



APIEMS2025 & SMILE2025



Manufacturing Data Science:

From Predictive, Prescriptive, to Optimization-Guided Learning

Chia-Yen Lee, Ph.D. (李家岩)

Associate Dean, College of Management
Professor, Department of Information Management
Director, Entrepreneurship and Innovation MBA
National Taiwan University

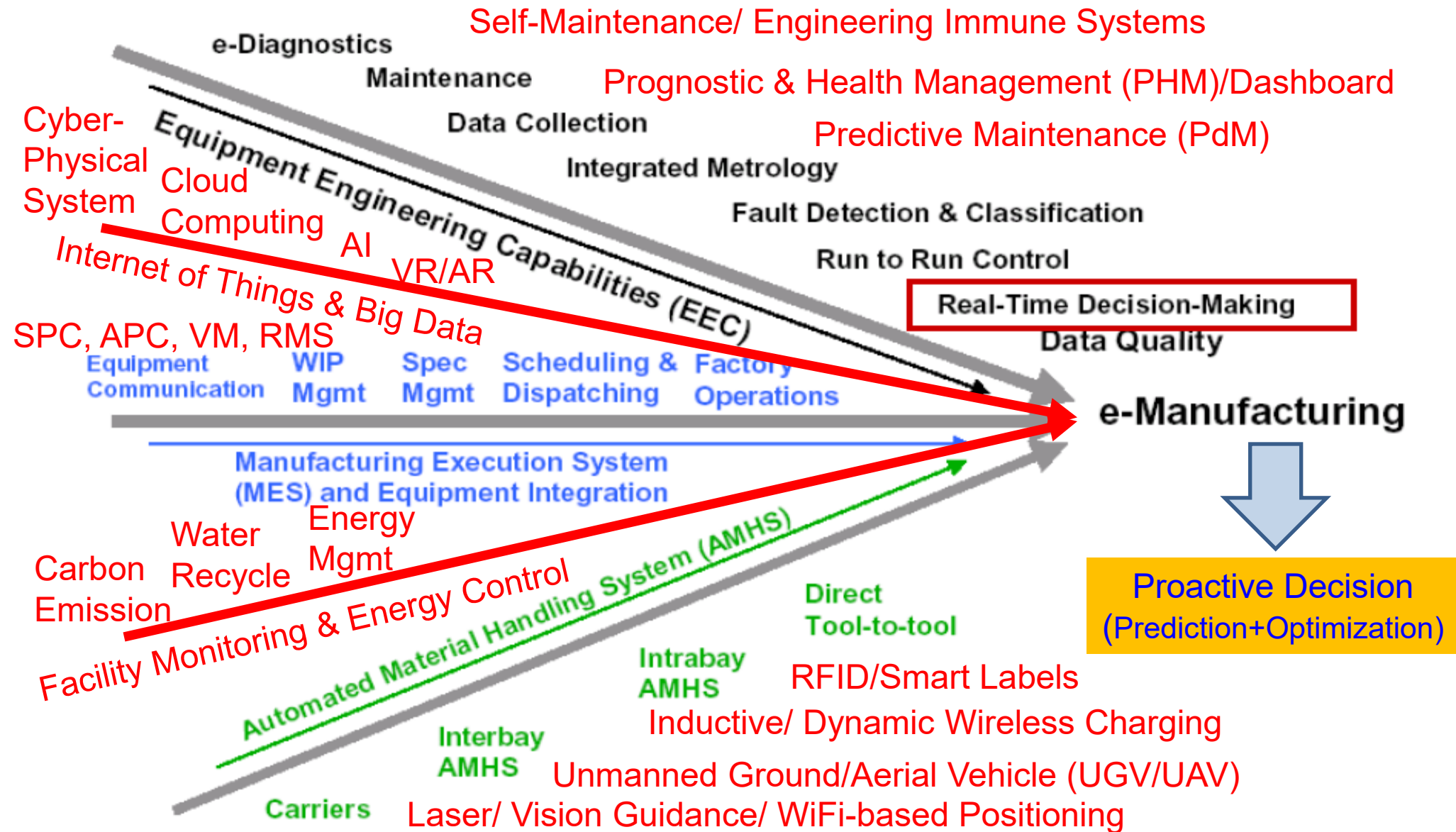
- Smart Manufacturing and Five-Phase Data Science Analytics
- From Predictive to Prescriptive Analytics
- From Prescriptive to Optimization-Guided Learning
- Takeaway

Smart Manufacturing is a **multi-objective decision-oriented** system which has the **computational intelligence** and **self-adaptive learning** ability to **autonomously optimize** the manufacturing resources and processes.

Computational Intelligence: Multimodal Fusion w/ Diverse Data Sources

Self-learning: Real-Time Monitoring & Spatio-Temporal Adaptive Control





Source: International SEMATECH e-Diagnostics and EEC Guidance 2003

□ Digitization (數位化)

- Reduce paper transmission
- Reduce **information asymmetry**

□ Digitalization (數位優化)

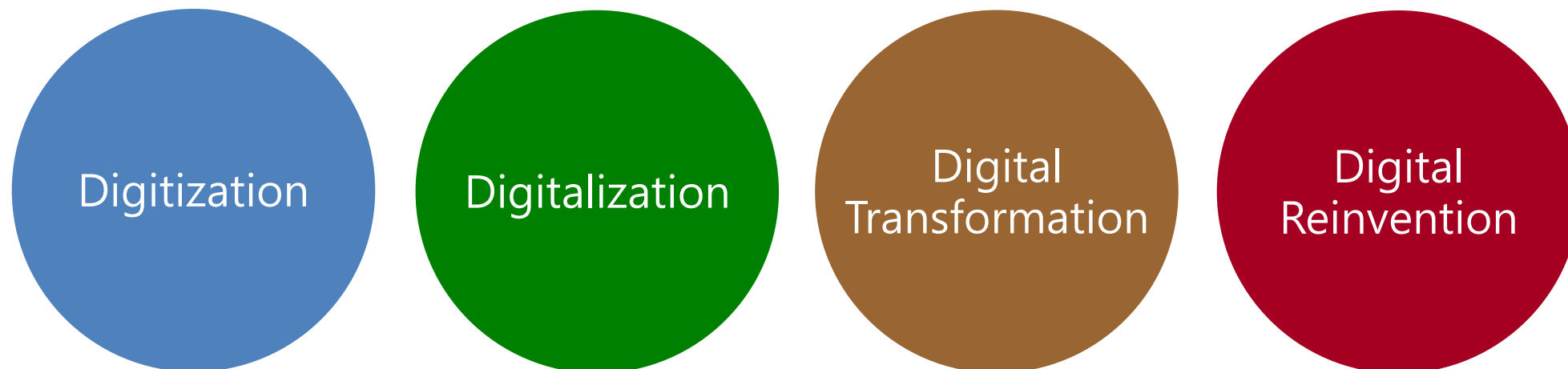
- **IT+OT**: integrating technology into the existing operation process via **ECRS** for **Lean Process**

□ Digital Transformation (數位轉型)

- **Not what you think, but what others see you.**
- **Re-define resources** and develop a novel value chain

□ Digital Reinvention (數位再造)

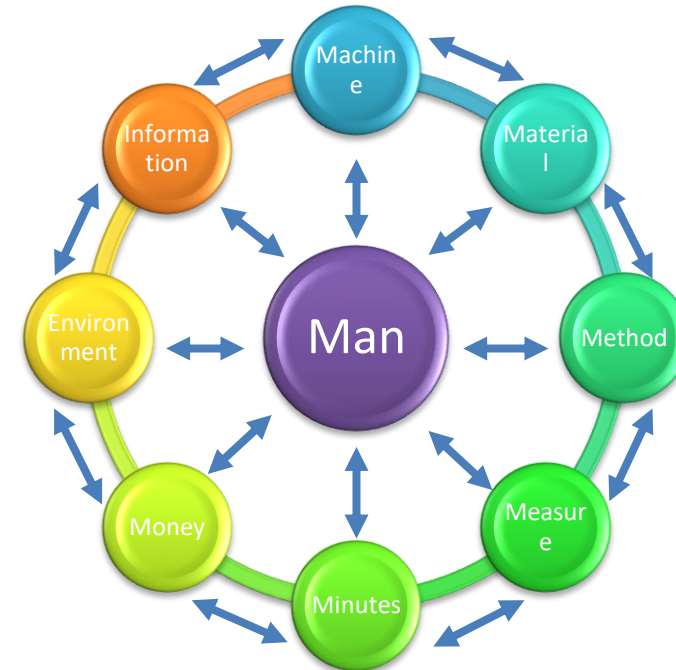
- Re-claim the **value proposition**
- Organizational revolution: human-machine collaboration



Human-centered



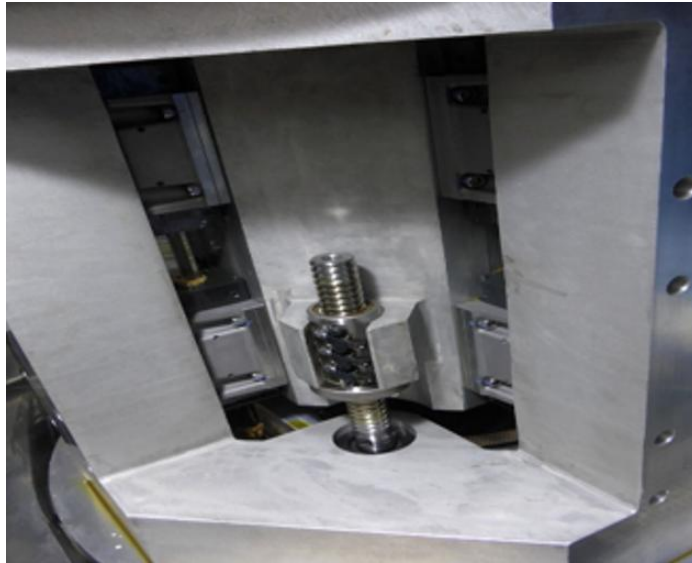
Internet of Things (IoT)



Resource 8M1I: 人(Man)、機(Machine)、料(Material)、方法(Method)、測量(Measure)、時間(Minutes)、資金(Money)、環境(Mother nature/environment)、資訊(information)

- ❑ **Pitfall 1.** Can ML/DS identify the **important** variables/features?
- ❑ **Pitfall 2.** Can put all raw variables (e.g. 1M) into feature selection technique?
- ❑ **Pitfall 3.** How to enhance the **interpretability** between predictors and response variable?
- ❑ **Pitfall 4.** How to handle when transforming categorical variable into too many dummy variables? (**one-hot encoding**)
- ❑ **Pitfall 5.** **Many missing values (e.g. over 50%)** in one variable/column (or observation/row)
- ❑ **Pitfall 6.** Does the **multicollinearity** problem matter?
- ❑ **Pitfall 7.** Does a higher prediction **accuracy support a better decision-making**?
- ❑ **Pitfall 8.** Does the selected variable not show the physical **causal relation**?
- ❑ **Pitfall 9.** Merging data tables and handling many missing values after the merge.
- ❑ **Pitfall 10.** How reliable is the conclusion derived from ML/DS?
- ❑ **Pitfall 11.** How to start the ML/DS works? **How much data we need**?
- ❑ **Pitfall 12.** How to develop roadmap and future works for smart factory?
- ❑ ... many many

□ Data Collection



Accelerometer



Current sensor



AE sensor



Autoencoder-based detector for distinguishing process anomaly and sensor failure

Chia-Yen Lee, Kai Chang and Chien Ho

Department of Information Management, National Taiwan University, Taipei, Taiwan

ABSTRACT

Anomaly detection is a frequently discussed topic in manufacturing. However, the issues of anomaly detection are typically attributed to the manufacturing process or equipment itself. In practice, the sensor responsible for collecting data and monitoring values may fail, leading to a biased detection result – false alarm. In such cases, replacing the sensor is necessary instead of performing equipment maintenance. This study proposes an effective framework embedded with autoencoder-based control limits that can dynamically distinguish sensor anomaly from process anomaly in real-time. We conduct a simulation numerical study and an empirical study of semiconductor assembling manufacturers to validate the proposed framework. The results show that the proposed model outperforms other benchmark methods and can successfully identify sensor failures, even under conditions of (1) large variations in process values or sensor values and (2) heteroscedasticity effect. This is particularly beneficial in various practical applications where sensors are used for numerical measurements and support equipment maintenance.

ARTICLE HISTORY

Received 8 July 2023
Accepted 4 February 2024

KEYWORDS

Prognostics and health management; sensor failure; anomaly detection; deep learning; autoencoder

1. Introduction

Prognostics and health management (PHM) (Lu and Lee 2022; Mobley 2002) is an approach to assess the health indicators of the equipment and estimate the life cycle, aiming to reduce inspection and time-based maintenance through real-time monitoring, incipient anomaly detection, and prediction of impending anomalies. In the manufacturing industry, the cost of repairs and maintenance typically accounts for 15–70% of total production cost, and thus it is critical to investigate the maintenance strategies, ranging from run-to-failure maintenance (e.g. reactive maintenance, RM), preventive maintenance (PM), conditional-based maintenance (CBM), to predictive maintenance (PdM). Deciding which maintenance strategy to apply to specific equipment conditions is crucial for achieving a better trade-off between maintenance frequency, capacity loss, and cost.

Typically, two anomalies may trigger the system alarm due to the out of control of the monitored value during machine degradation (Lee, Wu, and Hung 2021; Liu et al. 2019). We define these two anomalies as follows.¹ The first is a process anomaly, indicating that the equipment (or the environment where the equipment is located) may gradually become unreliable or deteriorate during pro-

not refer to an outlier in the monitored value; instead, it signifies a degradation process (e.g. characterised by the Wiener process) or increasing variation that persists after removing the outlier. In this case, PM is required to repair or recover the equipment's production function. The second scenario in this study is a sensor anomaly, signifying a problem unrelated to the manufacturing process but rather a sensor failure causing the monitored value to gradually drift upward or downward (i.e. time-varying drift). In this scenario, we usually replace the sensor instead of opting for PM. However, the monitored value is implicitly affected by the mixed effect of both scenarios, making it challenging to distinguish between process anomaly and sensor anomaly.

Although sensor anomalies randomly occur in the factory, the health status of the sensor is typically tested manually and periodically, with almost no inspection data of the sensor recorded in the database. Consequently, clarifying the root cause and distinguishing between the two scenarios becomes a time-consuming task. In practice, all anomalies are often treated as process anomalies, potentially leading to incomplete resolution of anomalies caused by the sensor failure and significant waste in maintenance costs. The concession of treating

□ Four Shortages

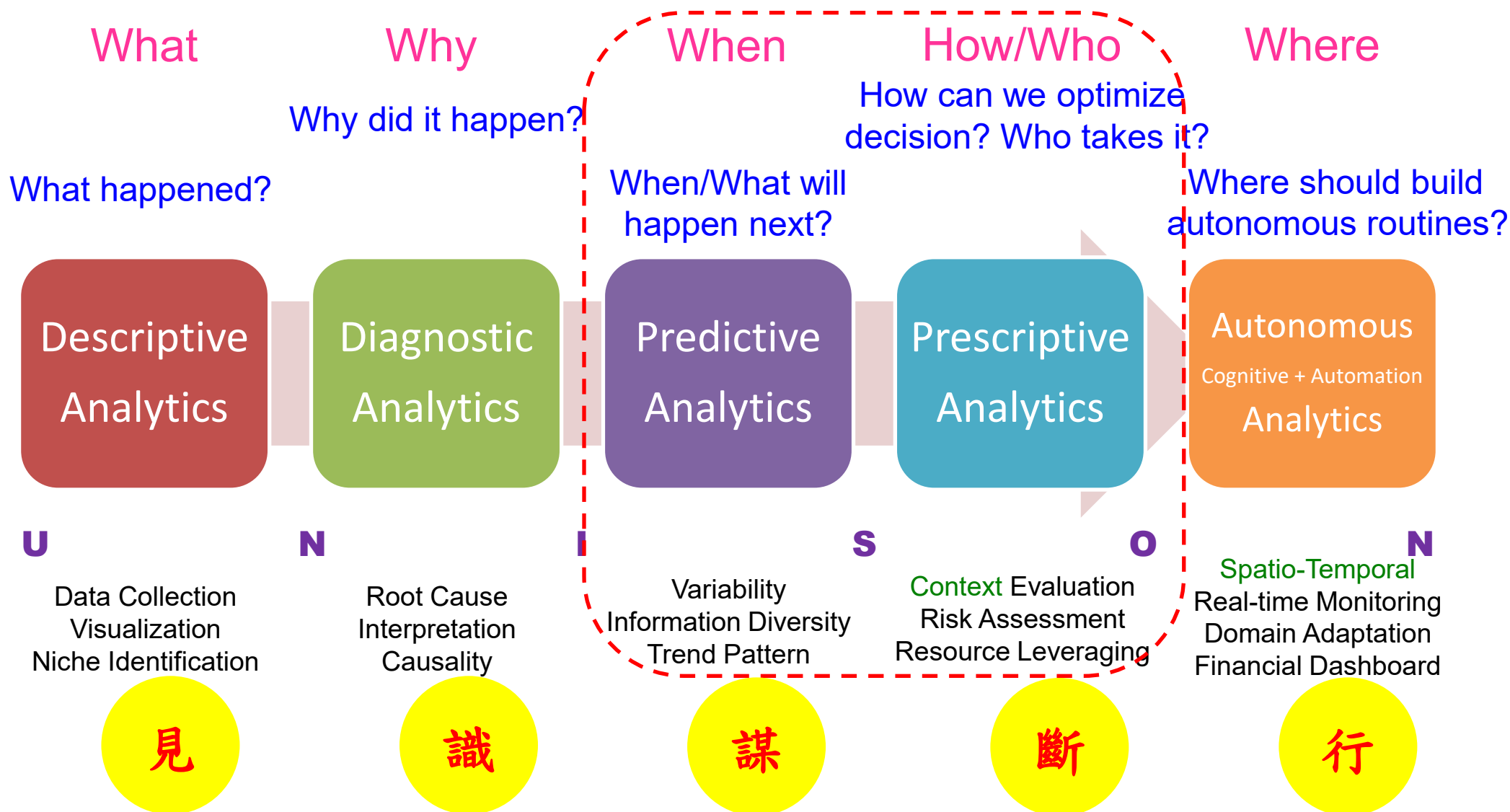
- Money
- Time
- Talent
- Best Practice

□ Build a new AI/DS team... but limited performance...

- Clean data (preprocessing, imputation, noise, poor quality)
- Keep asking user for **relabelling**... (quickly-changed environment)
- Prompt + RAG + API... (eliminate **hallucination**)
- Limited performance from AI projects... (**low morale**...staff quit...)
- No team to **bridge** between User and AI
- No user... (just for the sake of doing it)
- **Unclear** KPI/OKR (project does not connect to the business **strategy roadmap**)
- AI results argued by **domain expert** ("I won't let AI take my job")
- ... many many

Five-Phase Data Science Analytics

□ A Journey from *POINT*, *LINE*, to *PLANE*



□ We typically build AI models for prediction or scenario analysis

- CNN, LSTM, SVM, Random Forest, Boosting, PLS, ...
- Then...Which model is better? What's the next step after prediction?

□ Prediction is Risky!

- How about the potential risk (i.e. loss) after decision-making?

□ Example

- Model A with accuracy 95%, however, inaccurate prediction could lead to big loss.
- Model B with accuracy 90%, however, inaccurate prediction could lead to small loss.
- Which model do you prefer?

□ Predictive Thinking → Prescriptive Decision

❑ Confusion Matrix for Binary Classification

- Two risks: false alarm (type I) and miss rate (type II) → Prescriptive
- Trade-off between two misclassified errors → cost sensitive

Model A		Predict	
		FAIL	PASS
True	FAIL	61	7
	PASS	29	31

Model B		Predict	
		FAIL	PASS
True	FAIL	47	21
	PASS	7	53

	Testing	
	Accuracy	AUC
Model A	71.9%	70.2%
Model B	78.1%	78.9%

AUC: Area under the Curve of ROC

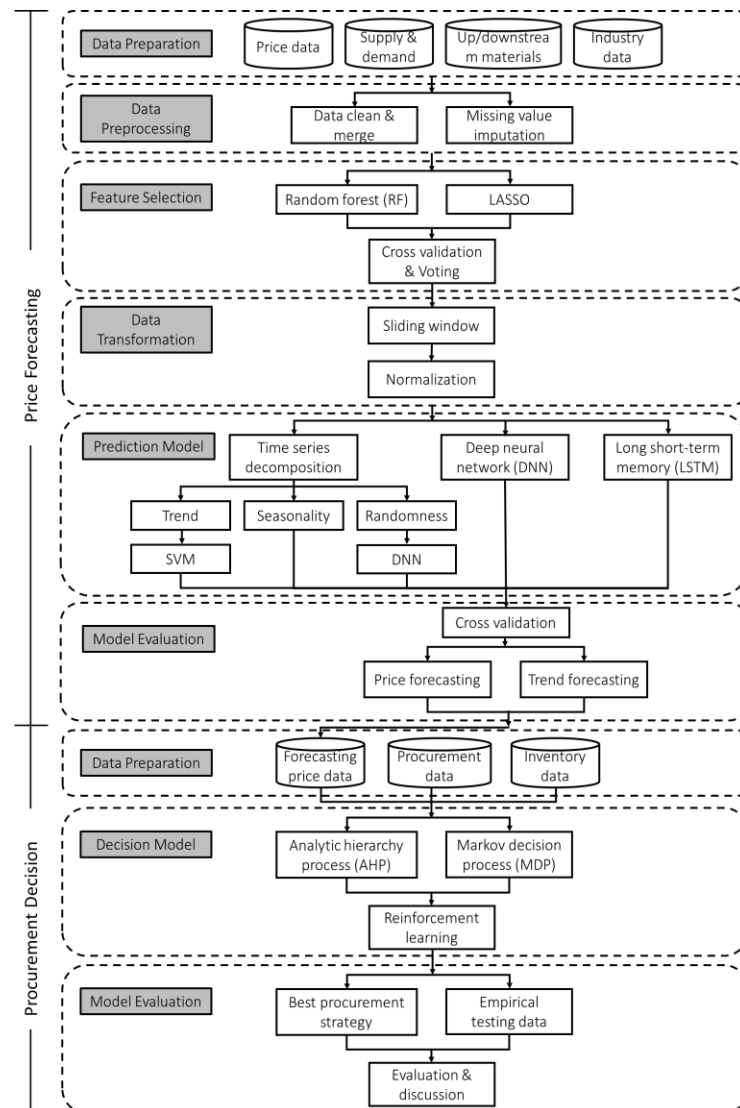
Lee, C.-Y., and Chien, C.-F., 2022. Pitfalls and protocols of data science in manufacturing practice. *Journal of Intelligent Manufacturing*, 33, 1189–1207.

Decisions take into account the **RISKS** associated with the realization of **uncertain** events.

Raw Material Price Prediction and Procurement

Feature Extraction

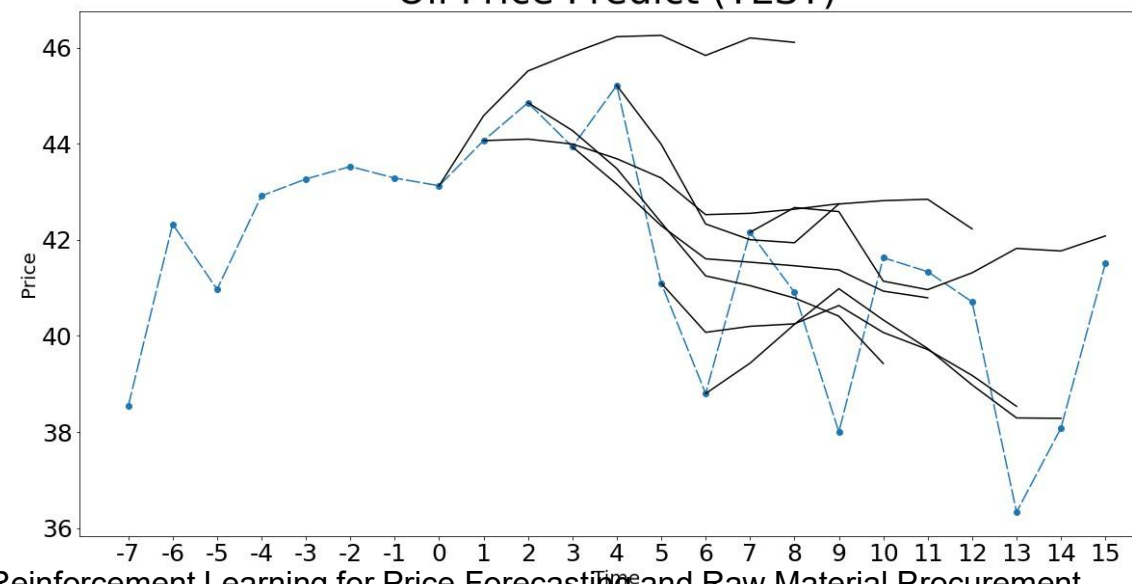
Price Forecast (Predictive Analytics)



Crude Oil			Upstream			Butadiene			Downstream		
Variable	r.c.	c.c.	Variable	r.c.	c.c.	Variable	r.c.	c.c.	Variable	r.c.	c.c.
X01	17.9591	0.5195	X03	2.2597	0.8511	Y			X22	5.6786	0.7958
X02	18.3465	0.5229	X04	0.8613	0.1990	X10	0.9936	0.9997	X23	0.7643	0.9258
			X05	1.6192	0.5878	X11	1.0051	0.9997	X24	0.8698	0.9424
			X06	-0.3626	-0.1536	X12	0.8651	0.8741	X25	0.9213	0.7151
			X07	2.0611	0.5830	X13	0.9153	0.8584	X26	0.8003	0.9272
			X08	-0.4406	-0.4160	X14	0.8382	0.9401	X27	0.8897	0.9313
			X09	0.0002	0.4336	X15	0.8153	0.9406	X28	0.0008	0.5036
						X16	0.8615	0.9390	X29	0.0005	0.4319
						X17	0.8751	0.8425	X30	1.0112	0.9810
						X18	0.8802	0.9512	X31	1.3185	0.9753
						X19	0.2148	0.0719	X32	0.1570	0.1345
						X20	0.3655	0.1411	X33	0.7595	0.3052
						X21	0.0012	0.4902	X34	1.4520	0.6067

Legend	
	Butadiene(BD)
	r.c. > 1.5
	r.c. < 0
★	Important variables identified by feature selection

Prediction Model Oil Price Predict (TEST)



Lee, C.-Y., Chou, B.-J., and Huang, C.-F. 2021. Data Science and Reinforcement Learning for Price Forecasting and Raw Material Procurement in Petrochemical Industry. *Advanced Engineering Informatics*, 51, 101443.

Raw Material Price Prediction and Procurement

Reinforcement Learning (Prescriptive Analytics)

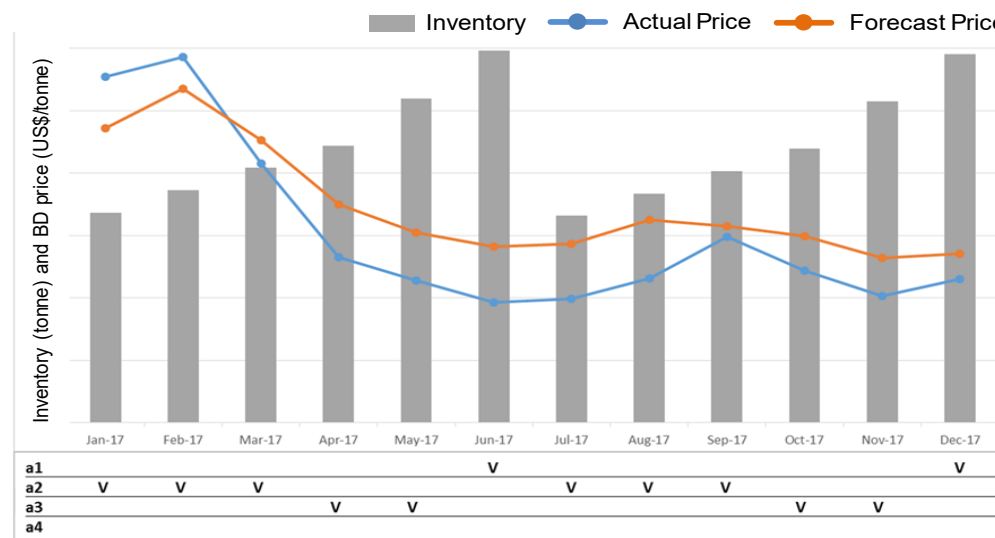
Human Judgment

Current Policy



Reinforcement Learning

Optimal Policy

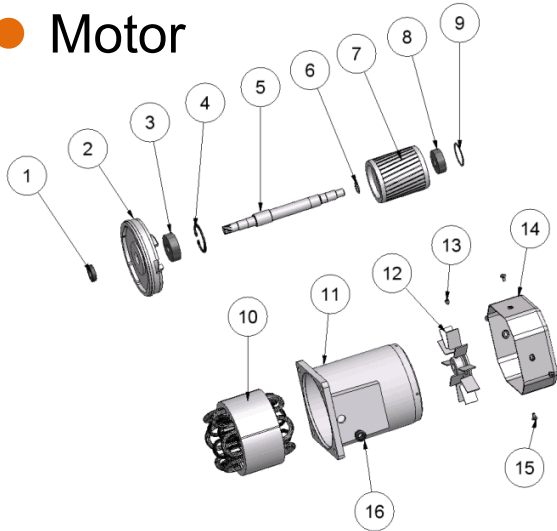


	Current policy	(s,S) policy	Optimal policy
Average inventory (tonne)	3112	1812	3197
Standard deviation of inventory (tonne)	743	302	489
Amount purchased (tonne)	25,301	35,430	36,835
Total cost (US\$)	44,596,113	42,324,694	39,091,618

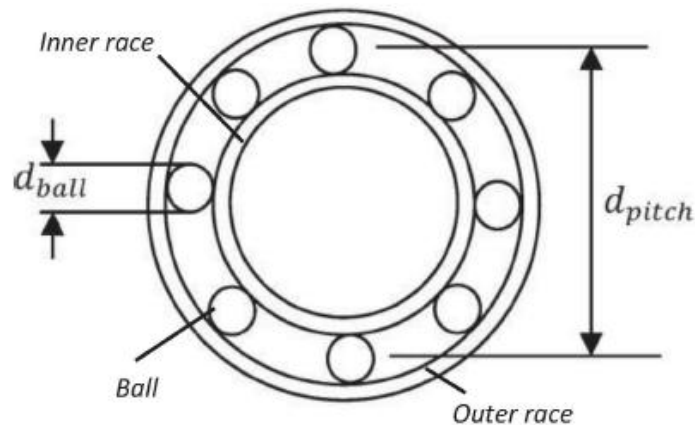
Lee, C.-Y., Chou, B.-J., and Huang, C.-F. 2021. Data Science and Reinforcement Learning for Price Forecasting and Raw Material Procurement in Petrochemical Industry. *Advanced Engineering Informatics*, 51, 101443.

Remaining Useful Life (RUL) (Predictive Analytics)

Motor

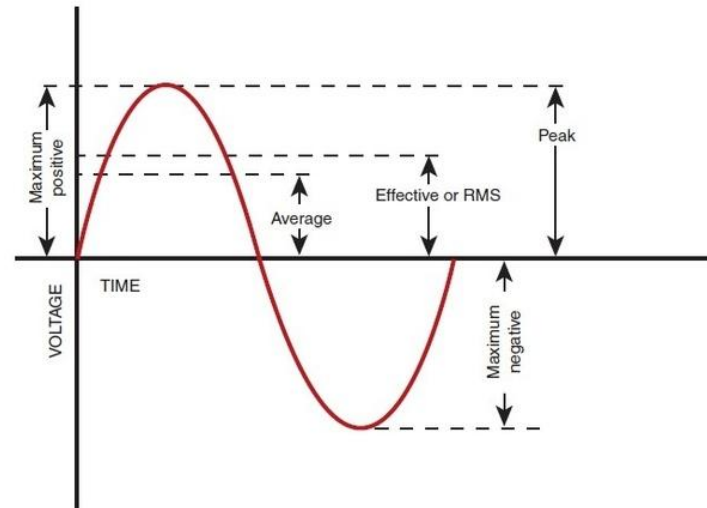
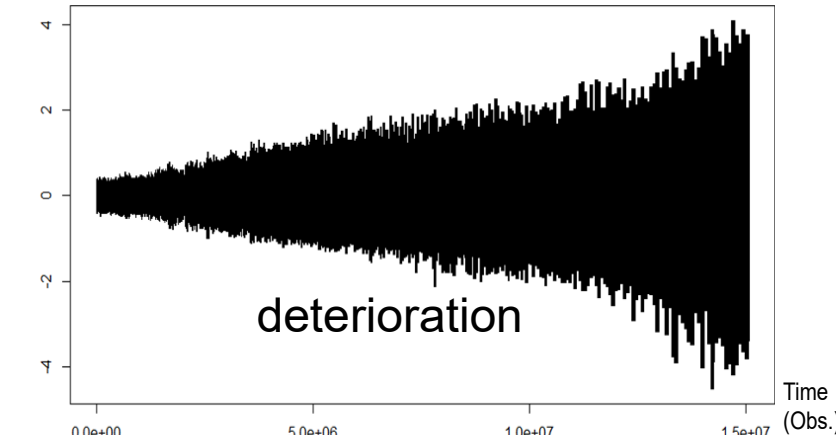


Bearing



Data Source

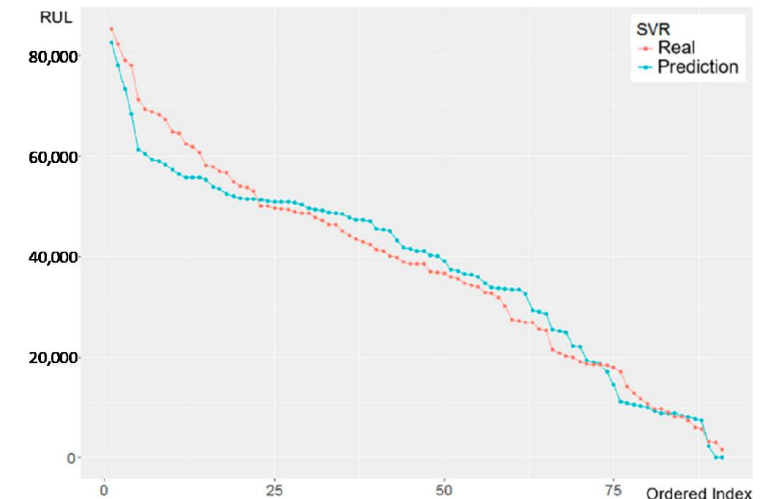
— vibration acceleration signal



Time Domain, Frequency Domain, Time-frequency Domain

Feature	Equation
Maximum Value	$f_{1i} = \max(X(i))$
Mean Value	$f_{2i} = \frac{1}{N} \sum_{i=1}^N X(i)$
Minimum Value	$f_{3i} = \min(X(i))$
Standard Value	$f_{4i} = \sqrt{\frac{1}{N} \sum_{i=1}^N (X(i) - f_{1i})^2}$
Peak to Peak Value	$f_{5i} = f_{1i} - f_{3i}$
Mean Amplitude	$f_{6i} = \frac{1}{N} \sum_{i=1}^N X(i) $
Root Mean Square Value	$f_{7i} = \sqrt{\frac{1}{N} \sum_{i=1}^N X(i)^2}$
Skewness Value	$f_{8i} = \frac{1}{N} \sum_{i=1}^N X(i)^3$
Kurtosis Value	$f_{9i} = \frac{1}{N} \sum_{i=1}^N X(i)^4$
Waveform Indicator	$f_{10i} = \frac{f_{7i}}{f_{6i}}$
Pulse Indicator	$f_{11i} = \frac{f_{1i}}{f_{6i}}$
Kurtosis Index	$f_{12i} = \frac{f_{9i}}{f_{7i}}$
Peak Index	$f_{13i} = \frac{f_{1i}}{f_{7i}}$
Square Root Amplitude	$f_{14i} = (\frac{1}{N} \sum_{i=1}^N \sqrt{ X(i) })^2$
Margin Indicator	$f_{15i} = \frac{f_{1i}}{f_{14i}}$
Skewness Indicator	$f_{16i} = \frac{f_{8i}}{f_{7i}^3}$

RUL Prediction



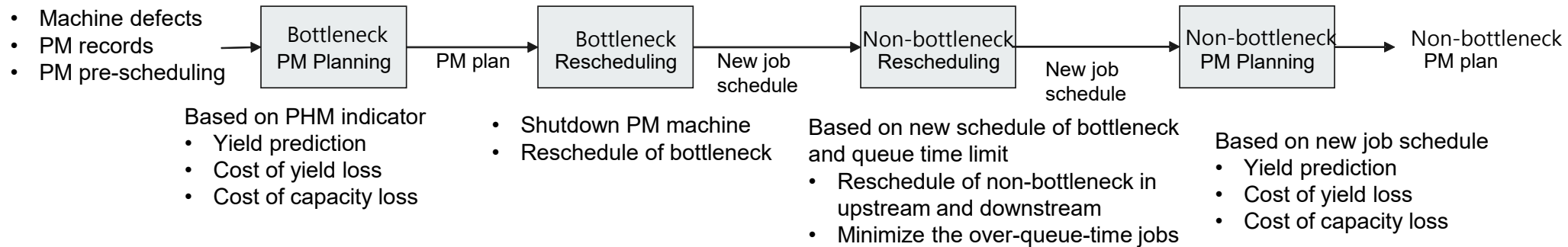
<http://www.li-ming.com.tw/>

Lee, C.-Y., T.-S. Huang, M.-K. Liu, and C.-Y. Lan. 2019. Data Science for Vibration Heteroscedasticity and Predictive Maintenance of Rotary Bearings. *Energies*, 12 (5), 801.

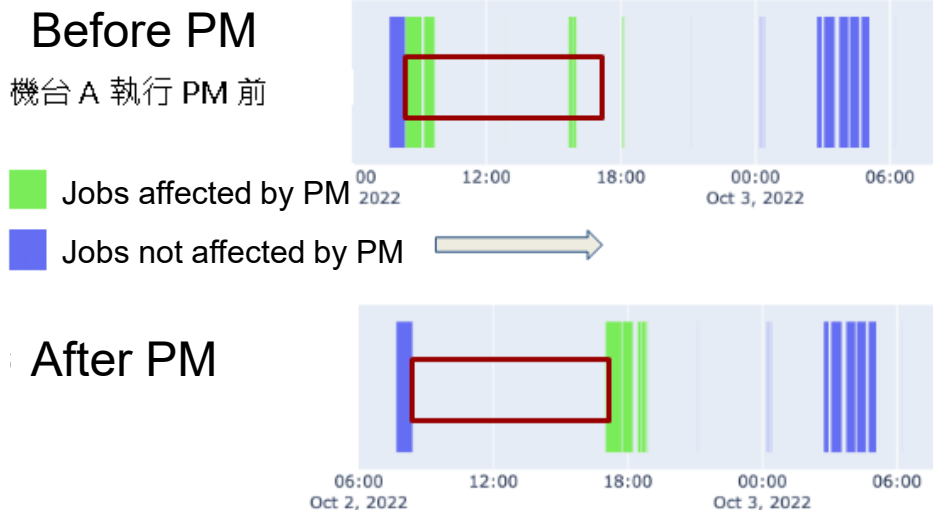
Jiang, W., Hong, Y., Zhou, B., He, X. and Cheng, C. 2019. "A gan-based anomaly detection approach for imbalanced industrial time series," *IEEE Access*, vol. 7, pp. 608–619.

□ Predictive Maintenance (PdM) Scheduling (**Prescriptive Analytics**)

- The proposed four-stage PM (preventive maintenance) algorithm provides a tradeoff between machine workload (**capacity loss**) and condition (**PHM indicator**; **yield loss**), and integrates non-bottleneck machines in upstream and downstream of the bottleneck.



Bottleneck machine PM at 9am



- The system considers bottleneck and connects them to upstream and downstream;
- Supports machine **yield prediction**;
- Estimates the costs of **yield loss** and **capacity loss**;
- Incorporates queue time limit and maintenance resources (available labor hour) into the model;
- Considers the production **uncertainty** for developing stochastic dynamic programming;
- And recommends the priority of machine PM.

Kung, L.-C., and Liao, Z.-Y. (2022). "Optimization for a joint predictive maintenance and job scheduling problem with endogenous yield rates", IEEE Transactions on Automation Science and Engineering, 19(3), 1555-1566.

Prediction is the **Process**; Decision is the **Purpose**.

Workflow: From Predictive to Prescriptive Analytics

1st stage: Predictive Analytics

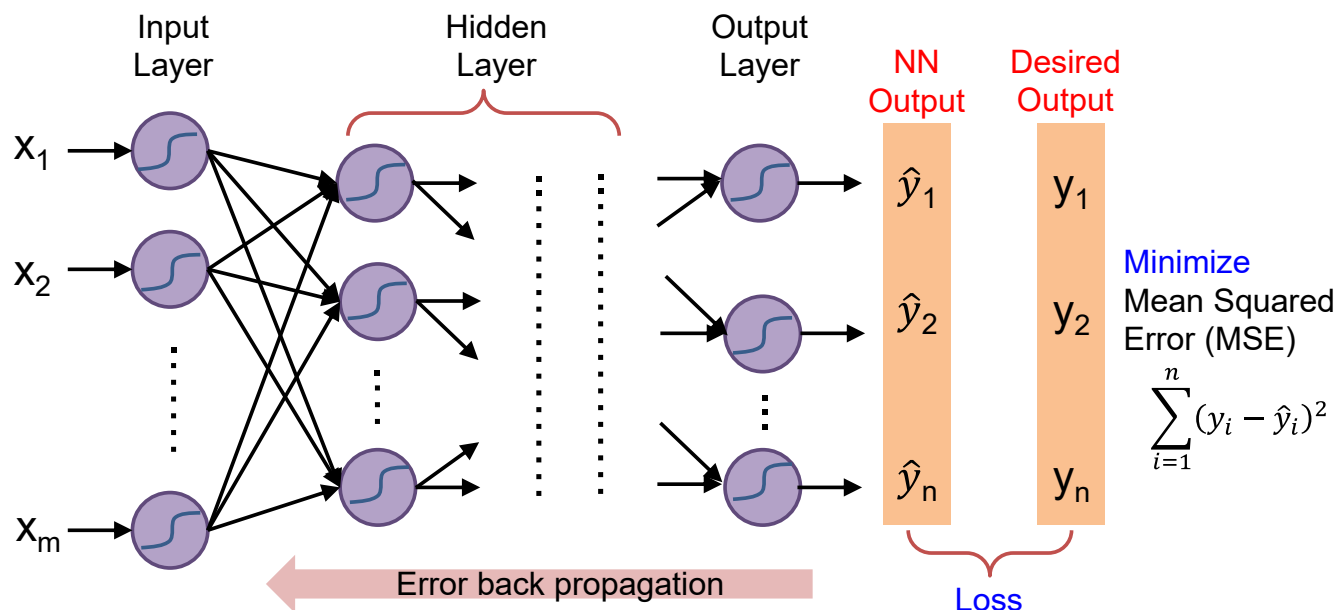
- Estimation
- Prediction or forecast is difficult
 - because it's about the **FUTURE**
- Estimation or imputation is reasonable
 - Use known information to estimate **UNKNOWN** information

2nd stage: Prescriptive Analytics

- Optimization
- Decision-maker's **preference structure**
 - **Multi-objective** decision analysis
- Resource allocation optimization
 - 8M1I: 人(Man)、機(Machine)、料(Material)、方法(Method)、測量(Measure)、時間(Minutes)、資金(Money)、環境(Mother nature/environment)、資訊(information)
- Risk assessment & diversification

Example

1st Stage



2nd Stage

$$\begin{aligned} \min \quad & C^T x \\ \text{s.t.} \quad & Ax \geq b \\ & \tilde{T}x \geq \tilde{r} \\ & x \geq 0 \end{aligned}$$

Problems with “Predict-then-Optimize”

□ Prediction **errors** do not align with decision costs (Muñoz et al., 2022)

- The cost of over-prediction and under-prediction is often **asymmetric**.

Classifier A		Predict	
		FAIL	PASS
True	FAIL	61	7
	PASS	29	31

Classifier B		Predict	
		FAIL	PASS
True	FAIL	47	21
	PASS	7	53

□ Example:

- In electricity markets or reserve scheduling, **system operators often intentionally bias forecasts**, since under-prediction incurs extremely high costs (CAISO, 2020).

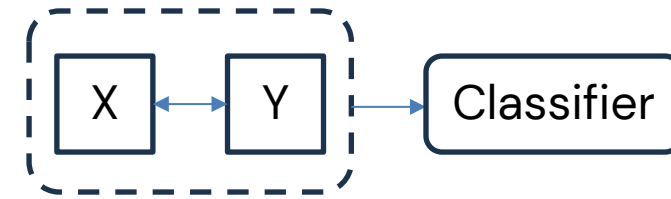
□ Insights

- The forecasting model should account for decision costs — sometimes, a biased forecast actually leads to better decisions.

□ Inspired by Feature Selection Technique

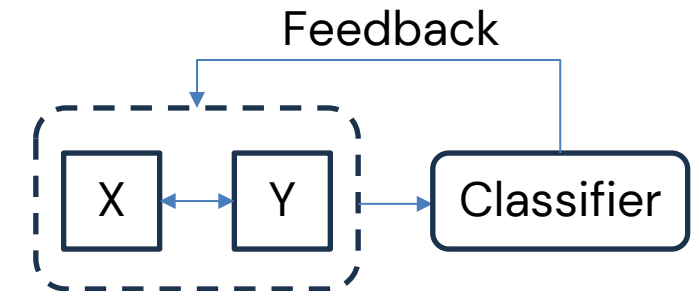
□ Filter Method

- Feature Ranking
- Method: correlation, p-value, coefficient ranking (w/ normalized data) in regression



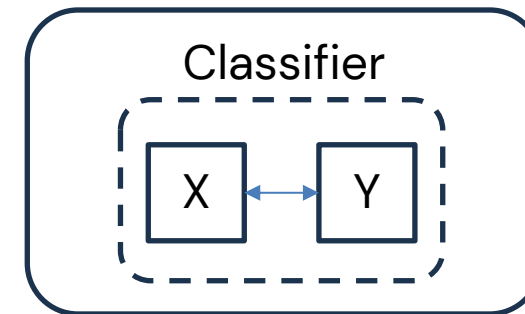
□ Wrapper Method

- Iterative feedback loop
- Method: Best Subset Selection, Stepwise Regression



□ Embedded Method

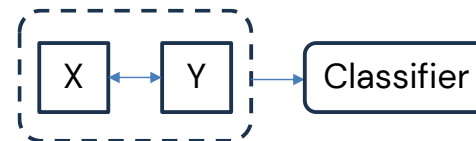
- Shrinkage method w/ regularization term
- Method: LASSO, Ridge, ElasticNet



Predictive and Prescriptive Integration

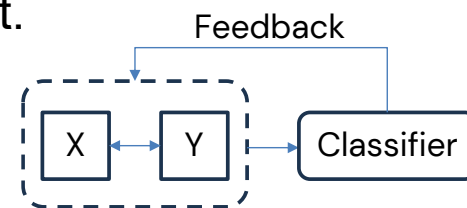
□ Open Loop Model

- Predict-then-Optimize
- Method: **Three** Phases including **Forecasting** Phase, **Decision** Phase, **Realization** Phase
 - Realization: When the actual data is observed, the decision maker adapts by balancing redispach within scheduled reserves (tolerance) and calculating the exact cost.



□ Closed Loop Model (Garcia et al., 2025)

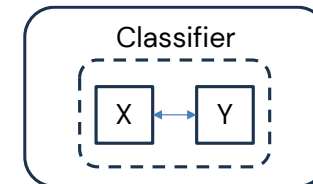
- **Iterative feedback loop**: The model **learns forecasting parameters (θ)** that directly improve **real operational outcomes**, not only prediction accuracy.
- Method: Bi-Level Optimization



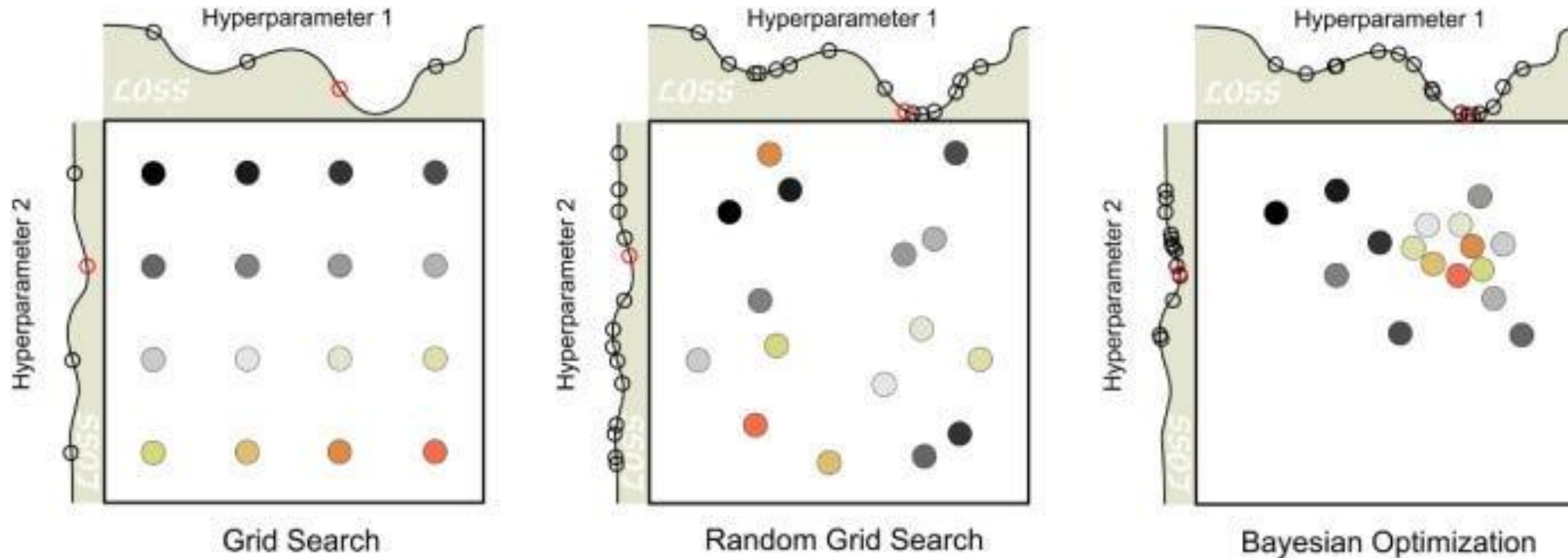
$$\begin{aligned} \theta_T \in \arg \min_{\theta \in \Theta, \hat{y}_t, z_t^*} & \quad \frac{1}{T} \sum_{t \in \mathbb{T}} G_a(z_t^*, y_t) \\ \text{s.t.} & \quad \hat{y}_t = \Psi(\theta, x_t) \quad \forall t \in \mathbb{T} \\ & \quad z_t^* \in \arg \min_{z \in Z} G_p(z, \hat{y}_t) \quad \forall t \in \mathbb{T} \end{aligned}$$

□ Optimization-Guided Learning (Lee et al, 2024a; 2024b)

- **Embedded method**: use optimization to guide the model training process
- Method: Genetic Algorithm embedded with Reinforcement Learning (GAeRL), Reinforcement Learning embedded with Robust Optimization (RLeRO)

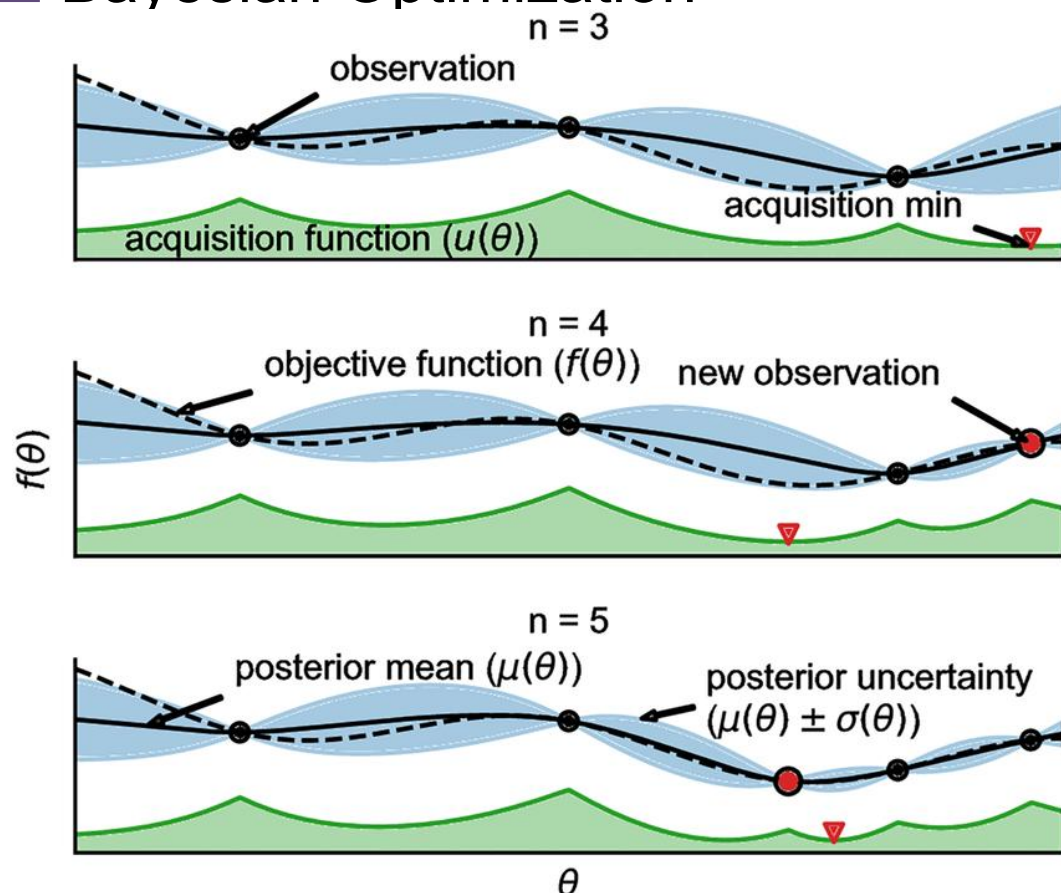


□ Example: Hyperparameter Optimization in Learning Algorithm



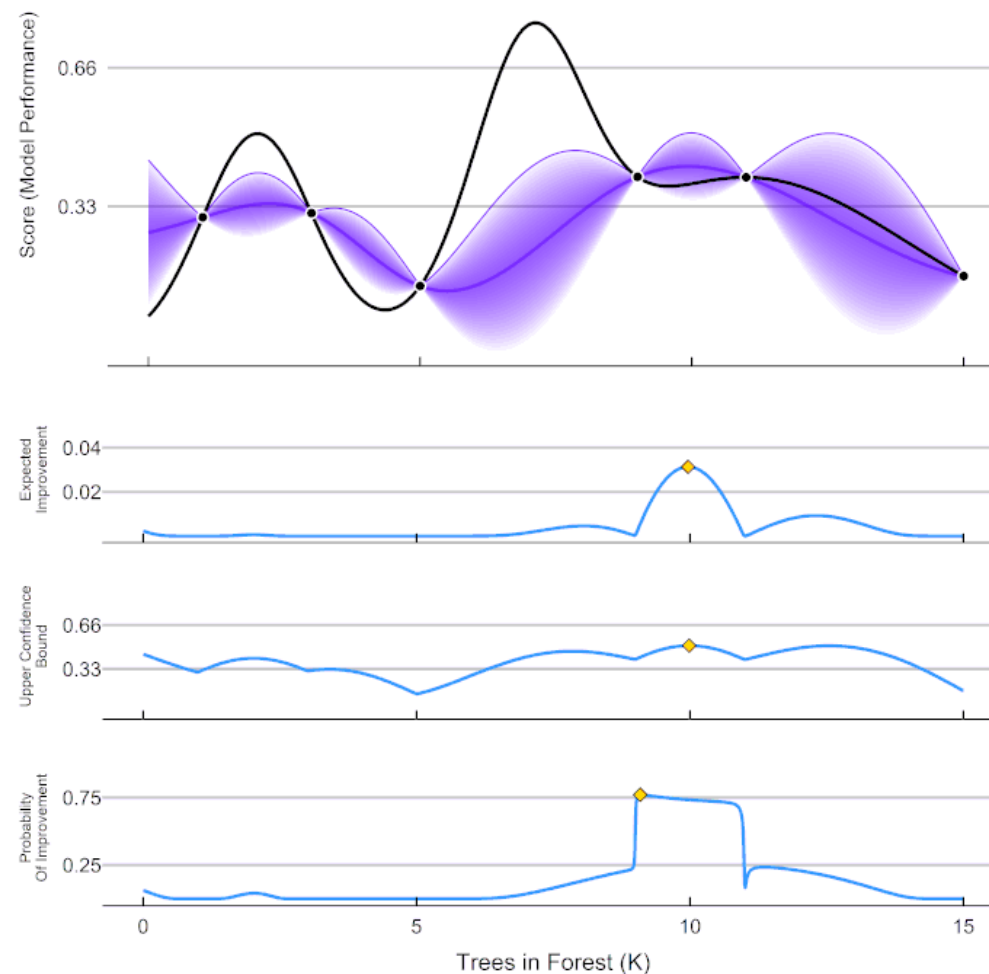
Closed Loop Model: Iterative Feedback Loop

Bayesian Optimization



Bayesian optimization example: Three iterations of Bayesian optimization minimizing a 1D function. The figure shows a Gaussian process (GP) approximation (solid black line and blue shaded region) of the underlying objective function (dotted black line). The figure also shows the acquisition function (green). The acquisition function (GP-LCB, lower confidence bound) is the difference of the mean and variance of the GP (multiplied by a constant), which Bayesian optimization minimizes to determine where to sample next.

ParBayesianOptimization in Action (Round 1)

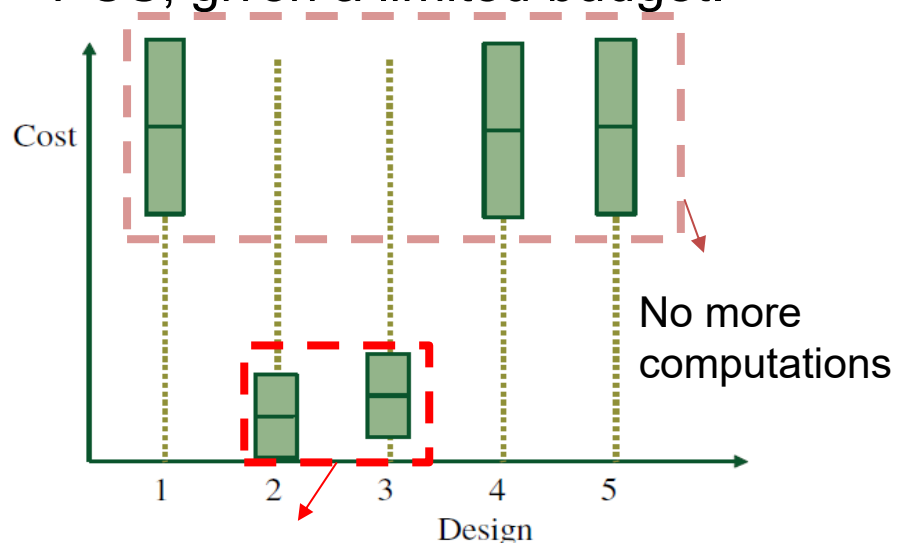


<https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1006606>
https://en.wikipedia.org/wiki/Bayesian_optimization

Closed Loop Model: Iterative Feedback Loop

Optimal Computing Budget Allocation (OCBA) (Chen and Lee, 2011)

- Given a limited computing power, OCBA finds the **best alternative** (i.e. parameter design) by maximizing the probability of correct selection (PCS). OCBA maximizes PCS, given a limited budget.



Need more computations
for simulation

Notations:

K : a set of designs (alternatives)

$b \in K$: the best design

n_i : # of simulation allocated for design $i \in K$

μ_i : the mean of fitness value for design i

T : the total computing budget

OCBA Model

$$\max_{N_1, \dots, N_{|K|}} PCS = 1 - \sum_{i \in K, i \neq b} P\{\tilde{\mu}_b > \tilde{\mu}_i\}$$

Approximate Probability of Correct Selection (APCS)

$$\text{s.t. } \sum_{i \in K} n_i \leq T, \quad PCS \geq APCS$$

$$n_i \geq 0, \forall i \in K.$$

Let σ_i be the variance for design i . PCS can be asymptotically maximized when the relationship between two non-best design i and j , where $i \neq j \neq b$, in the l th iteration.

$$\begin{cases} \frac{n_i^{l+1}}{n_j^{l+1}} = \left(\frac{\sigma_i / (\tilde{\mu}_b - \tilde{\mu}_i)}{\sigma_j / (\tilde{\mu}_b - \tilde{\mu}_j)} \right)^2 \\ n_b^{l+1} = \sigma_b \sqrt{\sum_{i \in K, i \neq b} (n_i^{l+1} / \sigma_i)^2} \end{cases}$$

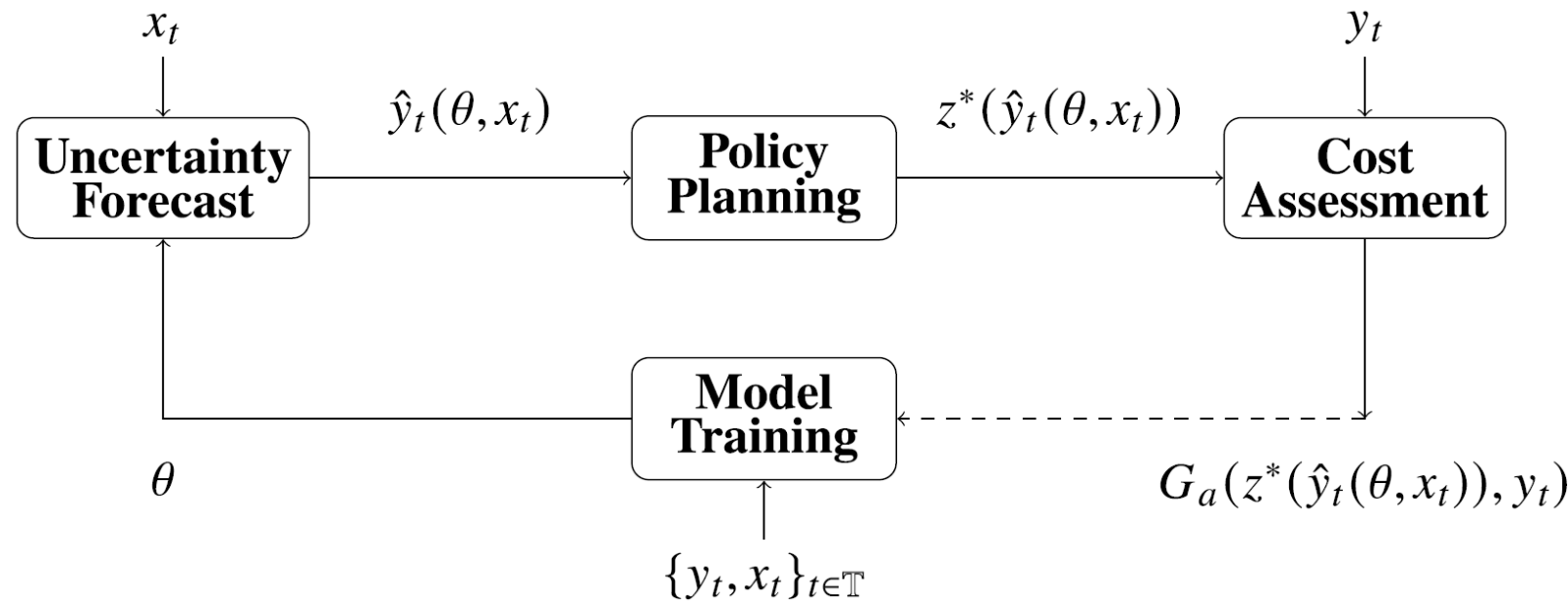
of simulation replications for the best design

OCBA should be the best Ranking and Selection process (Branke et al., 2007)

Closed Loop Model: Iterative Feedback Loop

□ Closed Loop Model (Garcia et al., 2025)

- Forecasts are generated by the prediction model $\Psi(\theta, x_t)$, which produces $\hat{y}_t$ based on parameters θ and features x_t , and then used as inputs to optimize G_p to obtain optimal decision z_t^* for each t . Each decision $z_t^*(\hat{y}_t)$ is used as an input to optimize G_a , which is then **evaluated against the actual outcome y_t** using an assessment cost. G_a is used to **update** the forecasting parameter θ , **creating feedback**.



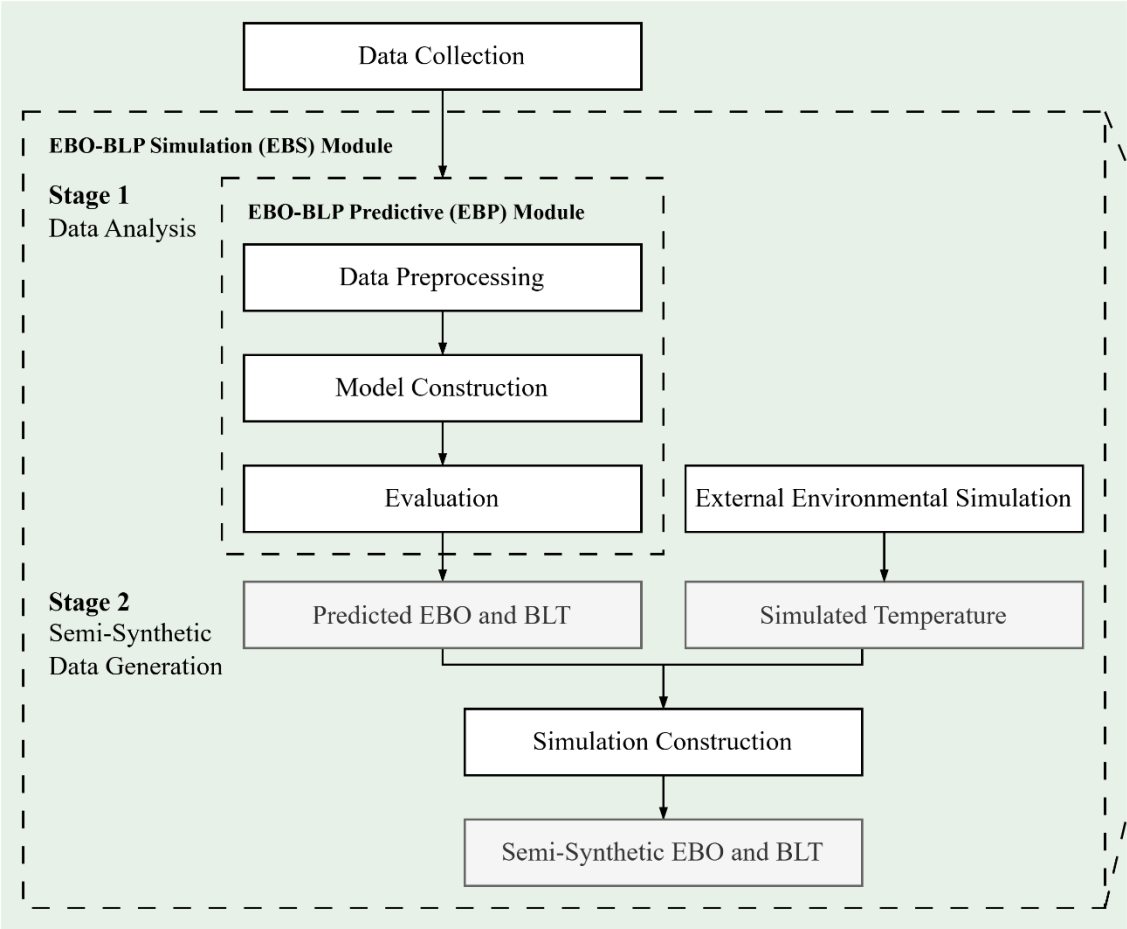
$$\theta_T \in \arg \min_{\theta \in \Theta, \hat{y}_t, z_t^*} \frac{1}{T} \sum_{t \in T} G_a(z_t^*, y_t)$$

$$s.t. \quad \hat{y}_t = \Psi(\theta, x_t) \quad \forall t \in T$$

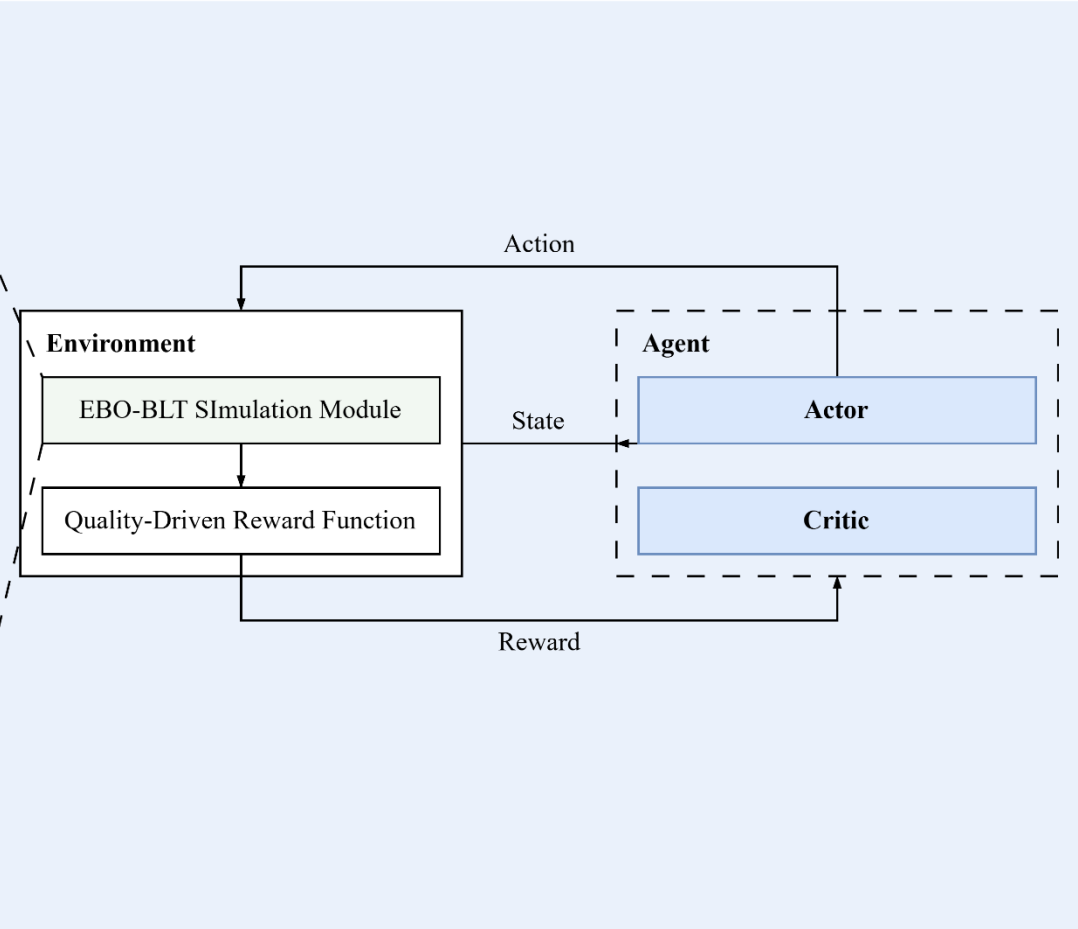
$$z_t^* \in \arg \min_{z \in Z} G_p(z, \hat{y}_t) \quad \forall t \in T$$

Predictive Analytics

(Data or Environment Simulator)



Prescriptive Analytics



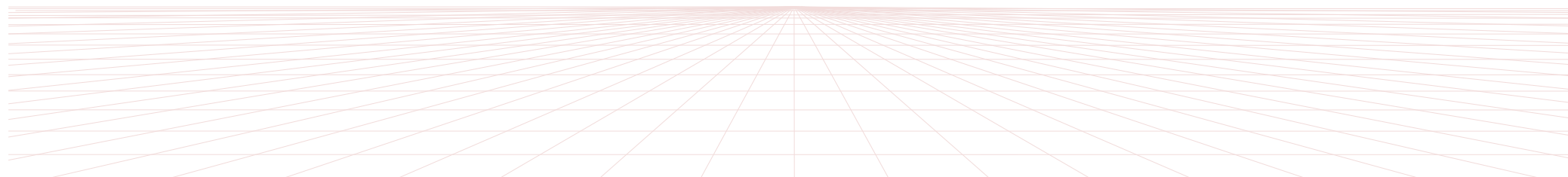
Optimization-Guided Learning (OGL)

- Genetic Algorithm embedded with Reinforcement Learning (GAeRL)

Lee, C.-Y., Ho, C.-Y., Hung, Y.-H., and Deng, Y.-W., 2024. Multi-objective genetic algorithm embedded with reinforcement learning for petrochemical melt-flow-index production scheduling. ***Applied Soft Computing***, 159, 111630.

- Reinforcement Learning embedded with Robust Optimization (RLeRO)

Lee, C.-Y., Huang, Y.-T., and Chen, P.-J., 2024. Robust-optimization-guiding deep reinforcement learning for chemical material production scheduling. ***Computers and Chemical Engineering***, 187, 108745.



□ Petrochemical Production Scheduling (化工廠排程特性)

● Objective Functions

— Minimize tardiness (satisfying due date)

➤ Total Tardiness = $\sum_{j \in J} T_j = \sum_{j \in J} \max\{C_j - D_j, 0\}$

— Minimize # of conversion, transition time, volume of transition product

➤ Transition Products = $\sum_{t \in T} (MFI_{t+1} - MFI_t)^2$

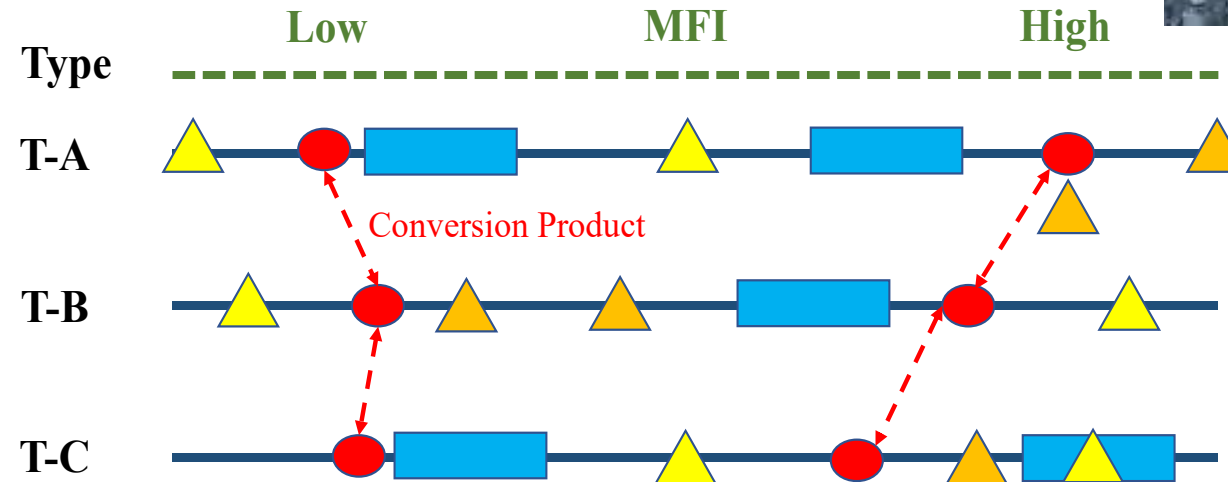
● Constraints

— Type Conversion Constraint

— Specific Group Constraint

— Melt-flow-index (MFI) Slowly-Rise-and-Fall Constraint

— Sequence-Dependent Transition Time Constraint



Blömer, F., & Günther, H.-O. (1998). Scheduling of a multi-product batch process in the chemical industry. Computers in industry, 36(3), 245-259.

Blömer, F., & Günther, H.-O. (2000). LP-based heuristics for scheduling chemical batch processes. International Journal of Production Research, 38(5), 1029-1051.

Scheduling Methodology

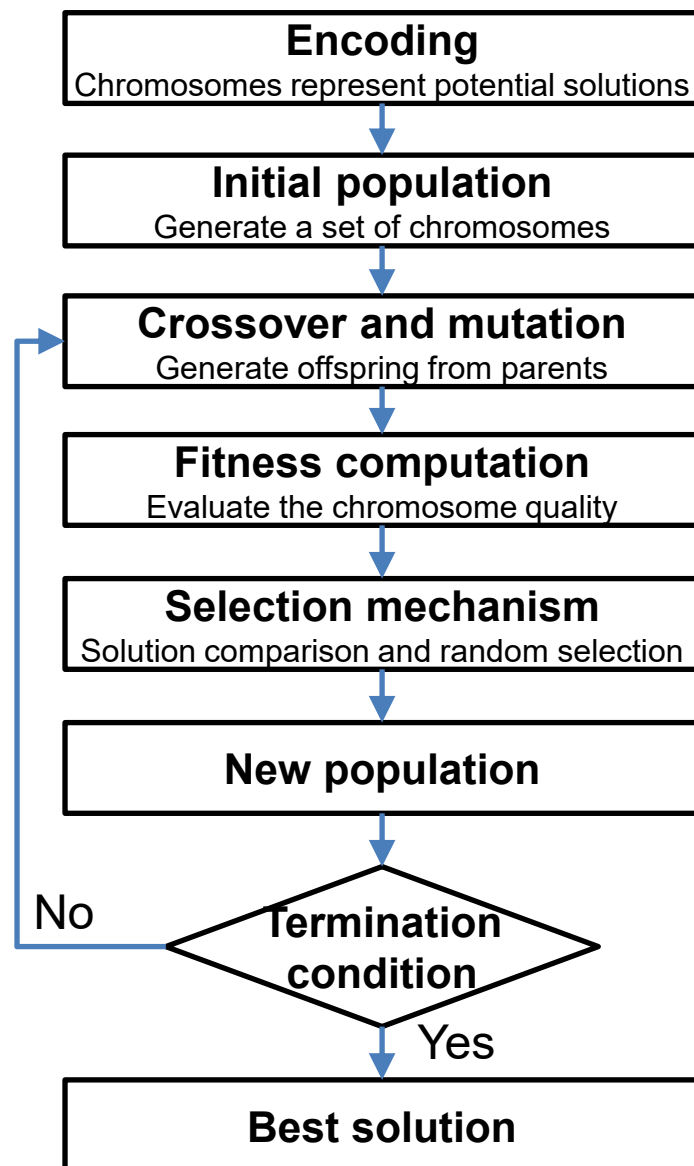
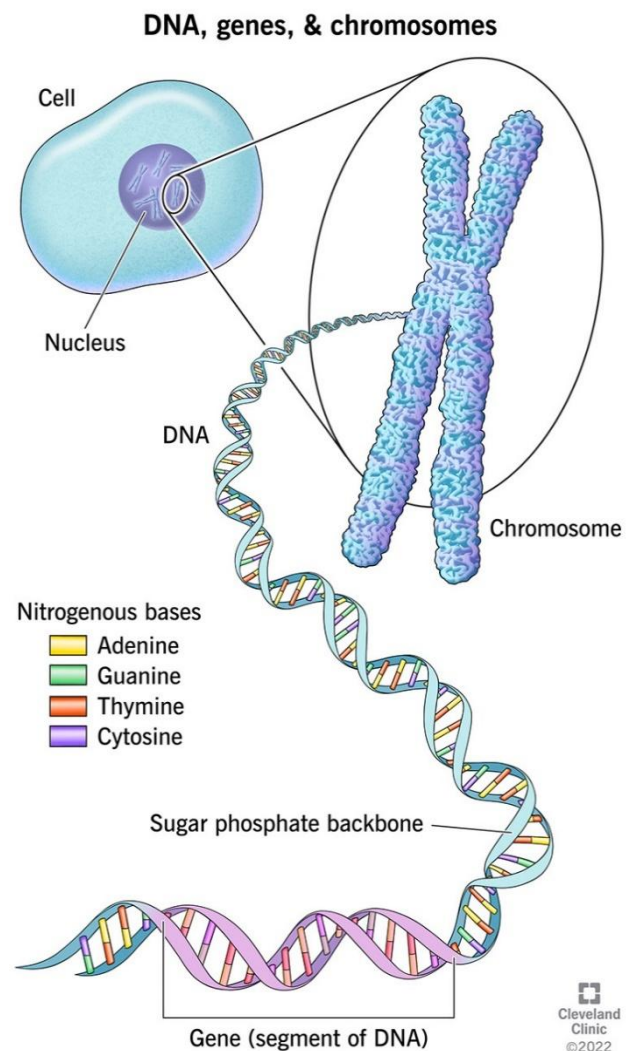
- ❑ JSP is among the hardest **combinatorial optimization** problems.
 - NP-hard problem
- ❑ Heuristic Method (Priority Rule)
 - Shortest processing time (SPT), earliest due date (EDD), etc.
 - Pros: easy to understand
 - Cons: **poor performance for complicated production line**
- ❑ Meta-Heuristic Algorithm (Tabu, Simulated Annealing, **Genetic Algorithm**)
 - Approximated-optimization approach
 - Pros: **provide a good solution efficiently**
 - Cons: cannot guarantee the global optimum
- ❑ Reinforcement Learning
 - Optimal control approach to take actions in a dynamic environment
 - Pros: consider decision over time for dynamic flexible job shop scheduling (DFJSS)
 - Cons: **convergence issue** in a large state space and action space
- ❑ Mathematical Programming
 - Optimization-based approach formulated by mixed integer programming
 - Pros: Guarantee global optimum
 - Cons: computational burden for large-scale problem (not suitable for **frequent rescheduling**)

Local optimum
Short running time

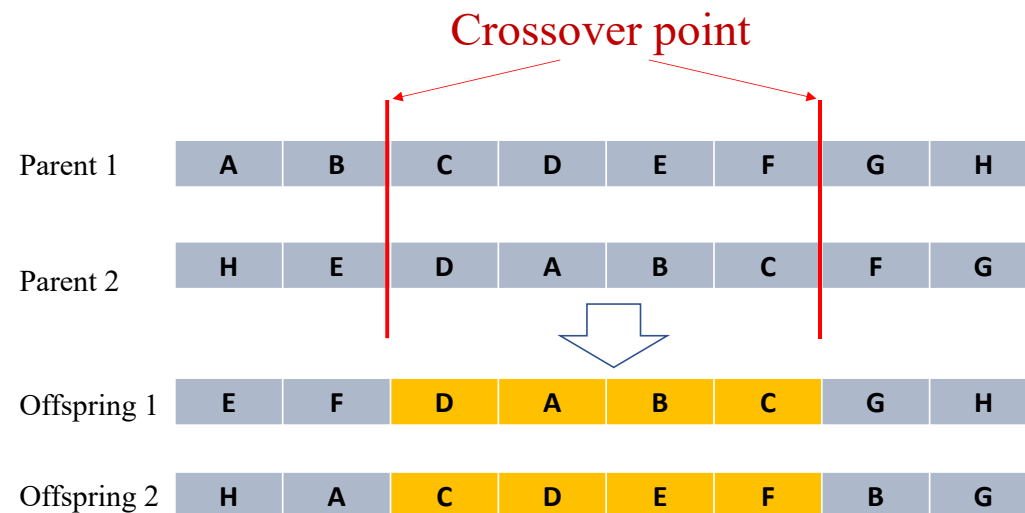


Global optimum
Long running time

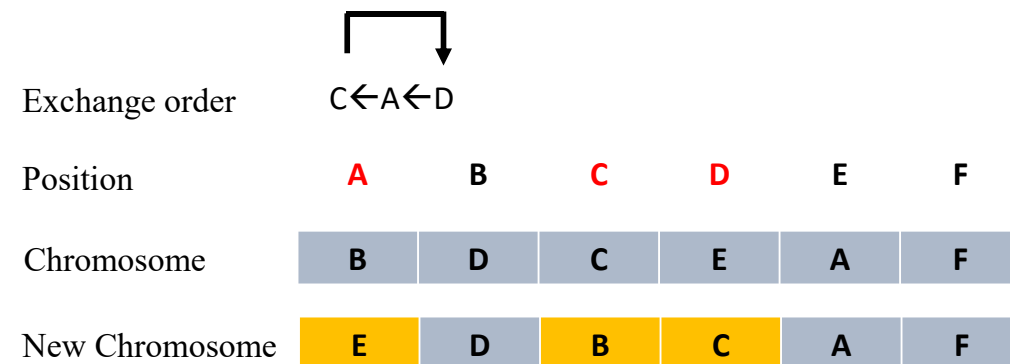
Genetic Algorithm (GA)



Two-point order crossover mechanism



Arbitrary multiple-point shift mutation mechanism



Holland, J. H. (1975). "Adaptation in Natural and Artificial Systems," University of Michigan Press, Ann Arbor.

How does RL guide GA?

Population Similarity

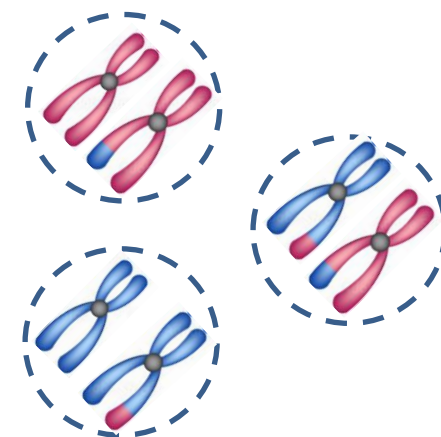
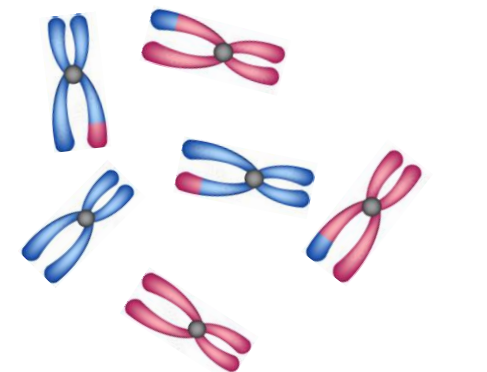
- Population similarity: cluster the population with DBSCAN, and then use the Spearman rank correlation coefficient to estimate the correlation within the groups and between the groups.
- Correlation: total - within group = between groups
- Population similarity as the correlation within the groups divided by the number of the clusters.
- Guide the mutation and crossover to balance exploration and exploitation.
 - if chromosomes appear similar, the population might be premature: need exploration and increase mutation and crossover rates
 - if chromosomes appear dissimilar, the population are not converged: need exploitation and decrease mutation and crossover rates.

Phase of Iterations

- Phase of iteration: the number of iterations that GA did not find the better solutions.

State Space in RL

- Discretize the two states by assigning quartiles to each value (i.e., the state space consists of the $4 \times 4 = 16$ states).

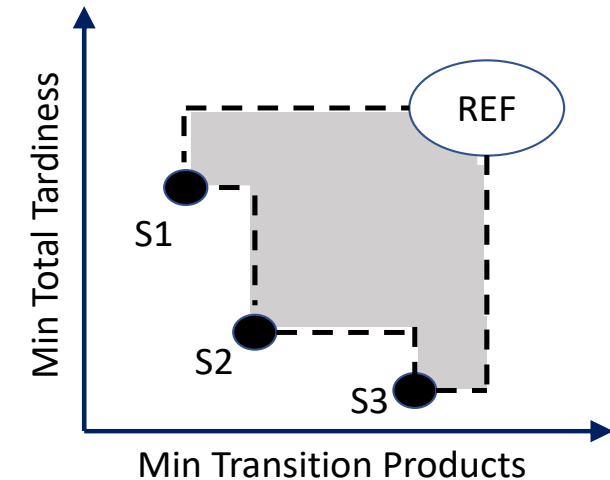


□ Action

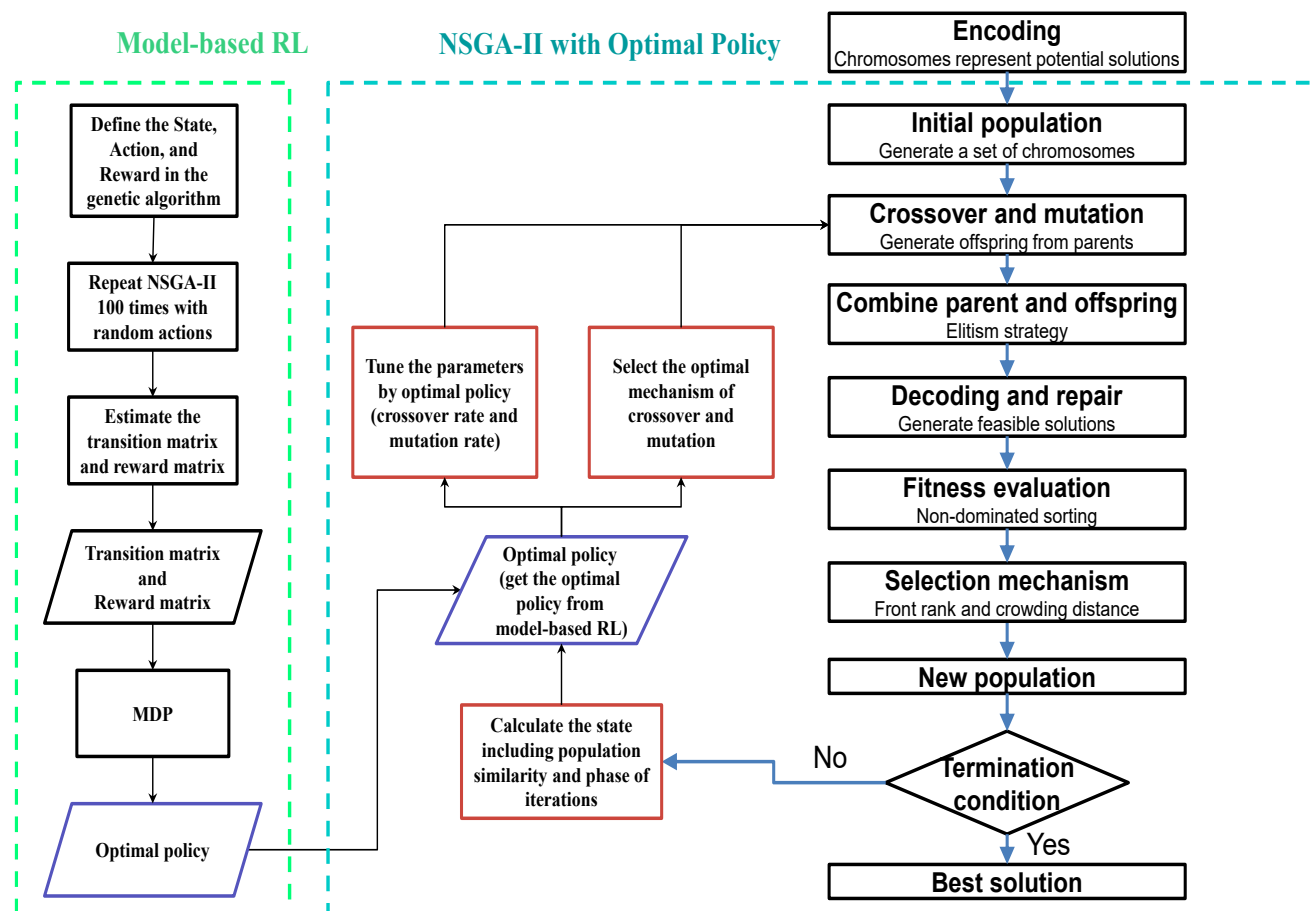
- Rate tuning and mechanism design related to crossover and mutation
- Rate tuning: “fix”, “increase”, and “decrease” by multiplying the original rate by 1, 1.02, and 0.98.
- Combining the crossover and mutation rates to generate $3 \times 3 = 9$ actions.
- Crossover mechanism: “one-point order crossover”, “two-point order crossover”, and “position-based order crossover”
- Mutation mechanism: “adjacent two-point change mutation”, “arbitrary multiple-point shift mutation”, and “shift change mutation”.
- Combining the crossover and mutation mechanisms to generate $3 \times 3 = 9$ actions

□ Reward

- Two-objective reward: minimization of (1) transition products and (2) total tardiness.
- Hypervolume as the volume surrounded by the solutions and a reference point (REF) (i.e., the poorest solution having the highest limit of each objective).
- Goal: maximize the hypervolume: If the present hypervolume indicator is better than last time, the reward is with +1; otherwise, if it is worse than last time, the reward is with -1.



GA embedded with RL (GAeRL)



Pseudocode of GAeRL

Begin

Model-based RL

While not maximal number of iterations

Repeat NSGA-II with random action

Collect data (state, action, reward)

End

Estimate the transition matrix and reward matrix

MDP (value iteration)

Obtain optimal policy (adjustment of mechanism or rate of crossover and mutation)

NSGAw/OP

Input: Optimal policy and initial parameters

Generate a population from EEH or generate a population randomly

While not termination condition

Generate offspring by crossover and mutation

Combine offspring and parent population

For each chromosome from offspring and parents

Compute the fitness function

Non-dominated sorting

Select chromosomes based on non-domination front rank and crowding distance

Build new population

Compute the state according to new population

Choose optimal policy according to the state

Tune rate or select mechanism of mutation and crossover

End

Decode to the petrochemical production schedule

End

Lee, C.-Y., Ho, C.-Y., Hung, Y.-H., and Deng, Y.-W., 2024. Multi-objective genetic algorithm embedded with reinforcement learning for petrochemical melt-flow-index production scheduling. *Applied Soft Computing*, 159, 111630.

▣ Petrochemical Factory

- Leading manufacturer & supplier of polypropylene in Taiwan
 - Product portfolio consists primarily of SBS, SIS, SEBS, and SEP, including compound materials for footwear, modified asphalt, waterproofing membranes, adhesives, and plastics modification.
- Data Source: manufacturing execution system (MES)
- Time: First half of 2019
- Data Size
 - 38 orders including 199 batches, 4 types of catalyst, 4 types of donors, and 12 precedence groups.
 - Transform the data for proprietary information protection without loss of generality.
- Results
 - Reduce **transition products** in the petrochemical production line by more than 10% through minimizing the **change of the Material Flow Index (MFI)**.
 - It ensures the **fulfillment of customer due dates**.



Empirical Study of Petrochemical Scheduling

□ State Space

- 16 states (combination of 4 levels of similarity and 4 levels in phase due to quantile discretization).

□ Action Space

- 9 actions (combination of 3 levels in crossover and 3 levels in mutation) with respect to rate tuning and mechanism selection, respectively.

□ Initial Parameters Settings

- Population size 20, crossover rate 0.8, and mutation rate 0.2.

Action ID	Crossover rate	Mutation rate
R1	Decrease	Decrease
R2	Fix	Decrease
R3	Increase	Decrease
R4	Decrease	Fix
R5 (baseline)	Fix	Fix
R6	Increase	Fix
R7	Decrease	Increase
R8	Fix	Increase
R9	Increase	Increase

Action ID	Crossover	Mutation
M1	One-point order crossover	Adjacent two-point change
M2	Two-point order crossover	Adjacent two-point change
M3	Position-based order crossover	Adjacent two-point change
M4	One-point order crossover	Arbitrary multiple-point shift
M5 (baseline)	Two-point order crossover	Arbitrary multiple-point shift
M6	Position-based order crossover	Arbitrary multiple-point shift
M7	One-point order crossover	Shift change mutation
M8	Two-point order crossover	Shift change mutation
M9	Position-based order crossover	Shift change mutation

Empirical Study of Petrochemical Scheduling

□ Performance Comparison (with 30 replications)

- **Six Algorithms:** (1) Engineering Experience Heuristic (EEH), (2) NSGA-II, (3) NSGA-II with random action (NSGAw/RA) for rate tuning of crossover and mutation, (4) NSGA-II with random action (NSGAw/RA) for mechanism selection of crossover and mutation, (5) NSGAeRL for rate tuning, (6) NSGAeRL for mechanism selection. (Note: NSGAw/RA has the same set of actions with NSGAeRL, but with equal probability of selecting actions rather than the optimal policy.)

State (optimal policy)	Similarity-1	Similarity-2	Similarity-3	Similarity-4	State (optimal policy)	Similarity-1	Similarity-2	Similarity-3	Similarity-4
Phase-1	R1	R4	R1	R4	Phase-1	M7	M6	M1	M6
Phase-2	R8	R8	R4	R9	Phase-2	M4	M8	M4	M2
Phase-3	R3	R6	R3	R1	Phase-3	M1	M4	M6	M8
Phase-4	R5	R4	R8	R4	Phase-4	M4	M5	M8	M8

Mean (Standard Deviation)	EEH	NSGA-II	NSGAwRA for Rate Tuning	NSGAwRA for Mechanism Selection	NSGAeRL for Rate Tuning	NSGAeRL for Mechanism Selection
Transition	5993	7247	7603	6974	6791	6517
Products	(0)	(950)	(1283)	(952)	(940)	(841)
Total	672	316	292	260	256	234
Tardiness	(0)	(98)	(78)	(44)	(70)	(38)
# of Iterations	1 (0)	1287 (490)	872 (463)	1597 (378)	1418 (489)	1767 (292)
CPU Time (second)	3 (0)	1122 (448)	924 (442)	1474 (337)	1466 (483)	1722 (280)

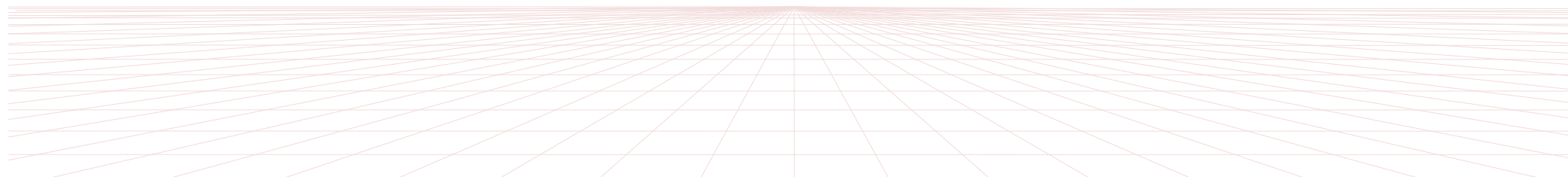
Optimization-Guided Learning (OGL)

- Genetic Algorithm embedded with Reinforcement Learning (GAeRL)

Lee, C.-Y., Ho, C.-Y., Hung, Y.-H., and Deng, Y.-W., 2024. Multi-objective genetic algorithm embedded with reinforcement learning for petrochemical melt-flow-index production scheduling. ***Applied Soft Computing***, 159, 111630.

- Reinforcement Learning embedded with Robust Optimization (RLeRO)

Lee, C.-Y., Huang, Y.-T., and Chen, P.-J., 2024. Robust-optimization-guiding deep reinforcement learning for chemical material production scheduling. ***Computers and Chemical Engineering***, 187, 108745.



□ Petrochemical Production Scheduling

● Uncertainty

- Demand fluctuation and yield rate
- Polyhedral uncertainty sets encode a budget of uncertainty into cardinality constraints.

● Objective function

- To maximize the gross profit of the chemical production schedule

$$\text{Max } \underbrace{\sum_{i \in I} \sum_{p \in P} V_i A_i^* x_{ip}}_{\text{Sales profit}} - \underbrace{\sum_{i \in I} \sum_{p \in P} C_i^S s_{ip}}_{\text{Inventory cost}} - \underbrace{\sum_{i \in I} C_i^L l_{ip}}_{\text{Stockout cost}} - \underbrace{\sum_{i \in I} \sum_{j \in I, j \neq i} \sum_{p \in P} C_{ij}^T z_{ijp}}_{\text{Transition cost}}$$

● Constraints

- State initialization for subproblems

$$s_{iP_0} = S_i^I, \forall i \in I; x_{if} = X_{ij}, \forall i \in I, f \in F \cup \{P_0\}$$

- Mass balance constraint

$$s_{ip} = s_{i(p-1)} + A_i^* x_{ip} - D_i^* + l_{ip}, \forall i \in I, p \in P$$

- Production transition identification

$$\sum_{i \in I} z_{ijp} = x_{jp}, \forall j \in I, p \in P$$

$$\sum_{j \in I} z_{ijp} = x_{i(p-1)}, \forall i \in I, p \in P$$

- Machine occupancy constraint

$$\sum_{i \in I} x_{ip} = 1, \forall p \in P$$

- Variable domains

$$x_{ip} \in \{0,1\}, \forall i \in I, p \in P \cup \{P_0\}$$

$$z_{ijp} \in \{0,1\}, \forall i \in I, j \in I, p \in P$$

$$s_{ip} \geq 0, \forall i \in I, p \in P \cup \{P_0\}$$

$$l_{ip} \geq 0, \forall i \in I, p \in P$$

□ Action

- Network output discrete probability distribution A
- Action a is sampled from A , corresponding to x_{ip} .

□ Episode

- Finish a complete scheduling window and rolling to the next.

□ Reward

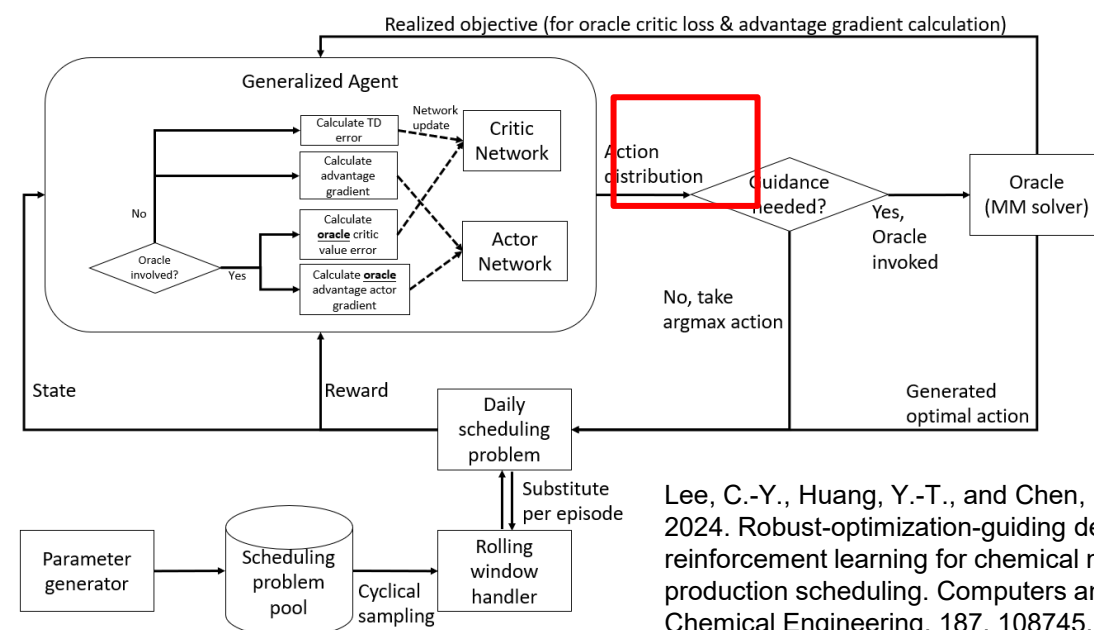
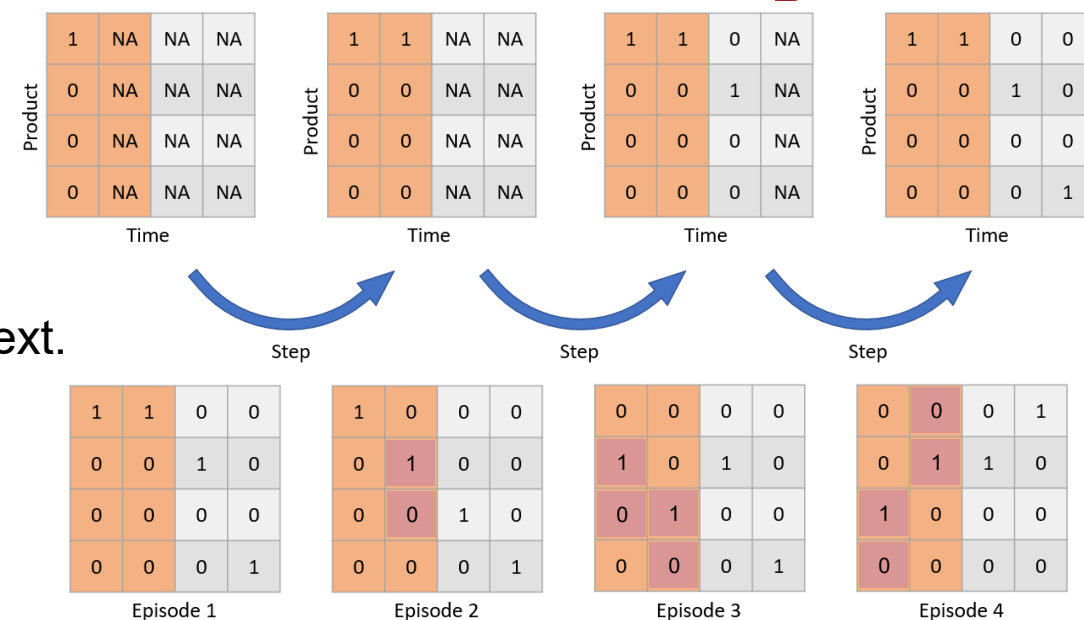
- The **change** in objective value after a particular action

□ State encoding

$$state_p = (\underbrace{l_{ip}}_{\text{Inventory level}}, \underbrace{x_{ip}}_{\text{Determined schedule}}, \underbrace{\tilde{D}_{ip}}_{\text{Predicted demand}}, \underbrace{\tilde{A}_i x_{ip} + l_{ip} - \tilde{D}_{ip}}_{\text{Estimated stockout}}, \underbrace{t}_{\text{Time counter}})$$

□ Methods

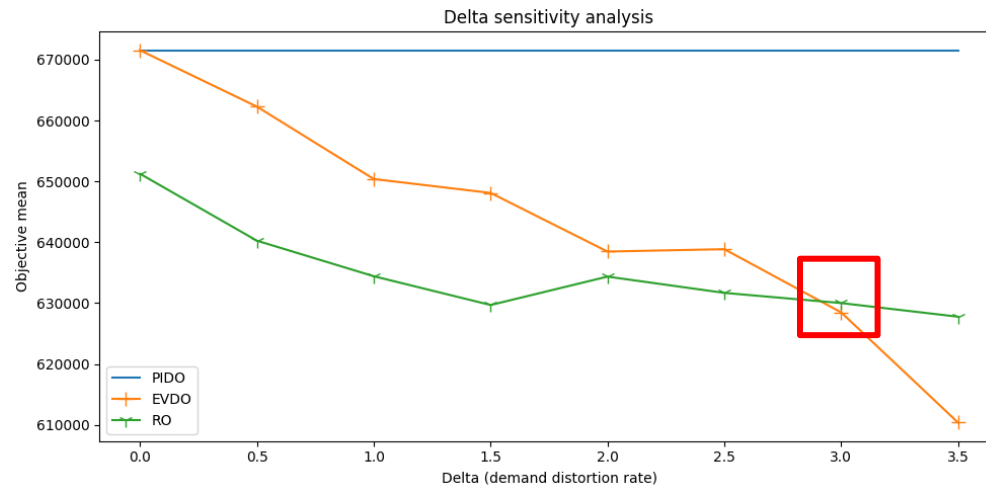
- Perfect information deterministic optimization (PIDO)
- Expected value deterministic optimization (EVDO)
- Robust optimization (RO)
- Advantage-Actor-Critic (A2C)
- A2C + EVDO guiding
- A2C + RO guiding



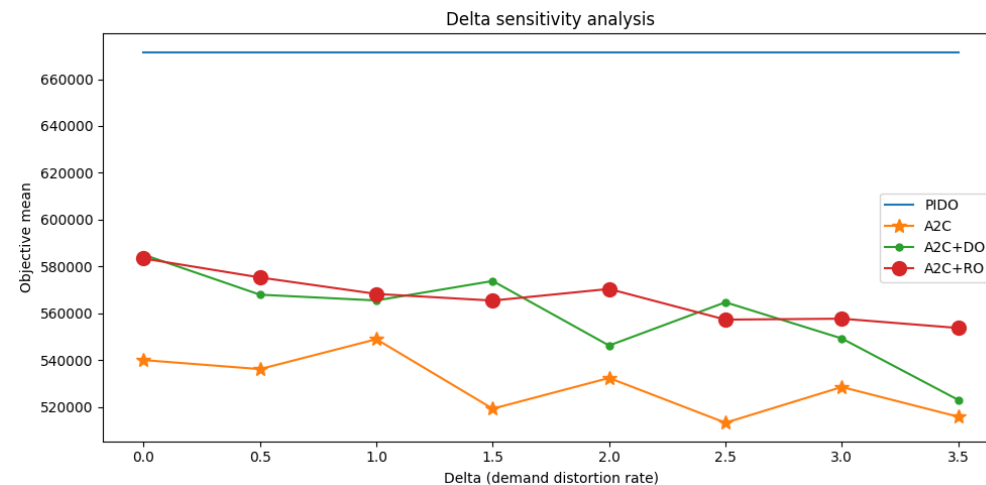
Lee, C.-Y., Huang, Y.-T., and Chen, P.-J., 2024. Robust-optimization-guiding deep reinforcement learning for chemical material production scheduling. Computers and Chemical Engineering, 187, 108745.

□ Sensitivity Analysis

● Optimization-based models



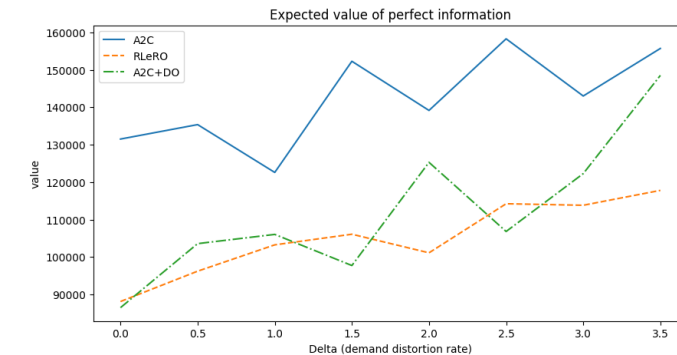
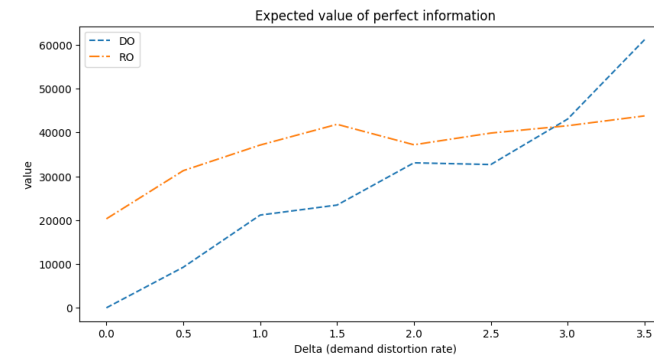
● RL-based models



□ Solution Value Analysis

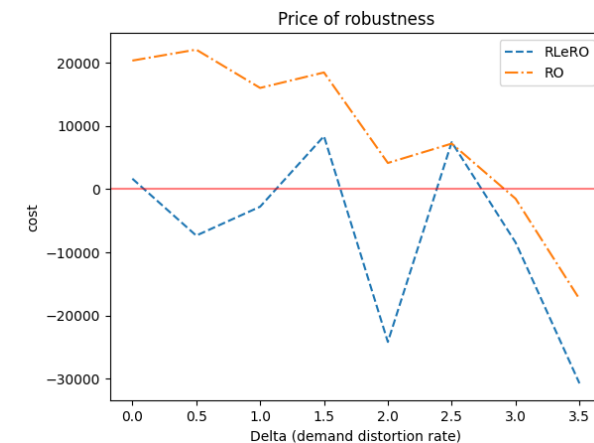
● Expected value of perfect information (EVPI)

- “How much a decision-maker would be willing to pay for perfect information when using the model”
- subtract the PIDO value from the target model’s
- The robust models are less needed of perfect infor.



● Price of robustness

- Distance of objective between the baseline (EVDO) and robust solution
- “How much it cost to apply robust solutions”
- Adopt conservative policies in high demand fluctuation.



I-Ching teaches people how to address the change

1. Error/MSE focus
2. Data-driven
3. Causality
4. Find the change in unchanging env. *Predict-then-Optimize*
- 在不變中找變 (infor. content)

1. Objective/KPI focus
2. Decision-oriented
3. Resource Allocation
4. Find the unchange in changing env.
- 在變中找不變 (robustness)

Demand
& Price
Forecast

Predictive
Analytics
(Environ.)

Train/Guide

Reflection

Prescriptive
Analytics
(Human)

Resource
Portfolio
Optimization

Closed Loop Model
Optimization-Guided Learning

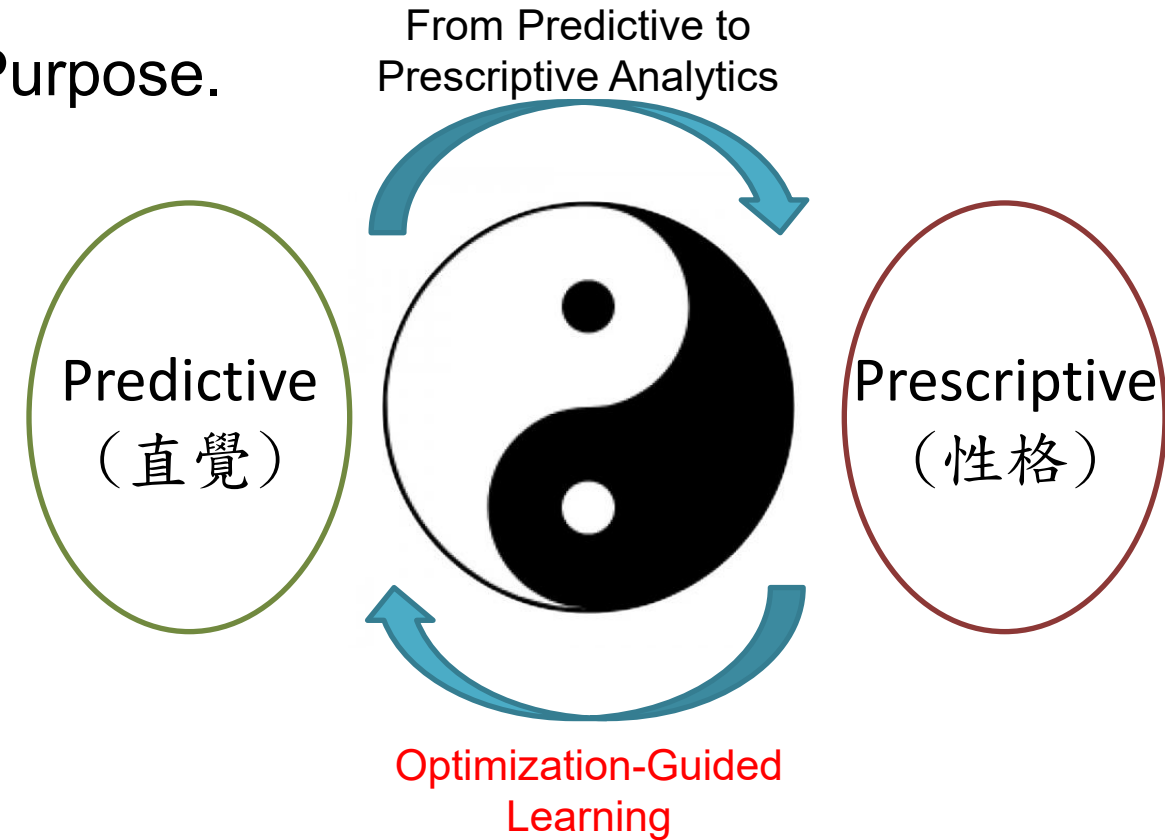


□ Prediction is the Process; Decision is the Purpose.

□ From Predictive to Prescriptive Analytics

□ Optimization-Guided Learning (OGL)

- Find the “**unchanged**” power in “**changing**” env.
- Make the learning system more “**stable**”.



We can observe them, but cannot change them.
我們只能觀察到，但不能改變他們

Light speed in vacuum (真空光速): 299,792,458 m/s
Planck constant (普朗克常數): $6.62607015 \times 10^{-34}$ J·s
Electron mass (電子質量): $9.10938291 \times 10^{-31}$ kg
Avogadro constant (亞佛加厥常數): $6.02214076 \times 10^{23}$ mol⁻¹
Boltzmann constant (波茲曼常數): $1.38064852 \times 10^{-23}$ J/K
Gravitational constant (重力常數): 6.67384×10^{-11} m³/(kg · s²)

Thanks for your attention!



Dept. of Information Management,
National Taiwan University
Name: Chia-Yen Lee
Phone: +886-2-33661206
Email: chiayenlee@ntu.edu.tw
Web: <http://polab.im.ntu.edu.tw/>