福州大学经济与管理学院
School of Economics and Management Fuzhou University

schedulingseminar.com

Feature-driven Robust Stochastic Scheduling for Printed Circuit Board Assembly

Debiao Li

School of Economics and Management

Fuzhou University

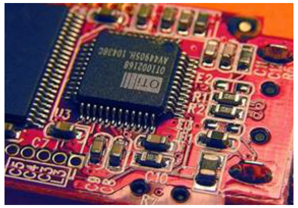
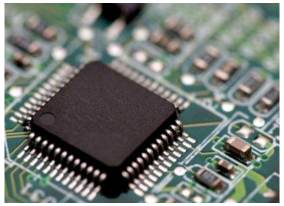
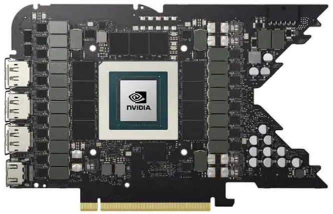
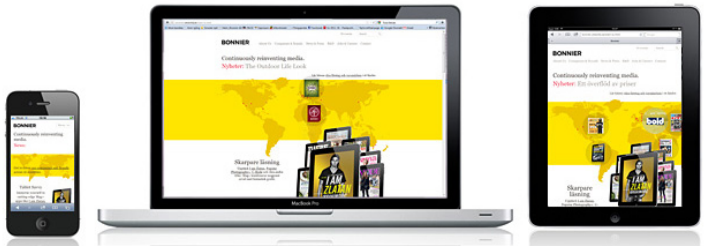
2026-05-13

Outline

1. **PCB assembly scheduling Problem**
2. **Feature-driven robust stochastic scheduling problem**
3. **Feature-driven scheduling model**
4. **Branch-and-price algorithm**
5. **Computational results and analysis**
6. **Summary and future work**

1.1 PCB Assembly

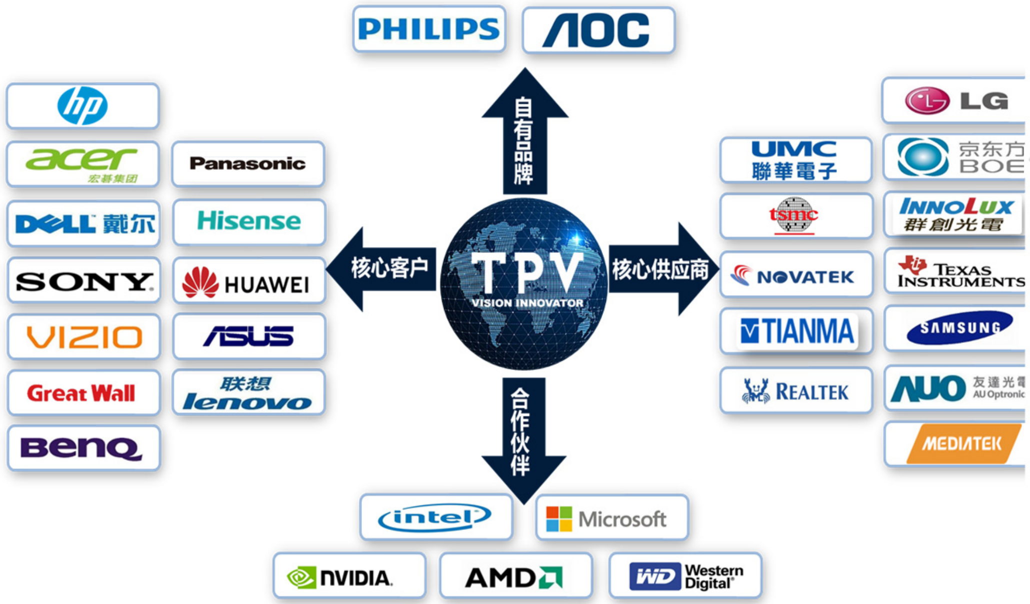
- Printed Circuit Board **(PCB): Mother of Electronics**
- **China** has **50%** Market Share, **21 billion US dollars** in 2025
- PCB Assembly is the **Bottleneck** of Electronics Production


Surface Mount Technology **(SMT)** Machine

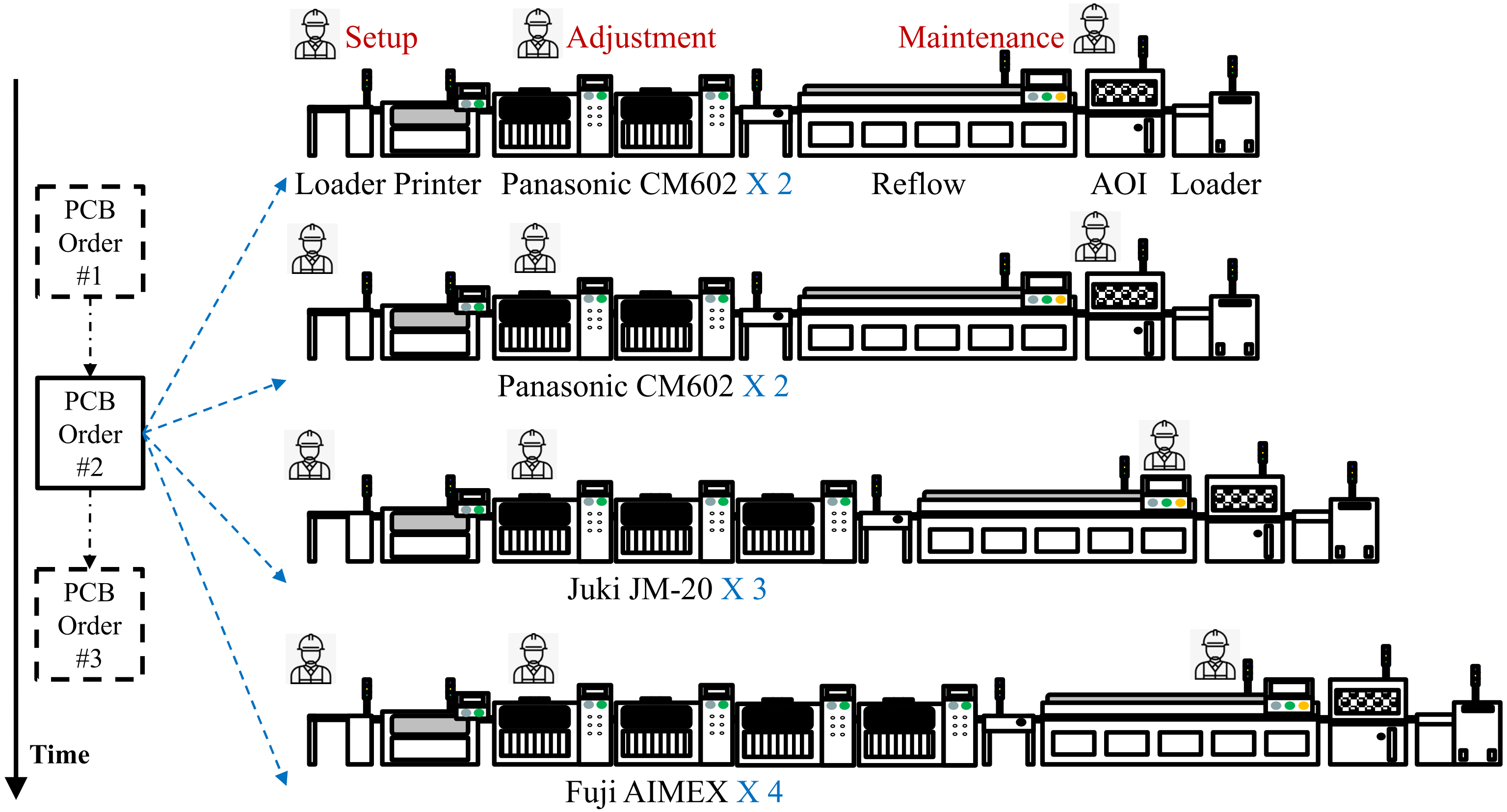
1.2 Practice Case

- **TPV: Global leading monitor manufacturer**
- **One billion US dollor** sales in 2025 (1.16%↑)
- **Fuqing** factory (**45** lines, **2000+** PCB types)
- **Industrial 4.0 (Scheduling for bottleneck)**

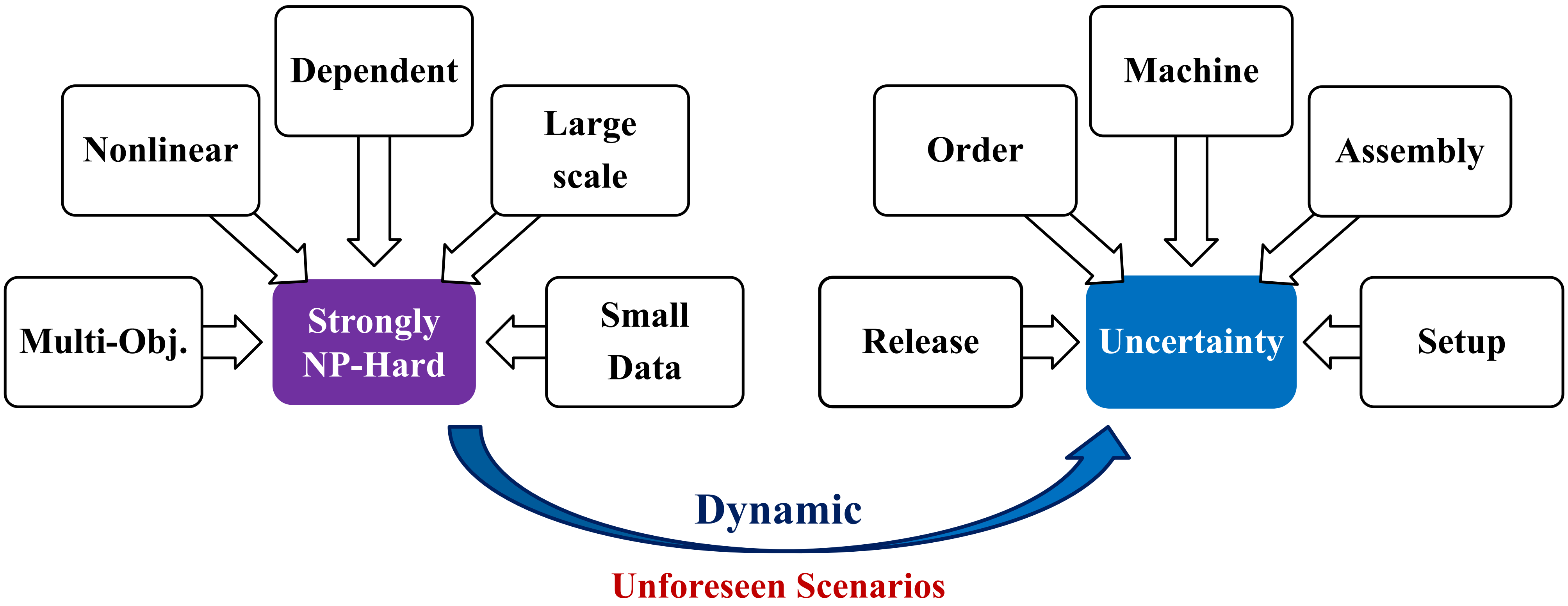


1.3 PCB Assembly Scheduling

- ➢ **Hybrid Flowshop**
- ➢ **Order Mix**
  - ❑ Normal Order
  - ❑ Pilot RD Order
  - ❑ **Customer Service**
- ➢ **Process Types**
  - ❑ Solder
  - ❑ Dispensing

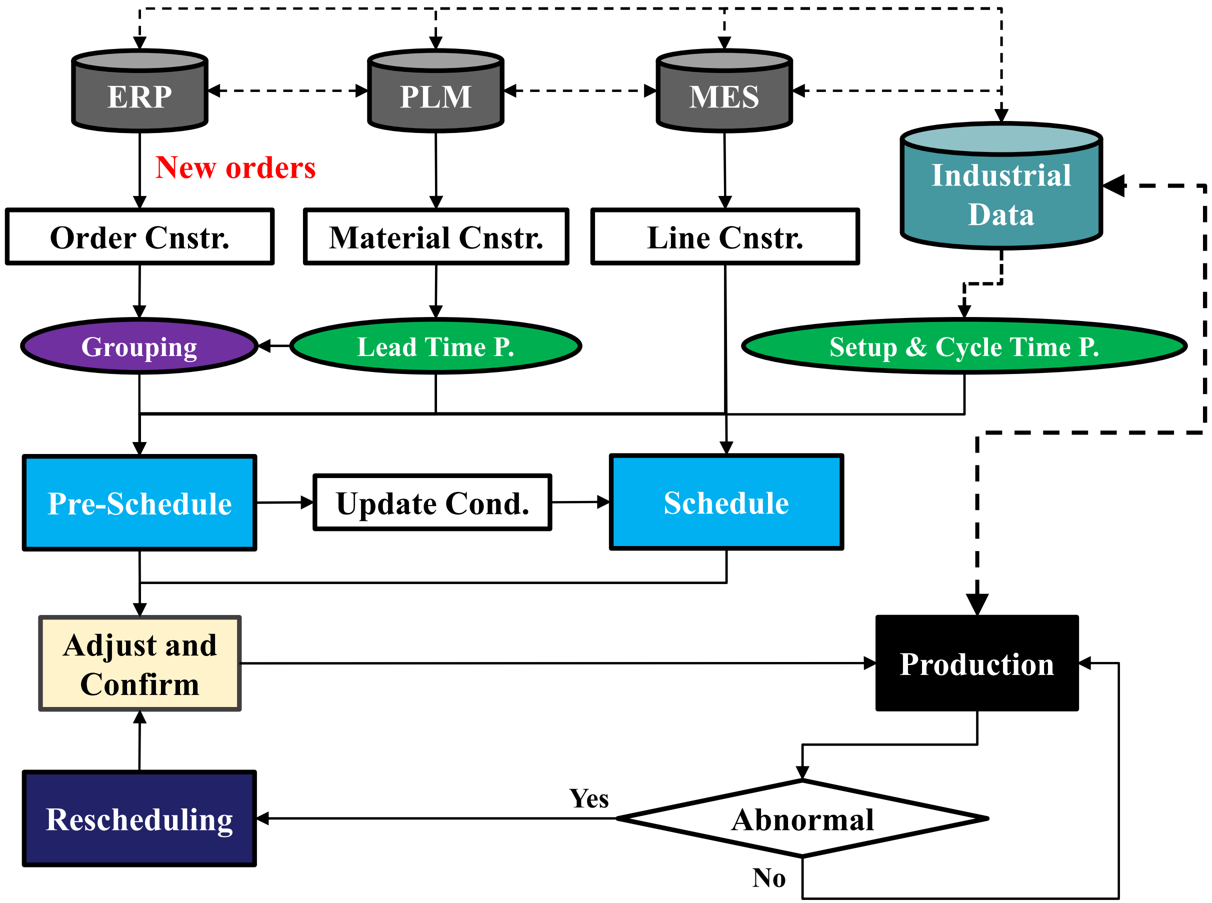


1.4 Optimization Challenges



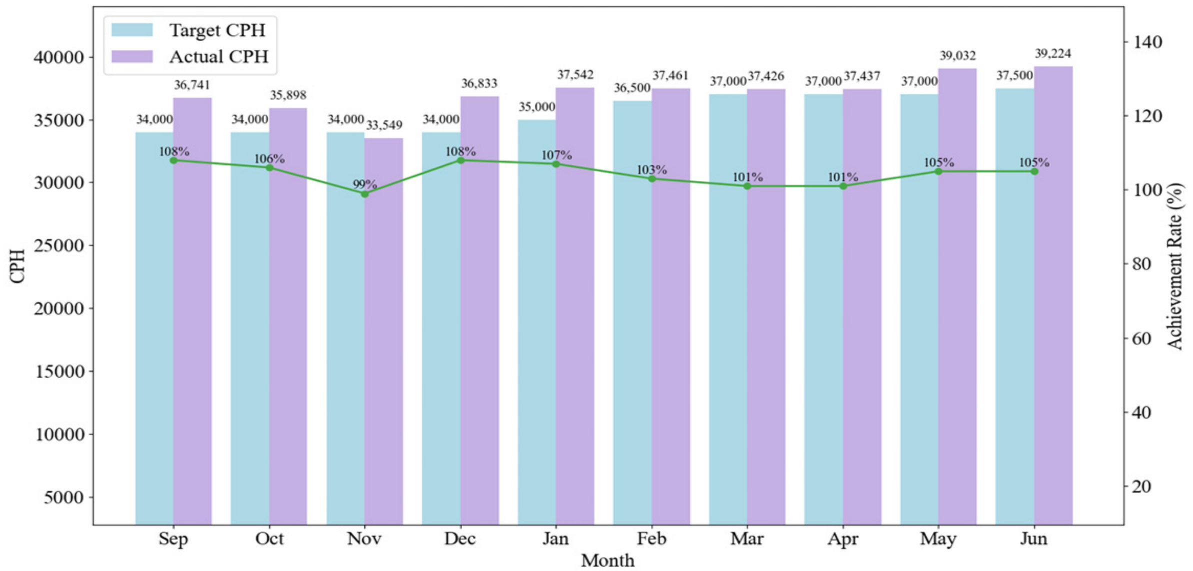
1.5 Industry Practice – System Design

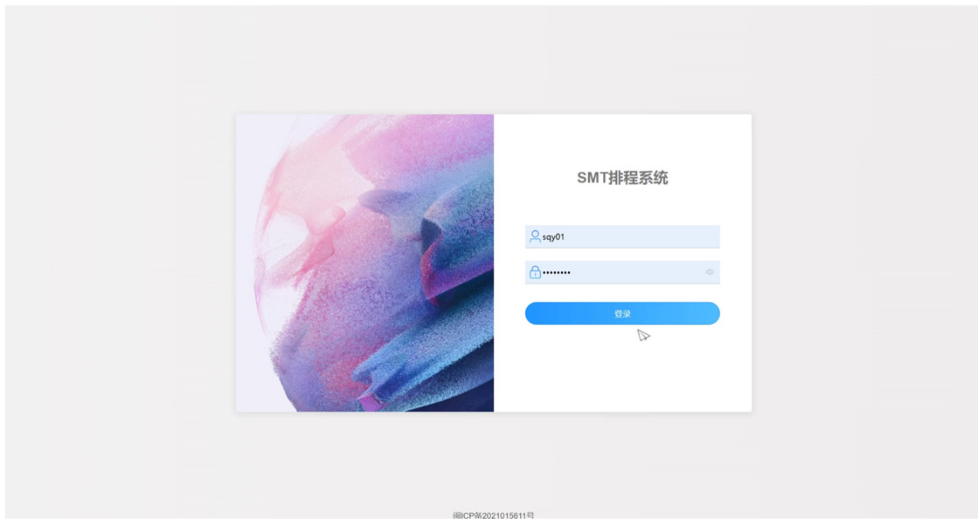
- Prediction
- Grouping
- Scheduling
- Rescheduling



1.6 Industry Implementation

- Implemented at TPV’s **Fuqing** factory since 2022
- Subsequently deployed in **Xiamen, Wuhan**, and **Thailand**
- Save **1.65 million US dollars annually**
- **Throughput gain:** Target CPH +10.3%; Actual CPH +6.8%.
- **Achievement rate** maintained at 99%–108%



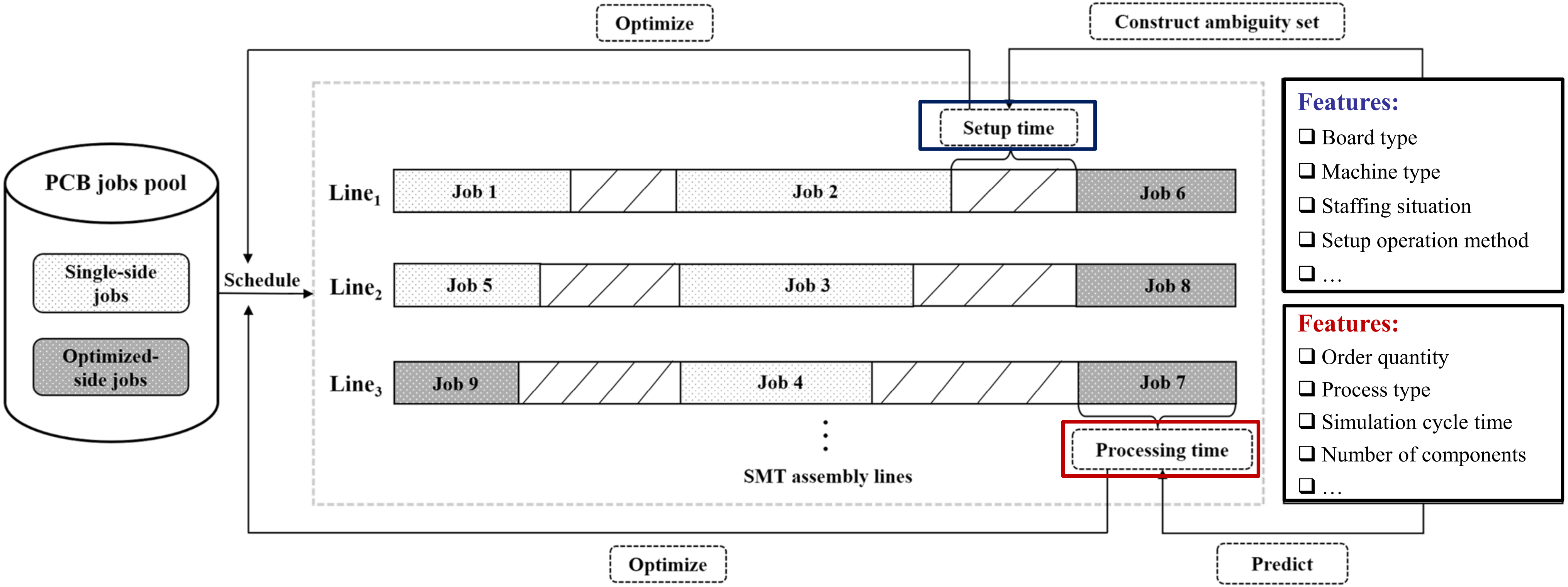


PCB assembly scheduling system Demonstration

2.1 Feature-driven scheduling Problem

Problem: Identical parallel machine scheduling problem (PMSP) for single/optimized double-sided PCBs.

Objectives: Minimize total completion time and makespan.



❖ Li, D., Ding, R., Liu, F., Wang R., Zhong, Y., "Feature-driven robust stochastic scheduling for printed circuit board assembly", *European Journal of Operational Research*, published online, 2026.

2.2 Literature Review

Table 1
Summary and comparison of relevant literature.

Ref.	Machine Environment[a]	Uncertainty Factor[b]		Uncertainty Set[c]	Objectives[d]	Feature-Driven
		p_j	t_j			
Daniels and Kouvelis (1995)	1	✓	×	US	TFT	×
Lu et al. (2012)	1	✓	×	US	TFT	×
Chang et al. (2017)	1	✓	×	AS	TFT	×
Zhang et al. (2018)	1	✓	×	AS	TFT	×
Pei et al. (2022)	1	✓	×	AS	TFT	×
Lu and Pei (2023)	1	✓	×	AS	TFT	×
Xu et al. (2013)	*Pm*	✓	×	US	Makespan	×
Chang et al. (2019)	*Pm*	✓	×	AS	TFT	×
Bruni et al. (2020)	*Pm*	✓	✓	AS	TCT	×
Wang et al. (2023)	*Pm*	✓	×	FDC-AS	TT	✓
Wang et al. (2024)	*Pm*	✓	×	AS	TC	×
Liu et al. (2021)	*Rm*	✓	×	AS	TC	✓
Yanıkoğlu and Yavuz (2022)	*Rm*	✓	×	US	TT	×
Wang et al. (2022a)	*Rm*	✓	×	US	Makespan	×
Our work	*Pm*	✓	✓	FDE-AS	TCT&Makespan	✓

[a] 1: single machine; *Pm*: identical parallel machine; *Rm*: unrelated parallel machine.
[b] p_j: processing time; t_j: setup time.
[c] US: uncertainty set; AS: ambiguity set; FDC-AS: feature-driven cluster-wise ambiguity set; FDE-AS: feature-driven event-wise ambiguity set.
[d] TFT: total flow time; TC: total cost; TWT: total waiting time; TCT: total completion time; TT: total tardiness.

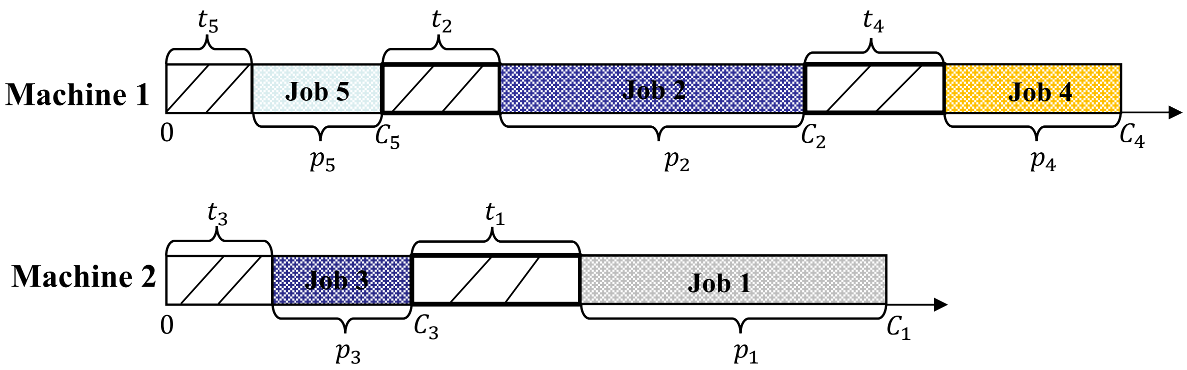
3. Feature-driven Scheduling Model

The notations and descriptions.

Notation	Definition
Indices:	
j	The index for job, $j \in \mathcal{J} = \{1, 2, \dots, J\}$
m	The index for machine, $m \in \mathcal{M} = \{1, 2, \dots, M\}$
n	The index for historical sample point, $n \in \mathcal{N} = \{1, 2, \dots, N\}$
k	The index for sequence position, $k \in \mathcal{J} = \{1, 2, \dots, J\}$
s	The index for scenario, $s \in \mathcal{S} = \{1, 2, \dots, S\}$
Parameters:	
J	The number of jobs
M	The number of machines
N	The number of historical data points
S	The number of scenarios
ζ	The relaxation ratio of $C_{\max}$ obtained by the longest processing time (LPT) rule
r_j	The release time of job j
$\tilde{t}_j$	The random setup time of job j
$\tilde{p}_j$	The random processing time of job j
$\boldsymbol{y}_j$	The feature vector of job j, $\boldsymbol{y}_j \in \mathbb{R}^4$
Variables:	
x_{jmk}	A 0–1 variable, $x_{jmk} = 1$ if job j is processed on machine m with the position being kth to the last, and $x_{jmk} = 0$ otherwise
C_j	The completion time of job j

position to the last

Example: 2 machines and 5 jobs with $r_j = 0$



- **A feasible solution:**

$$(\boldsymbol{x}) \quad \boldsymbol{x}(m=1) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}_{j \times k}$$

$$C_5 = p_5 + t_5, C_2 = C_5 + p_2 + t_2, C_4 = C_2 + p_4 + t_4.$$

$$C_5 + C_2 + C_4 = 3(p_5 + t_5) + 2(p_2 + t_2) + (p_4 + t_4)$$

$$\boldsymbol{x}(m=2) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{j \times k}$$

$$C_3 = p_3 + t_3, C_1 = C_3 + p_1 + t_1$$

$$C_3 + C_1 = 2(p_3 + t_3) + (p_1 + t_1)$$

Chang, Z., Ding, J. Y., & Song, S. (2019). Distributionally robust scheduling on parallel machines under moment uncertainty. *European Journal of Operational Research*, *272*(3), 832-846.

3.1 Feature-driven Uncertainty Modeling

❑ Two Uncertainties Parameters

Processing Time $\tilde{p}$: Feature-Driven Prediction

PCB features ▸ **Linear SVR Model** ▸ **Prediction**

- **Feature selection**

Quantity, Component Type, Simulated CT, Process Type

- **Prediction**

Keep the model linear and solvable

Predictive Model

$$\tilde{p}_j = g(\boldsymbol{y}_j) = \boldsymbol{\beta}^\top \boldsymbol{q}(\boldsymbol{y}_j)$$

Setup Time $\tilde{t}$: Event-Wise Ambiguity Set

Setup Scenario **Clustering** **Ambiguity Set**

- **Feature selection**

Component Type, Machine Type, Process Type

- **Set up scenario clustering**

K-Means, K-Means++

Event-Wise Ambiguity Set

$$\mathcal{F} = \left\{ \mathbb{P} \in \mathcal{P}_0(\mathbb{R}^J \times \mathcal{S}) \;\middle|\; \begin{array}{ll} (\tilde{\boldsymbol{t}}, \tilde{s}) \sim \mathbb{P} & \\ \mathbb{E}_{\mathbb{P}}[\tilde{\boldsymbol{t}} \mid \tilde{s} = s] = \boldsymbol{\mu}_s & \forall s \in \mathcal{S} \\ \mathbb{E}_{\mathbb{P}}[|\tilde{\boldsymbol{t}} - \boldsymbol{\mu}_s| \mid \tilde{s} = s] \leq \boldsymbol{\sigma}_s & \forall s \in \mathcal{S} \\ \mathbb{P}[\tilde{\boldsymbol{t}} \in \mathcal{Z}_s \mid \tilde{s} = s] = 1 & \forall s \in \mathcal{S} \\ \mathbb{P}[\tilde{s} = s] = w_s & \forall s \in \mathcal{S} \end{array} \right\}.$$

Chen, Z., Sim, M., & Xiong, P. (2020). Robust stochastic optimization made easy with RSOME. *Management Science*, *66*(8), 3329-3339.

3.2 Feature-driven Robust Stochastic Optimization Model

- **Trade-off between timely delivery and efficiency**

$$[\text{FDRSO}] \min_{x} \sup_{\mathbb{P}\in\mathcal{F}} \mathbb{E}_{\mathbb{P}}\left[\sum_{m\in\mathcal{M}}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{J}} k(\tilde{p}_j+\tilde{t}_j)x_{jmk}\right] \quad (1)$$

Total flow time

$$\text{s.t.} \sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{J}} x_{jmk} = 1, \quad \forall j\in\mathcal{J}, \quad (2)$$

$$\sum_{j\in\mathcal{J}} x_{jmk} \le 1, \quad \forall m\in\mathcal{M}, k\in\mathcal{J}, \quad (3)$$

$$\sum_{j\in\mathcal{J}} x_{jmk} \ge \sum_{j\in\mathcal{J}} x_{jm(k+1)}, \quad \forall m\in\mathcal{M}, k\in\mathcal{J}\setminus\{1\}, \quad (4)$$

$$\sup_{\mathbb{P}\in\mathcal{F}} \mathbb{E}_{\mathbb{P}}\left[\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{J}} (\tilde{p}_j+\tilde{t}_j)x_{jmk}\right] \le \zeta C_{\max}(LPT), \quad \forall m\in\mathcal{M}, \quad (5)$$

Makespan: ϵ-constraint

$$\tilde{p}_j = g(\mathbf{y}_j), \quad \forall j\in\mathcal{J}, \quad (6)$$

$$x_{jmk}\in\{0,1\}, \quad \forall j\in\mathcal{J}, m\in\mathcal{M}, k\in\mathcal{J}. \quad (7)$$

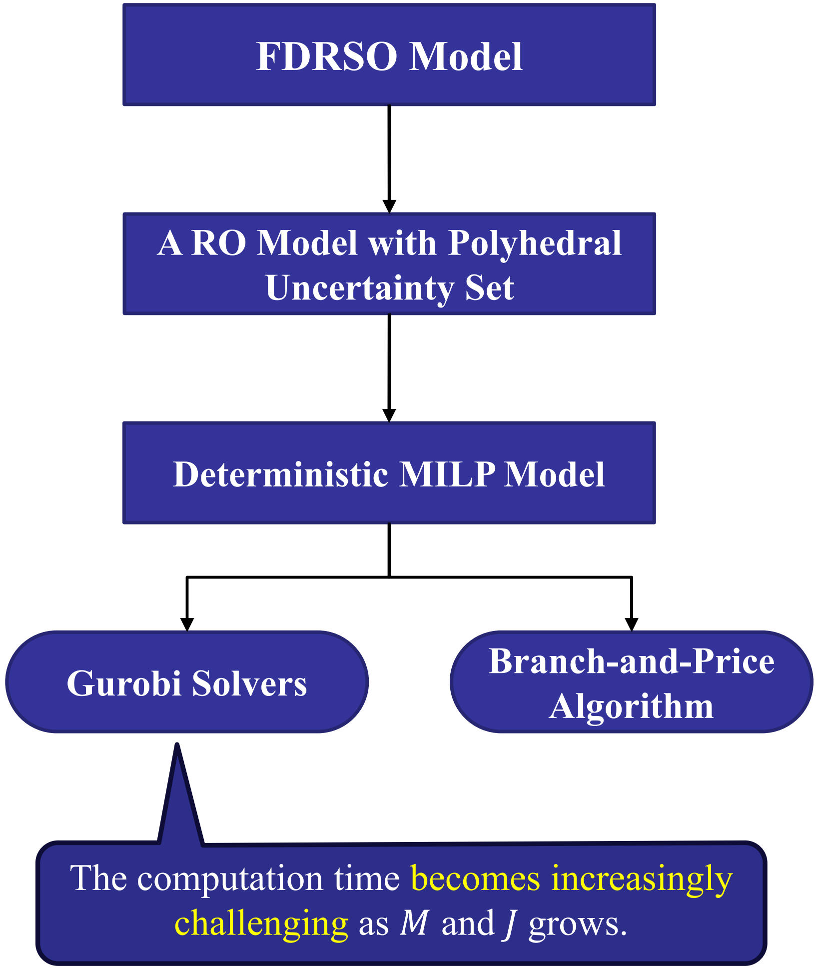
- Objective (1) minimizes the expected total flow time under the worst-case distribution in $\mathcal{F}$.
- Constraints (2)–(4) ensure a feasible schedule.
- Constraint (5) caps the worst-case makespan over $\mathcal{F}$ by $\zeta C_{max}(LPT)$.
- Constraint (6) links processing time to job features via a linear prediction model.
- Constraint (7) enforces binary scheduling decisions.

3.3 Robust Counterpart Model

Proposition 1

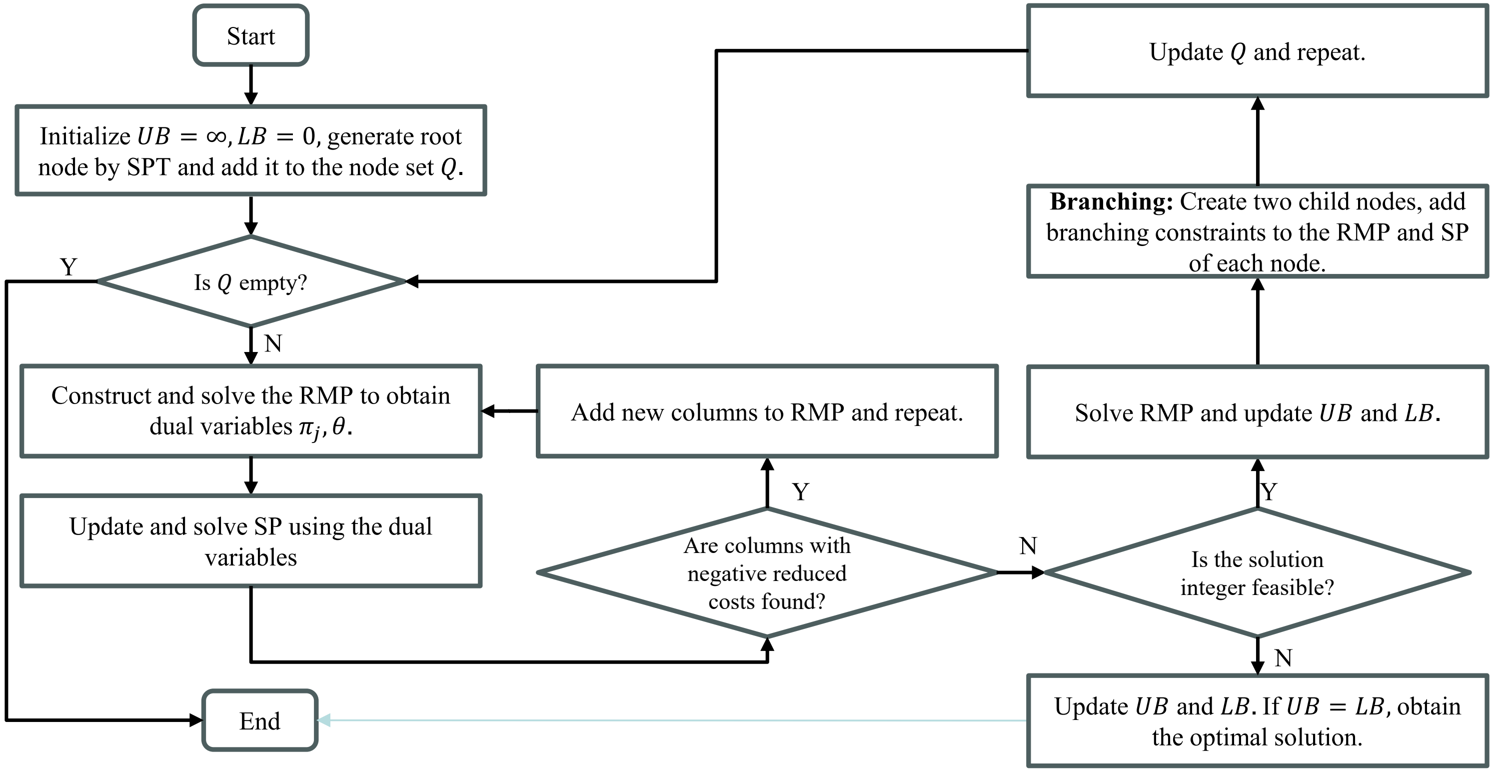
The FDRSO model can be transformed into a MILP problem.

$$
\begin{aligned}
\inf\ & \sum_{s\in\mathcal{S}}\left(\alpha_s + \sum_{j\in\mathcal{J}}\beta_j^s\mu_j^s + \sum_{j\in\mathcal{J}}\gamma_j^s\sigma_j^s\right) && \\
\text{s.t.}\ & \sum_{r\in\mathcal{S}}\left[\rho_{rs}\sum_{j\in\mathcal{J}}\left((\mu_j^r)^2-(\mu_j^s)^2\right)\right] && \\
& + \sum_{j\in\mathcal{J}}\lambda_j^s\mu_j^s - \sum_{j\in\mathcal{J}}\lambda_j'^s\mu_j^s + \sum_{j\in\mathcal{J}}\eta_j\bar{t}_j - \sum_{j\in\mathcal{J}}\underline{t}_j\eta_j' \le \alpha_s - w_s\sum_{m\in\mathcal{M}}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{J}}k\tilde{p}_jx_{jmk}, && \forall s\in\mathcal{S}, \\
& \sum_{r\in\mathcal{S}}\left[\rho_{rs}\sum_{j\in\mathcal{J}}\left((\mu_j^r)^2-(\mu_j^s)^2\right)\right] + \sum_{j\in\mathcal{J}}\eta_j\bar{t}_j - \sum_{j\in\mathcal{J}}\eta_j'\underline{t}_j && \\
& \le \zeta C_{\max}(LPT) - \sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{J}}\tilde{p}_jx_{jmk}, && \forall m\in\mathcal{M}, s\in\mathcal{S}, \\
& 2\rho_{rs}(\mu_j^r-\mu_j^s) + \lambda_j^s - \lambda_j'^s + \eta_j - \eta_j' && \\
& \ge w_s\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{J}}kx_{jmk} - \beta_j^s - \gamma_j^s, && \forall j\in\mathcal{J}, r\in\mathcal{S}, s\in\mathcal{S}, \\
& 2\rho_{rs}(\mu_j^r-\mu_j^s) + \eta_j - \eta_j' \ge \sum_{k\in\mathcal{J}}x_{jmk}, && \forall j\in\mathcal{J}, m\in\mathcal{M}, r\in\mathcal{S}, s\in\mathcal{S}, \quad (21) \\
& \lambda_j^s + \lambda_j'^s \ge \gamma_j^s, && \forall j\in\mathcal{J}, s\in\mathcal{S}, \\
& \sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{J}}x_{jmk} = 1, && \forall j\in\mathcal{J}, \\
& \sum_{j\in\mathcal{J}}x_{jmk} \le 1, && \forall m\in\mathcal{M}, k\in\mathcal{J}, \\
& \sum_{j\in\mathcal{J}}x_{jmk} \ge \sum_{j\in\mathcal{J}}x_{jm(k+1)}, && \forall m\in\mathcal{M}, k\in\mathcal{J}\setminus\{1\}, \\
& \tilde{p}_j = g(\mathbf{y}_j), && \forall j\in\mathcal{J}, \\
& x_{jmk}\in\{0,1\}, && \forall m\in\mathcal{M}, j\in\mathcal{J}, k\in\mathcal{J}, \\
& \rho_{rs}, \gamma_j^s, \lambda_j^s, \lambda_j'^s, \eta_j, \eta_j' \ge 0, && \forall j\in\mathcal{J}, r\in\mathcal{S}, s\in\mathcal{S}, \\
& \alpha_s, \beta_j^s, \gamma_j^s \in\mathbb{R}, && \forall j\in\mathcal{J}.
\end{aligned}
$$



4. Branch-and-Price Algorithm

The B&P algorithm integrates column generation (C&G) and branch-and-bound (B&B) to effectively solve large-scale integer programming problems.



- C&G solves the linear relaxation of the problem.
- B&B enforces integrality constraints to achieve global optimality.
- A depth-first search strategy is employed in B&B tree.

4.1 B&P: Master Problem

Dantzig-Wolfe decomposition: Model (21) is reformulated into a master problem (MP) and M independent subproblems (SP).

$$[\text{MP}]\ \min \sum_{l \in \Psi} \epsilon_l \phi_l \qquad (22)$$

The expected total completion time of partial schedule l

$$\text{s.t.} \sum_{l \in \Psi} a_{jl} \phi_l = 1, \qquad \forall j \in \mathcal{J}, \qquad (23)$$

$$\sum_{l \in \Psi} \phi_l \leq M, \qquad (24)$$

$$\phi_l \in \{0, 1\}, \qquad \forall l \in \Psi. \qquad (25)$$

- (22) minimize the total completion time of all selected partial schedules.
- (23) ensures that each job is assigned to only one machine.
- (24) restricts the number of selected partial schedules to at most M.
- MP contains exponentially many decision variables (columns) and involves only $J + 1$ constraints.

- Treat a machine schedule as a column/ partial schedule $l \in \Psi$.
- $\phi_l = 1$ if l is selected in the final solution, $\phi_l = 0$ otherwise.
- $a_{jl} = 1$ if job j is covered by l for every $j \in \mathcal{J}$ and $l \in \Psi$, $a_{jl} = 0$ otherwise.

4.2 B&P: Subproblems

$$
\begin{aligned}
[SP]\ \min\ & \boxed{\epsilon - \sum_{j\in\mathcal{J}} a_j \pi_j - \theta} \quad \text{the reduced cost of a new column} \\
\text{s.t.}\ & \sum_{k\in\mathcal{J}} X_{jk} \le 1, && \forall j \in \mathcal{J}, \\
& \sum_{j\in\mathcal{J}} X_{jk} \le 1, && \forall k \in \mathcal{J}, \\
& \sum_{j\in\mathcal{J}} X_{jk} \ge \sum_{j\in\mathcal{J}} X_{j(k+1)}, && \forall k \in \mathcal{J} \setminus \{1\}, \\
& a_j = \sum_{k\in\mathcal{J}} X_{jk}, && \forall j \in \mathcal{J}, \\
& \tilde{p}_j = g(\mathbf{y}_j), && \forall j \in \mathcal{J}, \\
& \epsilon \le \sum_{s\in\mathcal{S}} \left(\alpha_s + \sum_{j\in\mathcal{J}} \beta_j^s \mu_j^s + \sum_{j\in\mathcal{J}} \gamma_j^s \sigma_j^s \right), \\
& \sum_{r\in\mathcal{S}} \left[\rho_{rs} \sum_{j\in\mathcal{J}} \left((\mu_j^r)^2 - (\mu_j^s)^2 \right) \right] + \sum_{j\in\mathcal{J}} \lambda_j^s \mu_j^s - \sum_{j\in\mathcal{J}} \lambda_j'^s \mu_j^s + \sum_{j\in\mathcal{J}} \eta_j \bar{t}_j - \sum_{j\in\mathcal{J}} \eta_j' \underline{t}_j \\
& \le \alpha_s - w_s \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{J}} k \tilde{p}_j X_{jk}, && \forall s \in \mathcal{S}, \\
& \sum_{r\in\mathcal{S}} \left[\rho_{rs} \sum_{j\in\mathcal{J}} \left((\mu_j^r)^2 - (\mu_j^s)^2 \right) \right] + \sum_{j\in\mathcal{J}} \eta_j \bar{t}_j - \sum_{j\in\mathcal{J}} \eta_j' \underline{t}_j \le \zeta C_{\max}(LPT) - \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{J}} \tilde{p}_j X_{jk}, && \forall s \in \mathcal{S}, \\
& 2\rho_{rs}(\mu_j^r - \mu_j^s) + \lambda_j^s - \lambda_j'^s + \eta_j - \eta_j' \ge w_s \sum_{k\in\mathcal{J}} k X_{jk} - \beta_j^s - \gamma_j^s, && \forall j \in \mathcal{J}, r \in \mathcal{S}, s \in \mathcal{S}, \\
& 2\rho_{rs}(\mu_j^r - \mu_j^s) + \eta_j - \eta_j' \ge \sum_{k\in\mathcal{J}} X_{jk}, && \forall j \in \mathcal{J}, r \in \mathcal{S}, s \in \mathcal{S}, \\
& \lambda_j^s + \lambda_j'^s \ge \gamma_j^s, && \forall j \in \mathcal{J}, s \in \mathcal{S}, \\
& X_{jk} \in \{0, 1\}, && \forall j \in \mathcal{J}, k \in \mathcal{J}, \\
& \rho_{rs}, \gamma_j^s, \lambda_j^s, \lambda_j'^s, \eta_j, \eta_j' \ge 0, && \forall j \in \mathcal{J}, r \in \mathcal{S}, \in \mathcal{S}, \\
& \alpha_s, \beta_j^s, \gamma_j^s \in \mathbb{R}, && \forall j \in \mathcal{J}, \forall s \in \mathcal{S}.
\end{aligned}
$$

- SP leverages dual information to compute the reduced cost of candidate columns not currently included in RMPLR.
- Single-column pricing strategy in the C&G process.
- It finds the column with the minimum reduced cost and adds it to the RMP.
- X_{jk} be a binary variable indicating whether job j is scheduled in the k-th position to the last.

5.1 Computational Results and Analysis

- **Data source:** Real-world SMT production dataset from a leading electronics manufacturer (Fuzhou, China), including **24,648** records over **12 months.**
- 40 instances generated across line types (large/medium/small) with multiple (M,J) combinations.
- Setup scenarios derived from clustering.

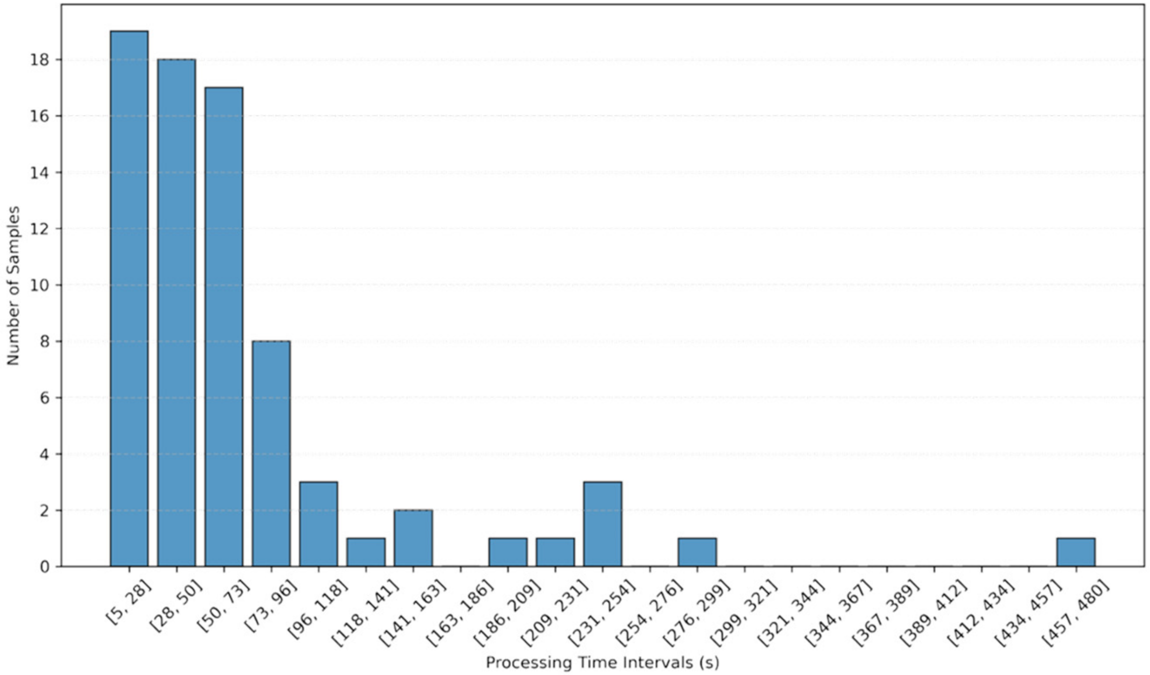
Partial production data information for PCB orders

Board number	Machine type	Line	Product Type	Order quantity	Completed Quantity	Number of boards	Number of components	Simulation CT	Average CT	Process type	Setup time	Total number of components	Line type	Process type_Double	Process type_Single
715***01	SMT***QU	SM22A	HLCD	1063	1063	3	348	35	60.85	T	20	1044	3	TRUE	FALSE
715***01	SMT***Q7	SM22A	HLCD	1709	918	3	289	30	13.47	T	20	867	3	TRUE	FALSE
715***01	SMT***Q7	SM22B	HLCD	625	625	3	289	40	40.19	T	20	867	3	TRUE	FALSE
715***85	SMT***Q1	SM22B	HLCD	1015	0	3	421	45	52.74	T	40	1263	3	TRUE	FALSE
715***63	SMT***Q2	SM01A	PD	334	334	2	709	52	75.45	T	20	1418	1	TRUE	FALSE
715***63	SMT***Q3	SM01A	PD	30	30	2	815	54	280	T	20	1630	1	TRUE	FALSE
715***63	SMT***X2	SM01A	PD	96	96	2	751	50	75	T	20	1502	1	TRUE	FALSE
715***70	SMT***Q3	SM02A	PD	24	24	1	1543	65	125	T	40	1543	1	TRUE	FALSE
715***37	SMT***Q3	SM02A	SMART	656	265	1	1230	65	44.63	T	20	1230	1	TRUE	FALSE
715***17	SMT***Q6	SM05A	HLCD	174	174	3	362	56	62.07	T	15	1086	1	TRUE	FALSE
715***17	SMT***Q9	SM05A	HLCD	20	20	3	362	56	85.71	T	15	1086	1	TRUE	FALSE
715***17	SMT***Q9	SM05A	HLCD	88	88	3	362	56	170	T	15	1086	1	TRUE	FALSE
715***17	SMT***Q2	SM05A	HLCD	60	60	3	308	50	165	T	15	924	1	TRUE	FALSE
715***17	SMT***Q6	SM05A	HLCD	24	24	3	309	56	262.5	T	15	927	1	TRUE	FALSE

5.1 Processing Time Empirical Evidence

Table A.1: Statistical information of historical processing times for some representative job types.

Board Number	Machine Type	Sample Size	Mean.CT	Std.CT	Min.CT	Max.CT
71****36	SMT****Q3	18	178.71	133.81	25.00	500.00
71****36	SMT****Q5	8	219.17	140.31	60.00	400.00
71****43	SMT****QR	10	57.27	39.78	7.03	138.00
71****43	SMT****Q3	23	96.24	61.06	12.50	187.50
71****43	SMT****QP	8	230.87	144.83	60.00	450.00
71****43	SMT****QR	29	76.37	87.50	3.60	450.00
71****43	SMT****QV	10	62.01	43.35	14.13	150.00
71****43	SMT****QY	75	73.32	75.99	5.77	480.00
71****01	SMT****Q2	12	182.96	118.91	4.01	342.86
71****49	SMT****Q2	14	105.65	129.84	24.49	400.00
71****80	SMT****Q2	12	203.88	147.22	52.17	450.00
71****80	SMT****Q4	40	134.52	93.59	22.50	337.50
71****80	SMT****Q6	19	139.45	152.31	3.06	500.00
71****80	SMT****Q7	8	124.36	118.87	24.00	400.00
71****80	SMT****Q2	8	105.83	107.67	3.26	360.00
71****94	SMT****Q2	9	143.95	92.00	44.44	333.33
71****93	SMT****Q4	28	114.61	87.55	4.93	300.00
71****93	SMT****QA	31	99.40	105.99	4.60	380.00
71****93	SMT****QB	40	107.87	104.54	7.36	450.00
71****57	SMT****Q4	34	52.43	34.03	6.86	177.27
71****57	SMT****Q5	29	68.44	33.28	7.50	120.00
71****82	SMT****A1	9	89.45	86.09	27.91	300.00
71****82	SMT****A2	58	88.22	91.54	1.07	400.00
71****00	SMT****Q2	22	96.90	119.30	4.80	456.00
71****00	SMT****Q1	12	86.50	60.24	3.62	180.00



- Even for the same PCB on the same machine, real-world variability causes processing times to be highly uncertain.

5.1 Processing Time Prediction

❑ Data cleaning and preprocessing

- The post-cleaning visualizations show a more stable and consistent weak linear trend
- One-hot for categorical features.
- Standardize continuous features.

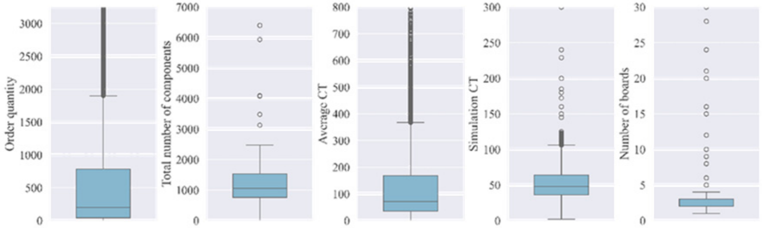
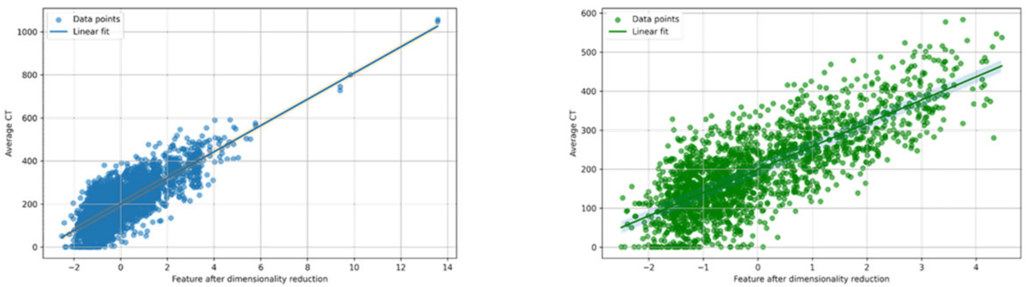


Figure C.2: Box plots of prediction features.



(a) Before removing outliers

(b) After removing outliers

Figure C.3: Linear regression scatter plots of integrated features and average CT.

❑ Feature selection

- Use correlation analysis + mutual information to assess relevance.
- Selected predictors: order quantity, total component count, simulation CT, process type.

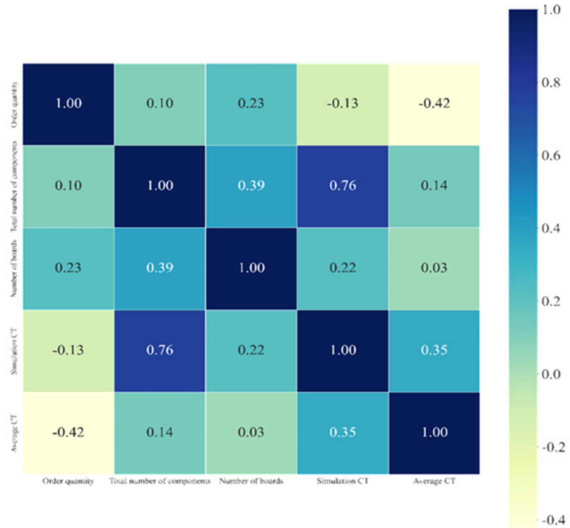


Figure C.4: Heatmap of Pearson correlation coefficients between prediction features and average CT.

5.1 Processing Time Prediction Model Selection

Table 3
Evaluation metrics of six regression models.

linear models are chosen for integration into the MILP/RSO framework

Regression Model	Large Order Lines				Medium Order Lines				Small Order Lines			
	RMSE	MAE	MAPE	R^2	RMSE	MAE	MAPE	R^2	RMSE	MAE	MAPE	R^2
Bayesian Ridge	24.817	18.387	31.057%	0.306	22.191	16.685	28.004%	0.188	30.109	23.857	27.519%	0.112
Linear Regression	24.917	18.444	31.108%	0.300	22.301	16.823	28.303%	0.180	30.113	24.261	27.902%	0.112
Elastic Net	24.805	18.355	31.000%	0.306	22.220	16.710	28.085%	0.186	30.063	23.888	27.554%	0.115
Linear SVR	26.213	**17.755**	**26.899%**	0.225	22.566	**15.572**	**23.531%**	0.160	30.698	24.674	28.282%	0.077
Ridge	24.917	18.444	31.107%	0.300	22.300	16.822	28.302%	0.180	30.111	24.255	27.896%	0.112
Lasso	24.805	18.356	31.002%	0.306	22.219	16.708	28.081%	0.186	30.075	**23.826**	**27.497%**	0.114

The bold values represent the minimum value in each column. The same notation is used in the subsequent tables.

Table 4
FDRSO's optimization results under different prediction models.

Sample #	*M*	*J*	FDRSO-linear SVR		FDRSO-Lasso		FDRSO-Ridge		Objective Improvement	
			Time (s)	Objective	Time (s)	Objective	Time (s)	Objective	Lasso	Ridge
1	3	5	0.30	515.20*	0.25	525.60*	0.25	525.90*	2%	2%
2	3	6	0.31	652.55*	0.32	666.74*	0.31	667.09*	2%	2%
3	3	7	0.45	814.76*	0.43	833.69*	0.40	834.11*	2%	2%
4	3	8	0.92	1006.50*	0.88	1031.40*	0.79	1032.00*	2%	2%
5	3	9	1.20	1102.40*	1.16	1130.10*	1.19	1130.70*	2%	2%
6	3	10	2.09	1253.00*	1.89	1285.90*	1.98	1286.60*	3%	3%
7	3	12	3.54	1410.20*	3.18	1472.40*	3.26	1473.10*	4%	4%
8	3	15	5.31	2012.40*	5.47	2099.00*	4.90	2100.00*	4%	4%
9	3	18	7.80	2694.10*	7.60	2819.30*	7.66	2820.70*	4%	4%
10	3	20	10.02	3110.80*	10.29	3315.00*	10.75	3316.70*	6%	6%
11	4	10	3.08	1064.50*	2.68	1090.90*	3.64	1091.50*	2%	2%
12	4	15	12.40	1710.10*	11.39	1778.70*	11.65	1779.50*	4%	4%
13	4	20	16.86	2578.20*	13.67	2730.30*	15.49	2731.60*	6%	6%
14	4	25	70.79	3557.68*	51.45	3807.26*	66.99	3809.07*	7%	7%
15	4	30	74.33	4750.60*	53.68	5075.30*	63.50	5077.70*	6%	6%

The values marked with the symbol * indicate optimal solutions, and the values marked with the symbol - indicate that no feasible solution was found. The same notation is used in the subsequent tables.

- Linear SVR achieves the best MAE/MAPE on medium- and large-order lines.
- When embedded into FDRSO (compared with Lasso/Ridge), Linear SVR yields consistently better objective values, with a 2–7% improvement, while runtime differences are minor.

5.2 Analysis of Setup Time Scenarios

❑ Ambiguity set for setup time

$$\mathcal{F} = \left\{ \mathbb{P} \in \mathcal{P}_0(\mathbb{R}^J \times \mathcal{S}) \;\middle|\; \begin{array}{ll} (\tilde{\boldsymbol{t}}, \tilde{\boldsymbol{s}}) \sim \mathbb{P} & \\ \mathbb{E}_{\mathbb{P}}[\tilde{\boldsymbol{t}} \mid \tilde{\boldsymbol{s}} = s] = \boldsymbol{\mu}_s & \forall s \in \mathcal{S} \\ \mathbb{E}_{\mathbb{P}}[|\tilde{\boldsymbol{t}} - \boldsymbol{\mu}_s| \mid \tilde{\boldsymbol{s}} = s] \leq \boldsymbol{\sigma}_s & \forall s \in \mathcal{S} \\ \mathbb{P}[\tilde{\boldsymbol{t}} \in \mathcal{Z}_s \mid \tilde{\boldsymbol{s}} = s] = 1 & \forall s \in \mathcal{S} \\ \mathbb{P}[\tilde{\boldsymbol{s}} = s] = w_s & \forall s \in \mathcal{S} \end{array} \right\}.$$

How to set the number of scenarios?

✓ **Use clustering to differentiate setup-time scenarios and build an event-wise ambiguity set.**

❑ Feature selection for clustering

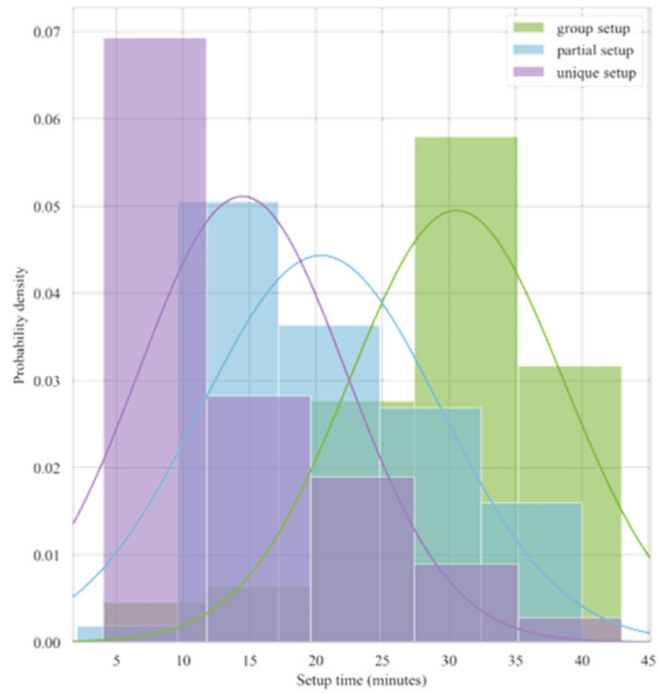


Fig. 2. Empirical distributions of setup time across three setup scenarios.

5.2 Analysis of Setup Time Scenarios

❑ Feature selection for clustering

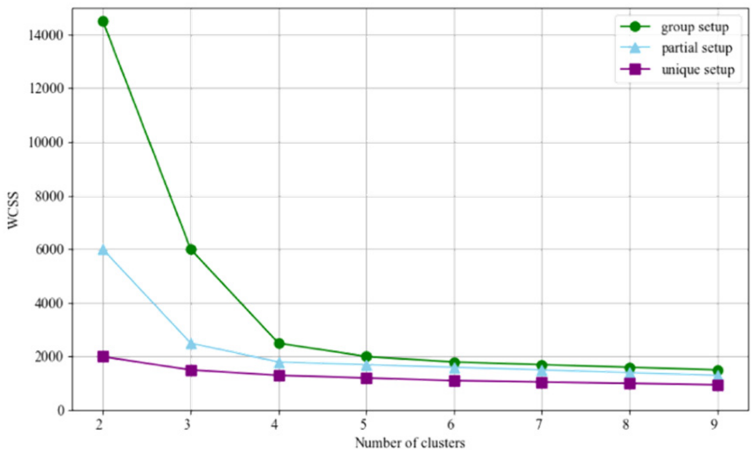


Fig. 3. Elbow plot of WCSS against different values of k.

- $K = 3$ enables the construction of a representative and computationally tractable event-wise ambiguity set.

❑ Comparison of clustering method

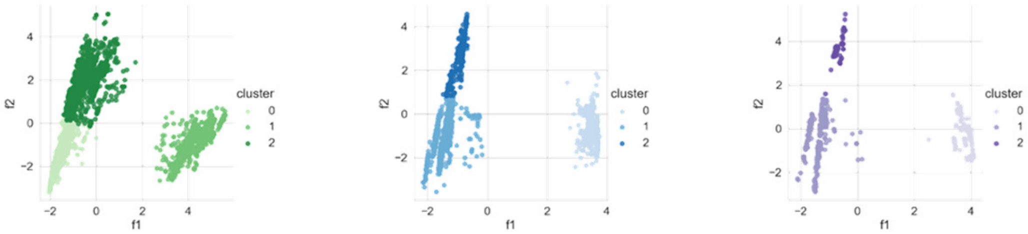
Table 5
SSE and CH Index of four clustering algorithms across three setup scenarios.

Setup Scenario	Metric	K-means	K-means++	Mini-Batch K-means	Bi-KMeans
Unique Setup	SSE	**2892.31**	**2892.31**	2892.53	2999.8
	CH	**6744.05**	**6744.05**	6743.63	6422.6
Group Setup	SSE	**1124.00**	**1124.00**	1124.09	1262.07
	CH	**7355.06**	**7355.06**	7355.06	6444.24
Partial Setup	SSE	**390.93**	**390.93**	390.94	390.93
	CH	**2741.16**	**2741.16**	2741.16	2741.16

❖ Calinski-Harabasz (CH) index

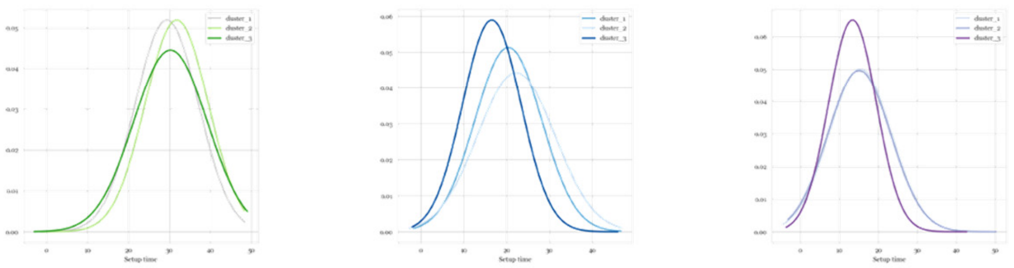
- K-means / K-means++ consistently best (lowest SSE & highest CH).
- Clustering method: K-means (simplicity + stability).

5.2 Analysis of Setup Time Scenarios



(a) Group setup (b) Partial setup (c) Unique setup

Fig. 4. Clustering results for the setup-related feature vectors under the group, partial, and unique setup scenarios.



(a) Group setup (b) Partial setup (c) Unique setup

Fig. 5. Kernel density estimation of setup time distributions corresponding to each cluster under the group, partial, and unique setup scenarios.

Table 6
Optimization results of the FDRSO model with different numbers of setup scenarios.

Sample #	M	J	$S=1$		$S=3$		$S=9$		Objective Improvement	
			Time (s)	Objective	Time (s)	Objective	Time (s)	Objective	$S=1$	$S=3$
16	3	5	0.01	689.42*	0.04	523.12*	0.19	469.48*	32%	10%
17	3	6	0.01	914.53*	0.04	680.00*	0.23	543.54*	41%	20%
18	3	7	0.02	1048.84*	0.07	860.60*	0.32	649.68*	38%	25%
19	3	8	0.02	1422.30*	0.09	1225.21*	0.55	923.76*	35%	25%
20	3	9	0.03	1504.43*	0.12	1235.22*	0.85	987.85*	34%	20%
21	3	10	0.03	1632.72*	0.13	1368.44*	1.34	1225.31*	25%	11%
22	3	12	0.05	1846.73*	0.18	1415.83*	2.23	1400.92*	24%	1%
23	3	15	0.08	2542.04*	0.30	2068.00*	3.27	1998.72*	21%	3%
24	3	18	0.12	3677.24*	0.38	2939.70*	4.89	2808.62*	24%	5%
25	3	20	0.14	4030.96*	0.47	3271.64*	6.40	3046.13*	24%	7%
26	4	10	0.04	1394.20*	0.15	1086.34*	2.02	1041.96*	25%	4%
27	4	15	0.11	2143.52*	0.33	1835.31*	6.17	1741.14*	19%	5%
28	4	20	0.20	3347.14*	0.63	2786.52*	11.47	2667.24*	20%	4%
29	4	25	0.25	4476.13*	0.99	3810.31*	40.15	3741.80*	16%	2%
30	4	30	0.41	5837.20*	1.40	4983.62*	44.24	4463.33*	24%	10%

- Selecting the number of setup scenarios trade-off between solution accuracy and computational efficiency.
- S=9 is suitable for smaller plants with acceptable computation.
- S=3 offers a near-optimal and practical choice for real-world PCB assembly scheduling.

5.3 Algorithmic Performance Comparison

- Medium-to-large instances, 20 test instances, each solved by B&P and Gurobi
- 50 repetitions per instance, report average objective value and runtime.

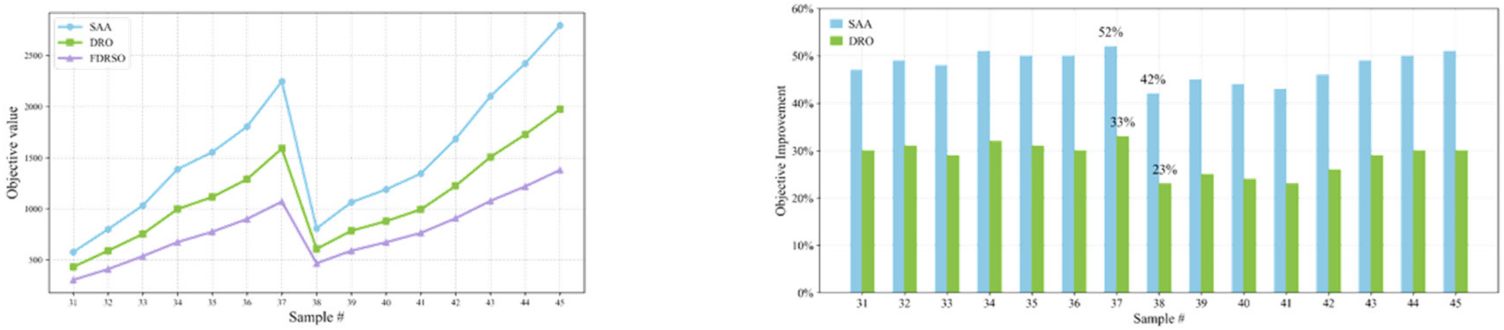
Table 7
Comparison between the B&P algorithm and GUROBI for the FDRSO model.

Sample #	M	J	GUROBI		B&P		Time Improvement
			Time (s)	Objective	Time (s)	Objective	
46	4	10	6.40	920.00*	11.20	920.00*	−75.0%
47	4	15	36.50	1491.00*	62.14	1491.00*	−70.2%
48	4	20	31.80	2395.90*	68.30	2395.90*	−114.8%
49	4	25	135.43	3042.90*	366.17	3042.90*	−170.4%
50	4	30	75.70	4382.80*	985.70	4382.80*	−1202.1%
51	5	10	12.20	800.00*	7.30	800.00*	40.2%
52	5	20	196.30	2247.30*	165.30	2247.30*	15.8%
53	5	30	422.76	3683.52*	389.52	3683.52*	7.9%
54	5	40	3600.00	6069.75*	954.63	6069.75*	73.5%
55	5	50	3600.00	–	1324.2	9237.51*	63.2%
56	10	60	3600.00	–	60.15	10271.07*	98.3%
57	10	70	3600.00	–	95.32	11754.43*	97.4%
58	10	80	3600.00	–	141.35	12707.47*	96.1%
59	10	90	3600.00	–	201.72	14642.41*	94.4%
60	10	100	3600.00	–	276.44	15538.11*	92.3%
61	15	30	3600.00	–	40.03	3219.88*	98.9%
62	15	60	3600.00	–	151.55	9451.50*	95.8%
63	15	90	3600.00	–	256.70	16942.51*	92.9%
64	15	120	3600.00	–	318.64	24464.99*	91.1%
65	15	150	3600.00	–	3600.00	43490.10	0.0%

- ✓ **Small** instances: Gurobi can be faster.
- ✓ As **scale grows**: Gurobi runtime increases sharply and often fails to find a feasible solution within 1 hour.
- ✓ **Large** instances: B&P remains reliable and solves to optimality.

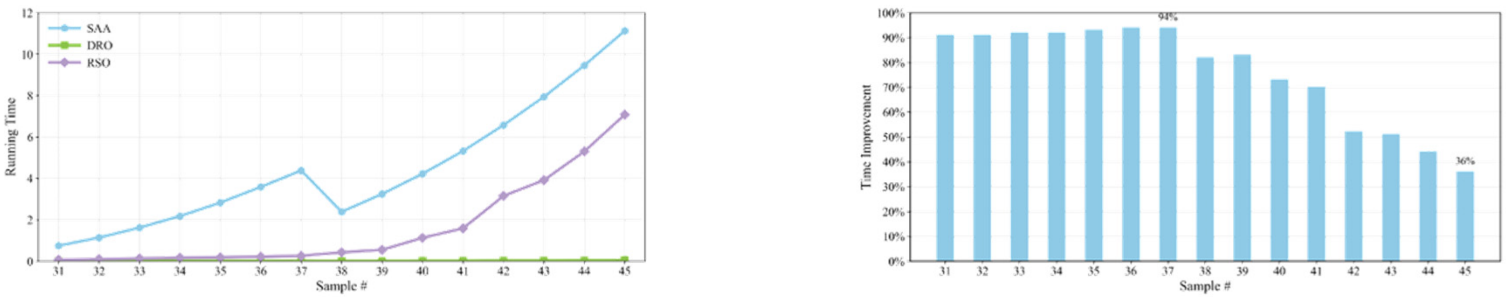
5.4 Efficiency of Feature Driven Approach

❑ Small-scale instances



(a) Average objective values (b) Relative improvements on objective

Fig. 6. Comparison of objective values and relative improvements of the FDRSO model over SAA and DRO in small-scale instances.



(a) Average computation time (b) Relative improvements on runtime

Fig. 7. Comparison of computation time and relative improvements of the FDRSO model over SAA in small-scale instances.

■ Benchmarks

- Compare FDRSO (with linear SVR) against SAA and DRO to assess solution quality and computational efficiency.
- DRO: FDRSO with S=1 (moment-based DRO without scenario differentiation).

5.4 Efficiency of Feature Driven Approach

❑ Medium-and large-scale instances

Table D.5: Comparison of B&P solutions for the FDRSO and DRO model.

Sample #	M	J	DRO		FDRSO		Improvement
			Time (s)	Objective	Time (s)	Objective	Objective
46	4	10	0.37	1134.10*	11.20	920.00*	18.9%
47	4	15	1.17	2084.70*	62.14	1491.00*	28.5%
48	4	20	31.80	3568.00*	68.30	2395.90*	32.9%
49	4	25	160.80	4662.00*	366.17	3042.90*	34.7%
50	4	30	644.70	6522.00*	985.70	4382.80*	32.8%
51	5	10	0.17	950.00*	0.53	769.33*	19.0%
52	5	20	8.27	3060.00*	3.13	2391.23*	21.9%
53	5	30	52.10	5503.00*	389.52	3683.52*	33.1%
54	5	40	105.60	8609.57*	954.63	6069.75*	29.5%
55	5	50	356.80	12866.10*	1324.20	9237.51*	28.2%
56	10	60	10.50	13900.76*	60.15	10271.07*	26.1%
57	10	70	15.80	15851.24*	95.32	11754.43*	25.8%
58	10	80	17.10	18075.18*	141.35	12707.47*	29.7%
59	10	90	25.60	19605.09*	201.72	14642.41*	25.3%
60	10	100	29.64	21730.51*	276.44	15538.11*	28.5%
61	15	30	15.40	4280.68*	40.03	3219.88*	24.8%
62	15	60	25.60	13521.03*	151.55	9451.50*	30.1%
63	15	90	105.60	23366.03*	256.70	16942.51*	27.5%
64	15	120	160.00	31183.52*	318.64	24464.99*	21.5%
65	15	150	3600.00	59895.60	3600.00	43490.10	27.4%

❑ Objective value

- FDRSO achieves the lowest objective, the advantage increases with problem size.
- SAA performs worst, DRO is intermediate.
- Average objective improvement:
  - vs SAA: 42%–52% reduction.
  - vs DRO: 23%–33% reduction.
  - Larger gains observed for larger J.

❑ Runtime

- FDRSO is slightly slower than DRO (more scenarios/complexity), but substantially faster than SAA.
- When S=9, FDRSO remain within a few seconds for small-scale instances and scale stably with size.

5.5 Analysis of Makespan Constraint

❑ Why a makespan constraint?

- Use C_{max} as an ε-constraint to cap schedule length while optimizing customer-oriented objectives.
- Bound is based on an LPT-derived upper bound and scaled by a slack factor ζ.

❑ Why slack is needed?

- The initial C_{max} is computed using predicted processing times + average setup times.
- LPT bound can be unreliable when processing times are similar, setup-time uncertainty further weakens makespan estimation.

Table 8
Computational time under different slack factors ζ.

Sample #	M	J	Slack Factor ζ					
			$\zeta = 0.75$	$\zeta = 1.00$	$\zeta = 1.25$	$\zeta = 1.50$	$\zeta = 1.75$	$\zeta = 2.00$
66	2	5	–	–	0.14	0.12	0.11	0.11
67	2	6	–	–	0.17	0.15	0.14	0.14
68	2	7	–	–	0.22	0.19	0.19	0.18
69	2	8	–	–	0.23	0.21	0.23	0.22
70	2	9	–	–	0.27	0.25	0.25	0.24
71	3	10	–	–	–	1.53	1.44	1.41
72	3	12	–	–	–	4.15	2.57	2.34
73	3	15	–	–	–	8.75	4.27	4.07
74	3	18	–	–	–	15.09	4.17	3.58
75	3	20	–	–	–	7.70	7.28	6.49
76	4	10	–	–	–	–	2.37	2.19
77	4	15	–	–	–	–	13.92	10.66
78	4	20	–	–	–	–	12.18	11.85
79	4	25	–	–	–	–	55.12	48.20
80	4	30	–	–	–	–	–	58.85

- Too-tight bounds ⇒ infeasibility.
- After feasibility is reached, increasing ζ further has minimal effect on objective, but runtime decreases as ζ.
- ζ=2 provides the most stable and efficient performance for large-scale, uncertain PCB scheduling.

6. Summary and Future Work

❑ Summary

- Proposed a **feature-driven robust stochastic scheduling framework**.
- Integrated: prediction, event-wise ambiguity sets, robust optimization.
- Developed an efficient **Branch-and-Price algorithm**.
- Outperformed SAA and traditional DRO methods.
- Successfully implemented in industrial PCBA systems.

❑ Future Work

- Reinforcement learning-based dynamic / online rescheduling.
- Energy-aware and multi-factory scheduling.
- LLM supported algorithm design.

Thanks!