upper quantile has lower pinball loss than a global buffer, yet its coverage is already conservative and the conformal correction is zero. Scheduling gains arise from combining conditional medians, context-residual distributions, an explicit tail scenario, and a CVaR objective—not from adding an arbitrary positive conformal margin. This distinction matters for auditability: a calibration procedure may legitimately return no correction, while the residual distribution still contains decision-relevant heterogeneity.

For operations managers, the four objectives separate three levers. Job sequencing controls due-date exposure and where delayed materials block the line. Machine assignment controls speed-energy tradeoffs. The risk objective controls concentration of material-induced waiting in adverse scenarios. The 30-job result shows why this separation is useful: a risk-averse plan can materially reduce tail waiting while accepting longer completion. A planner facing premium late-delivery penalties may prefer the risk-neutral schedule; a planner facing line-stoppage escalation or supplier volatility may select a lower-CVaR alternative.

7.2 Validity Boundaries

Four limitations define the scope of the conclusions. First, DataCo is a general supply-chain dataset, not a battery-component inbound-logistics dataset. It supports the methodology and residual-calibration experiment, not direct industrial effect-size transfer. Second, processing, power, setup, due-date, and family parameters are synthetic. Energy is an internally consistent processing-power proxy and excludes HVAC, standby, rework, and demand charges. Third, logistics delays are recorded in whole days and are rescaled to production hours; richer timestamp data would permit event-level calibration. Fourth, the scheduling study uses nine instances and a moderate evolutionary budget. The archived Wilcoxon tests correctly show that this sample cannot establish statistical significance.

Future validation should replace each synthetic layer independently: real supplier advanced-shipping notices for the arrival model, manufacturing-execution-system event logs for processing and setup distributions, meter data for energy, and actual service-level penalties for objective weighting. Online rescheduling could then use rolling conformal calibration to monitor coverage drift and trigger selective reoptimization, complementing dynamic and learning-based scheduling approaches [7, 24, 28-30].

8 CONCLUSION

This paper proposed a reproducible framework that links conditional arrival-delay learning to multi-objective battery-pack production scheduling. The framework uses a leakage-controlled chronological split, quantile gradient boosting, one-sided split-conformal calibration, context-stratified residual scenarios, and a four-objective NSGA-II. Real public logistics records calibrate uncertainty, while fixed-seed synthetic battery-pack instances make the shop-floor assumptions explicit. The learned upper quantile reduces pinball loss by 32.1% versus a global buffer. In scheduling, the proposed method lowers material-wait tail risk, is best on the aggregate compromise score at 20 and 30 jobs, and exposes the cost of robustness at the largest scale. The main claim is therefore not universal dominance, but a transparent and testable way to turn arrival uncertainty into Pareto scheduling decisions without inventing proprietary factory evidence.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

DATA AND CODE AVAILABILITY

The experiment archive accompanying this manuscript contains the complete Python source, exact reproduction entry point, fixed seeds, requirements, raw prediction and residual files, raw scheduling observations, summaries, figures, reference-verification list, and run manifest. The original DataCo CSV is obtainable from Mendeley Data at DOI 10.17632/8gx2fvg2k6.5 and is not redistributed in the archive [26]. The manuscript should not be submitted with the affiliation placeholder unchanged.

REFERENCES

[1] Kwade A, Haselrieder W, Leithoff R, et al. Current status and challenges for automotive battery production technologies. Nature Energy, 2018, 3: 290–300. DOI: 10.1038/s41560-018-0130-3.

[2] Kölmel A, Sauer A, Lanza G. Quality-oriented production planning of battery assembly systems for electric mobility. Procedia CIRP, 2014, 23: 149–154. DOI: 10.1016/j.procir.2014.10.075.

[3] VDMA Battery Production and PEM of RWTH Aachen University. Battery Module and Pack Assembly Process. 3rd ed. Frankfurt, Germany, 2018.

[4] Kampker A, Heimes H H, Offermanns C, et al. Framework and classification of battery system architectures. World Electric Vehicle Journal, 2023, 14(4): 88. DOI: 10.3390/wevj14040088.
[5] Pinedo M L. Scheduling: Theory, Algorithms, and Systems. 5th ed. Cham, Switzerland: Springer, 2016. DOI: 10.1007/978-3-319-26580-3.
[6] Jain A S, Meeran S. Deterministic job-shop scheduling: Past, present and future. European Journal of Operational Research, 1999, 113(2): 390–434. DOI: 10.1016/S0377-2217(98)00113-1.
[7] Ouelhadj D, Petrovic S. A survey of dynamic scheduling in manufacturing systems. Journal of Scheduling, 2009, 12: 417–431. DOI: 10.1007/s10951-008-0090-8.
[8] Vieira G E, Herrmann J W, Lin E. Rescheduling manufacturing systems: A framework of strategies, policies, and methods. Journal of Scheduling, 2003, 6(1): 39–62. DOI: 10.1023/A:1022235519958.
[9] Bertsimas D, Brown D B, Caramanis C. Theory and applications of robust optimization. SIAM Review, 2011, 53(3): 464–501. DOI: 10.1137/080734510.
[10] Ben-Tal A, El Ghaoui L, Nemirovski A. Robust Optimization. Princeton, NJ, USA: Princeton University Press, 2009. DOI: 10.1515/9781400831050.
[11] Rockafellar R T, Uryasev S. Optimization of conditional value-at-risk. Journal of Risk, 2000, 2(3): 21–41. DOI: 10.21314/JOR.2000.038.
[12] Módos I, Šůcha P, Hanzálek Z. Algorithms for robust production scheduling with energy consumption limits. Computers & Industrial Engineering, 2017, 112: 391–408. DOI: 10.1016/j.cie.2017.08.011.
[13] Mouzon G, Yildirim M B, Twomey J. Operational methods for minimization of energy consumption of manufacturing equipment. International Journal of Production Research, 2007, 45(18–19): 4247–4271. DOI: 10.1080/00207540701450013.
[14] May G, Barletta I, Stahl B, et al. Energy management in production: A novel method to develop key performance indicators for improving energy efficiency. Applied Energy, 2015, 149: 46–61. DOI: 10.1016/j.apenergy.2015.03.065.
[15] Dai M, Tang D, Giret A, et al. Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints. Robotics and Computer-Integrated Manufacturing, 2019, 59: 143–157. DOI: 10.1016/j.rcim.2019.04.006.
[16] Brandimarte P. Routing and scheduling in a flexible job shop by tabu search. Annals of Operations Research, 1993, 41: 157–183. DOI: 10.1007/BF02023073.
[17] Pezzella F, Morganti G, Ciaschetti G. A genetic algorithm for the flexible job-shop scheduling problem. Computers & Operations Research, 2008, 35(10): 3202–3212. DOI: 10.1016/j.cor.2007.02.014.
[18] Gao J, Sun L, Gen M. A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. Computers & Operations Research, 2008, 35(9): 2892–2907. DOI: 10.1016/j.cor.2007.01.001.
[19] Taillard E. Benchmarks for basic scheduling problems. European Journal of Operational Research, 1993, 64(2): 278–285. DOI: 10.1016/0377-2217(93)90182-M.
[20] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182–197. DOI: 10.1109/4235.996017.
[21] Romano Y, Patterson E, Candès E J. Conformalized quantile regression. In: Advances in Neural Information Processing Systems, 2019.
[22] Kuleshov V, Fenner N, Ermon S. Accurate uncertainties for deep learning using calibrated regression. In: Proceedings of the 35th International Conference on Machine Learning, PMLR, 2018, 80: 2796–2804.
[23] Pedregosa F. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 2011, 12: 2825–2830.
[24] Li Y, Carabelli S, Fadda E, et al. Machine learning and optimization for production rescheduling in Industry 4.0. International Journal of Advanced Manufacturing Technology, 2020, 110: 2445–2463. DOI: 10.1007/s00170-020-05850-5.
[25] Camur M C, Ravi S K, Saleh S. Enhancing supply chain resilience: A machine learning approach for predicting product availability dates under disruption. arXiv:2304.14902, 2023. DOI: 10.48550/arXiv.2304.14902.
[26] Constante F, Silva F, Pereira A. DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS. Mendeley Data, Version 5, 2019. DOI: 10.17632/8gx2fvg2k6.5.
[27] Li X, Guo X, Tang H, et al. Survey of integrated flexible job shop scheduling problems. Computers & Industrial Engineering, 2022, 174: 108786. DOI: 10.1016/j.cie.2022.108786.
[28] Zhang C, Song W, Cao Z, et al. Learning to dispatch for job shop scheduling via deep reinforcement learning. In: Advances in Neural Information Processing Systems 33, 2020.
[29] Liu R, Piplani R, Toro C. Deep reinforcement learning for dynamic scheduling of a flexible job shop. International Journal of Production Research, 2022, 60(13): 4049–4069. DOI: 10.1080/00207543.2022.2058432.
[30] Zhang L, Feng Y, Xiao Q, et al. Deep reinforcement learning for dynamic flexible job shop scheduling problem considering variable processing times. Journal of Manufacturing Systems, 2023, 71: 257–273. DOI: 10.1016/j.jmsy.2023.09.009.