Data-driven optimization of processes with degrading equipment

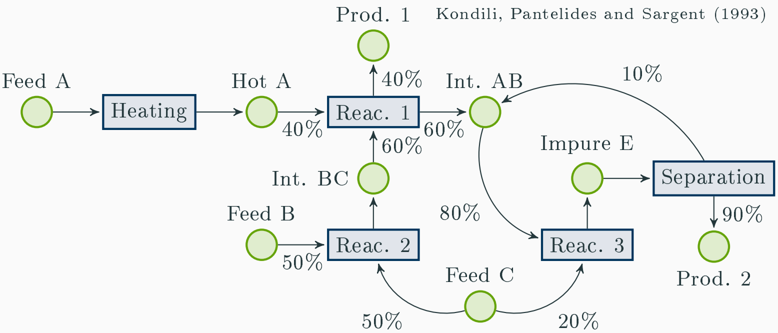
Johannes Wiebe[1], Inês Cecílio[2], Ruth Misener[1]

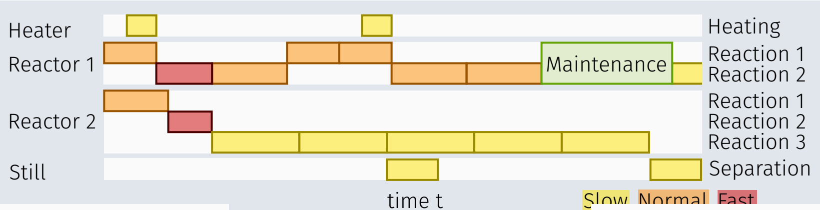
Tuesday 11th September, 2018

[1]Department of Computing, Imperial College London, London, UK

[2]Schlumberger Research Cambridge, Cambridge, UK London

Motivation: Why degradation matters





Starting point: Process level MI(N)LP model

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{m}} \quad & \text{cost}(\boldsymbol{x}, \boldsymbol{m} \quad) \\
\text{s.t.} \quad & \text{process model}(\boldsymbol{x}, \boldsymbol{m} \quad) && \text{(eg. balance equations)} \\
& \text{maintenance model}(\boldsymbol{x}, \boldsymbol{m} \quad) && \text{(eg. types of maint.)}
\end{aligned}$$

where $\boldsymbol{x}$ are process variables, $\boldsymbol{m}$ are maintenance variables

Starting point: Process level MI(N)LP model

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}} \quad & \text{cost}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}) & \\
\text{s.t.} \quad & \text{process model}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}) & \text{(eg. balance equations)} \\
& \text{maintenance model}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}) & \text{(eg. types of maint.)} \\
& \text{health model}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}), & \text{(eq. prognosis model)}
\end{aligned}
$$

where $\boldsymbol{x}$ are process variables, $\boldsymbol{m}$ are maintenance variables, and $\boldsymbol{h}$ are health related variables.

Starting point: Process level MI(N)LP model

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}} \quad & \text{cost}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}) & \\
\text{s.t.} \quad & \text{process model}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}) & \text{(eg. balance equations)} \\
& \text{maintenance model}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}) & \text{(eg. types of maint.)} \\
& \text{health model}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}), & \text{(eq. prognosis model)}
\end{aligned}
$$

where $\boldsymbol{x}$ are process variables, $\boldsymbol{m}$ are maintenance variables, and $\boldsymbol{h}$ are health related variables.

Related Work

Vassiliadis and Pistikopoulos (2001); Liu, Yahia and Papageorgiou (2014); Xenos, et int, Thornhill (2016); Aguirre and Papageorgiou (2018); Biondi, Sand and Harjunkoski (2017); Yildirim, Gebraeel and Sun (2017); Başçiftci, Ahmed, Gebraeel and Yildirim (2018)

Starting point: Process level MI(N)LP model

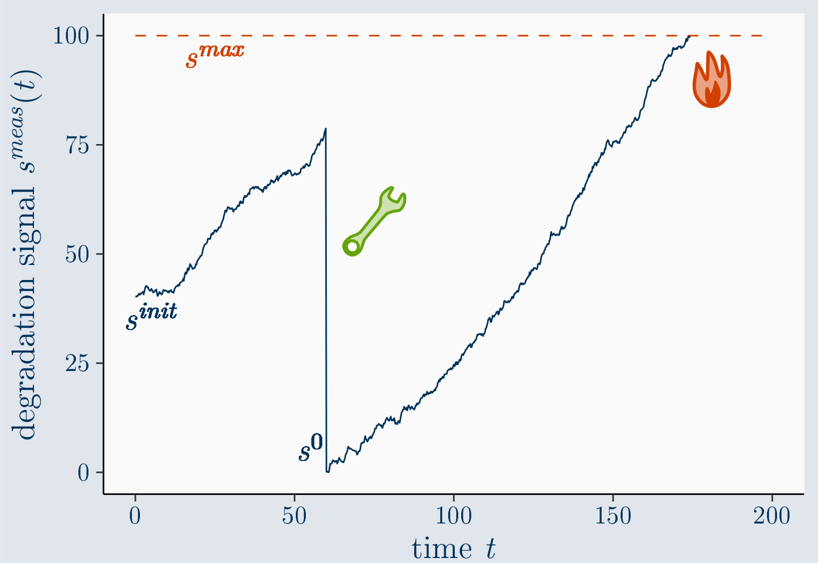
$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}} \quad & \text{cost}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}) & \\
\text{s.t.} \quad & \text{process model}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}) & \text{(eg. balance equations)} \\
& \text{maintenance model}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}) & \text{(eg. types of maint.)} \\
& \text{health model}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}), & \text{(eq. prognosis model)}
\end{aligned}
$$

where $\boldsymbol{x}$ are process variables, $\boldsymbol{m}$ are maintenance variables, and $\boldsymbol{h}$ are health related variables.

Idea

Combine process level MI(N)LP scheduling & planning with more sophisticated (stochastic) degradation modelling and robust optimization.

What is degradation modelling?



What is degradation modelling?

The degradation signal $s^{meas}(t)$ can be modelled by a stochastic process :

$$S(t) = \{S_t : t \in T\},$$

where S_t is a random variable (Alaswad and Xiang, 2017).

What is degradation modelling?

The degradation signal $s^{meas}(t)$ can be modelled by a stochastic process :

$$S(t) = \{S_t : t \in T\},$$

where S_t is a random variable (Alaswad and Xiang, 2017).

Often used: Lévy type processes (Applebaum, 2004)

- Independent increments: $S_{t_2} - S_{t_1}, ..., S_{t_n} - S_{t_{n-1}}$ are independent for any $0 < t_1 < t_2 < ... < t_n < \infty$
- Stationary increments: $S_t - S_s$ and S_{t-s} have the same distribution for any $s < t$

What is degradation modelling?

The degradation signal $s^{meas}(t)$ can be modelled by a stochastic process :

$$S(t) = \{S_t : t \in T\},$$

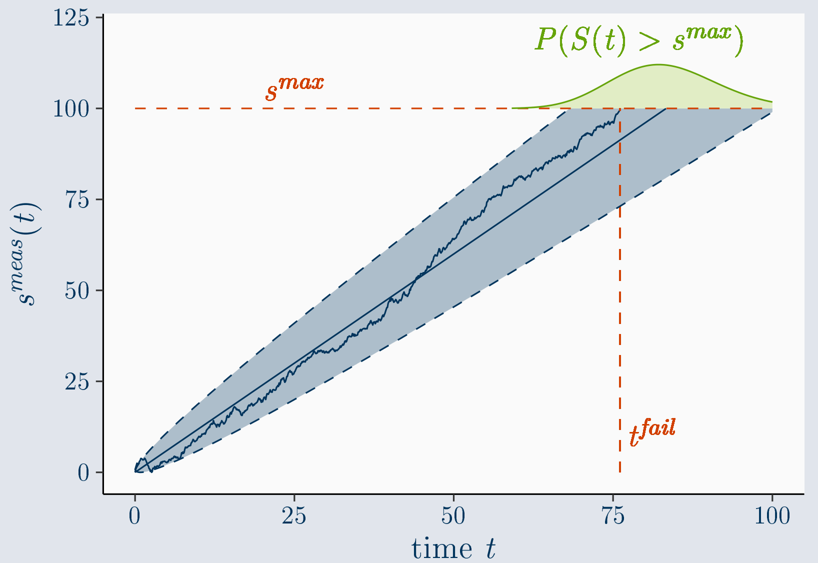
where S_t is a random variable (Alaswad and Xiang, 2017).

Often used: Lévy type processes (Applebaum, 2004)

- Independent increments: $S_{t_2} - S_{t_1}, \dots, S_{t_n} - S_{t_{n-1}}$ are independent for any $0 < t_1 < t_2 < \dots < t_n < \infty$
- Stationary increments: $S_t - S_s$ and S_{t-s} have the same distribution for any $s < t$

Therefore $S_t - S_{t-\Delta t} = D \sim \mathcal{D}(\mathbf{\Theta}, \Delta t)$, where $\mathbf{\Theta}$ are parameters of distribution $\mathcal{D}$.

Calculating failure probabilities



A health model based on Lévy processes

Assumption

The health of each unit j can be described by a Lévy process $S_j(t)$ with increments $S_{j,t} - S_{j,t-\Delta t} = D_j \sim \mathcal{D}_j(\mathbf{\Theta}, \Delta t)$.

A health model based on Lévy processes

Assumption

The health of each unit j can be described by a Lévy process $S_j(t)$ with increments $S_{j,t} - S_{j,t-\Delta t} = D_j \sim \mathcal{D}_j(\mathbf{\Theta}, \Delta t)$.

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}} \quad & \text{cost}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}) \\
\text{s.t.} \quad & \text{process model}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}) \\
& \text{maintenance model}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}) \\
& S_{j,t} \leq s_j^{max} && \forall t, j \in J \\
& S_{j,t} = \begin{cases} S_{j,t-1} + D_j & , \quad \text{if } m_{j,t} = 0 \\ s_j^0, & \quad \text{otherwise} \end{cases} && \forall t, j \in J
\end{aligned}
$$

where $m_{j,t} = 1$ if maintenance is performed on unit j at time t.

Accounting for effects of process variables

Assumption (Liao and Tian, 2013)

All relevant operating variables are piecewise constant – i.e. the process has a set of discrete operating modes $k \in K$.

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}} \quad & \text{cost}(\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}) \\
\text{s.t.} \quad & \text{process model}(\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}) \\
& \text{maintenance model}(\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}) \\
& S_{j,t} \leq s_j^{max} && \forall t, j \in J \\
& S_{j,t} = \begin{cases} S_{j,t-1} + \sum_{k \in \mathcal{K}} x_{j,k,t} \cdot D_{j,k}, & \text{if } m_{j,t} = 0 \\ s_j^0, & \text{otherwise} \end{cases} && \forall t, j \in J
\end{aligned}
$$

where $x_{j,k,t} = 1$ if unit j operates in mode k at time t.

Deriving a robust counterpart (Lappas and Gounaris, 2016)

Idea

Replace random variables $D_{j,k}$ and $S_{j,t}$ by uncertain parameter $\tilde{d}_{j,k} \in \mathcal{U}$ and second stage variable $s_{j,t}(\tilde{d}_{j,k})$.

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}} \quad & \text{cost}(\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}) \\
\text{s.t.} \quad & \text{process model}(\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}) \\
& \text{maintenance model}(\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}) \\
& s_{j,t}(\tilde{d}_{j,k}) \leq s_j^{max} && \forall t, j \in J \\
& s_{j,t} = \begin{cases} s_{j,t-1} + \sum_{k \in \mathcal{K}} x_{j,k,t} \cdot \tilde{d}_{j,k}\,, & \text{if } m_{j,t} = 0 \\ s_j^0\,, & \text{otherwise} \end{cases} && \forall t, j \in J
\end{aligned}
$$

$\forall \tilde{d}_{j,k} \in \mathcal{U}$.

Deriving a robust counterpart (Lappas and Gounaris, 2016)

Idea

Replace random variables $D_{j,k}$ and $S_{j,t}$ by uncertain parameter $\tilde{d}_{j,k} \in \mathcal{U}$ and second stage variable $s_{j,t}(\tilde{d}_{j,k})$.

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}} \quad & \text{cost}(\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}) \\
\text{s.t.} \quad & \text{process model}(\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}) \\
& \text{maintenance model}(\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}) \\
& s_{j,t}(\tilde{d}_{j,k}) \leq s_j^{max} & \forall t, j \in J \\
& s_{j,t} = \begin{cases} s_{j,t-1} + \sum_{k \in \mathcal{K}} x_{j,k,t} \cdot \tilde{d}_{j,k}\,, & \text{if } m_{j,t} = 0 \\ s_j^0\,, & \text{otherwise} \end{cases} & \forall t, j \in J
\end{aligned}
$$

$\forall \tilde{d}_{j,k} \in \mathcal{U}$. Approximate $s_{j,t}(\tilde{d}_{j,k})$ by linear decision rule. Utilize Robust Optimization reformulation techniques.

Deriving a robust counterpart (Lappas and Gounaris, 2016)

In special cases

Solve an approximate deterministic model with an order of magnitude fewer variables/constraints.

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}} \quad & \text{cost}(\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}) \\
\text{s.t.} \quad & \text{process model}(\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}) \\
& \text{maintenance model}(\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}) \\
& s_{j,t}(\tilde{d}_{j,k}) \leq s_j^{max} & \forall t, j \in J \\
& s_{j,t} = \begin{cases} s_{j,t-1} + \sum_{k \in \mathcal{K}} x_{j,k,t} \cdot \tilde{d}_{j,k}\,, & \text{if } m_{j,t} = 0 \\ s_j^0\,, & \text{otherwise} \end{cases} & \forall t, j \in J
\end{aligned}
$$

$\forall \tilde{d}_{j,k} \in \mathcal{U}$. Approximate $s_{j,t}(\tilde{d}_{j,k})$ by linear decision rule. Utilize Robust Optimization reformulation techniques.

How do we choose $\mathcal{U}$?

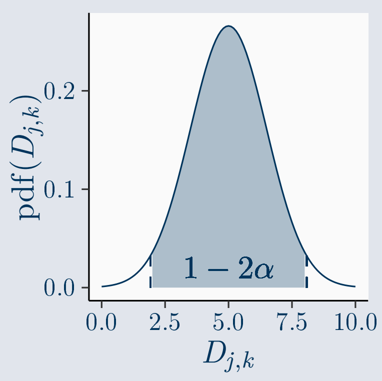
Assumption: $\mathcal{U}$ is a box uncertainty set

$$\mathcal{U} = \{\tilde{d}_{j,k} | \bar{d}_{j,k}(1 - \epsilon_{j,k}) \leq \tilde{d}_{j,k} \leq \bar{d}_{j,k}(1 + \epsilon_{j,k})\}$$

How do we choose $\mathcal{U}$?

Assumption: $\mathcal{U}$ is a box uncertainty set

$$\mathcal{U} = \{\tilde{d}_{j,k} | \bar{d}_{j,k}(1 - \epsilon_{j,k}) \leq \tilde{d}_{j,k} \leq \bar{d}_{j,k}(1 + \epsilon_{j,k})\}$$

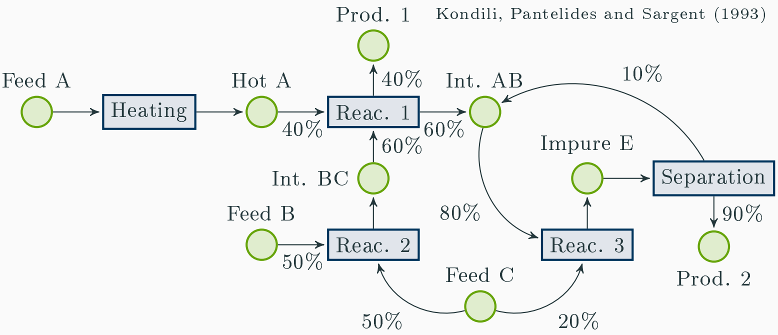


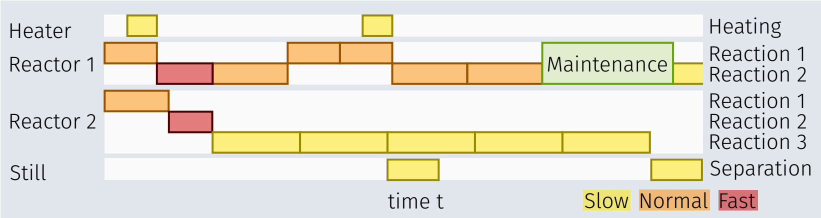
Choose $\epsilon_{j,k}$ from distribution $\mathcal{D}_{j,k}$ (Ning and You, 2017):

$$\epsilon_{j,k} = 1 - F^{-1}(\alpha)/\bar{d}_{j,k}$$

Size of $\mathcal{U}$ depends on a single parameter α!

Case study: State-Task-Network (Kondili et al., 1993)

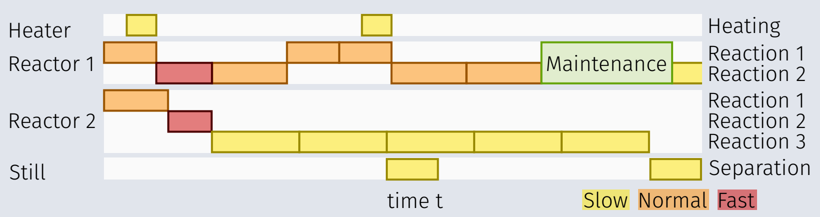




Case study: State-Task-Network (Kondili et al., 1993)

Biondi, Sand and Harjunkoski (2017) extend the STN to include...

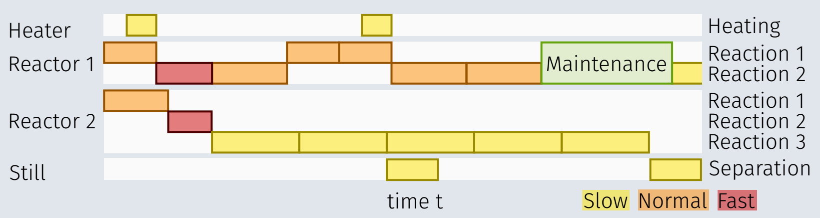
- ...unit health and maintenance scheduling,
- ...integrated scheduling and planning,
- ...multiple operating modes per task.



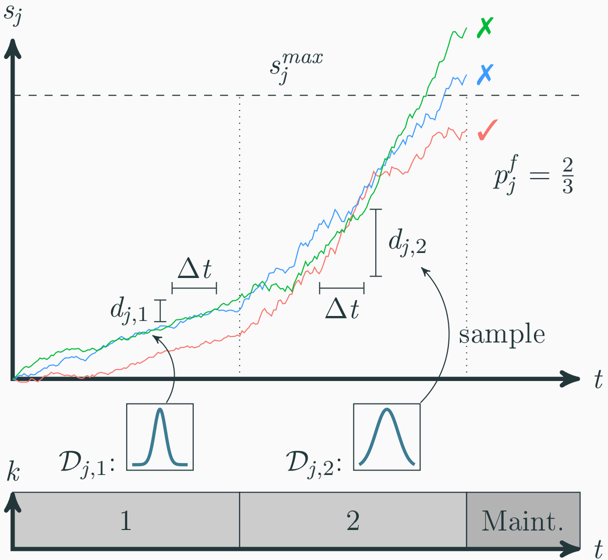
Case study: State-Task-Network (Kondili et al., 1993)

This work. . .

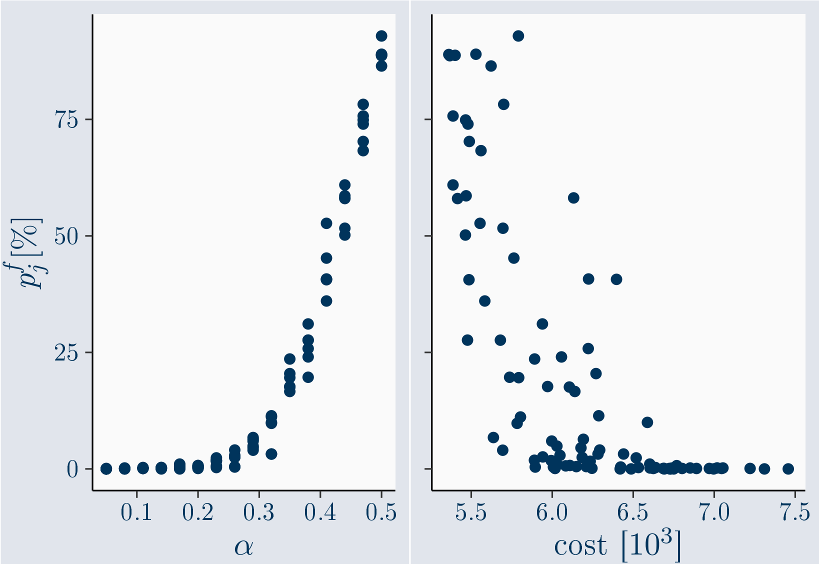
- . . . replaces their deterministic health model by the proposed approach based on degradation modelling,
- . . . utilizes robust optimization to obtain a solution that is likely to remain feasible,
- . . . analyzes the price of robustness.



Evaluating solution robustness



The price of robustness



Choosing α is its own optimization problem

We optimize α by solving

$$\min_{\alpha} c^*(\alpha) + \sum_j p_j^f(\alpha) \cdot c_j^f$$

- $c^*(\alpha)$ is the objective value of a MILP solution given α.
- $p_j^f(\alpha)$ is the corresponding probability of failure (of unit j).
- c_j^f is the cost of an unexpected failure.

Alternative objective: Li and Li (2015)

Choosing α is its own optimization problem

We optimize α by solving

$$\min_{\alpha} c^*(\alpha) + \sum_j p_j^f(\alpha) \cdot c_j^f$$

- $c^*(\alpha)$ is the objective value of a MILP solution given α.
- $p_j^f(\alpha)$ is the corresponding probability of failure (of unit j).
- c_j^f is the cost of an unexpected failure.

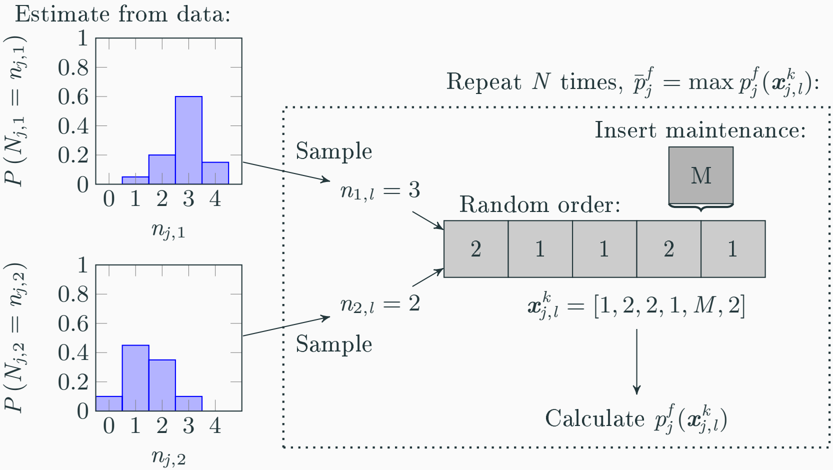
Alternative objective: Li and Li (2015)

Idea: Use Bayesian Optimization (BO)

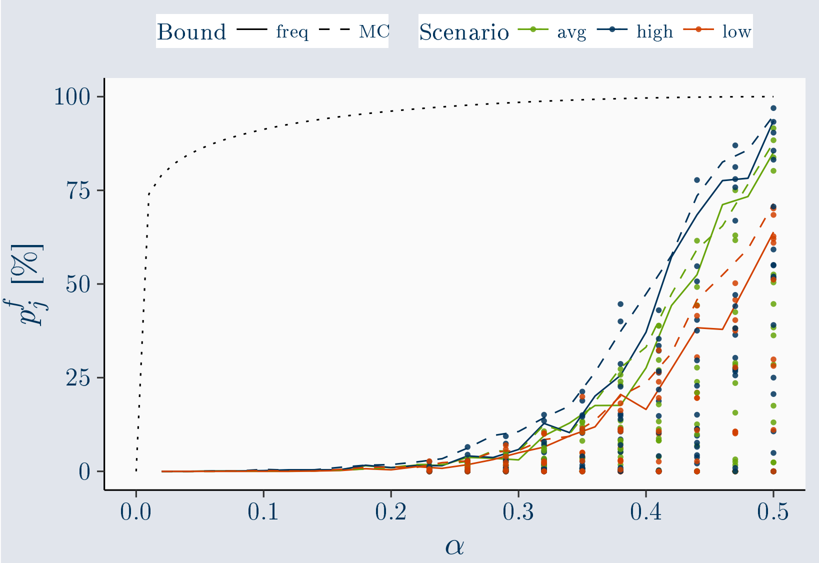
Both c^* and p_j^f can be viewed as expensive black box functions. BO is very suitable for this setting (Jones et al., 1998).

Saving time: data-driven approximations

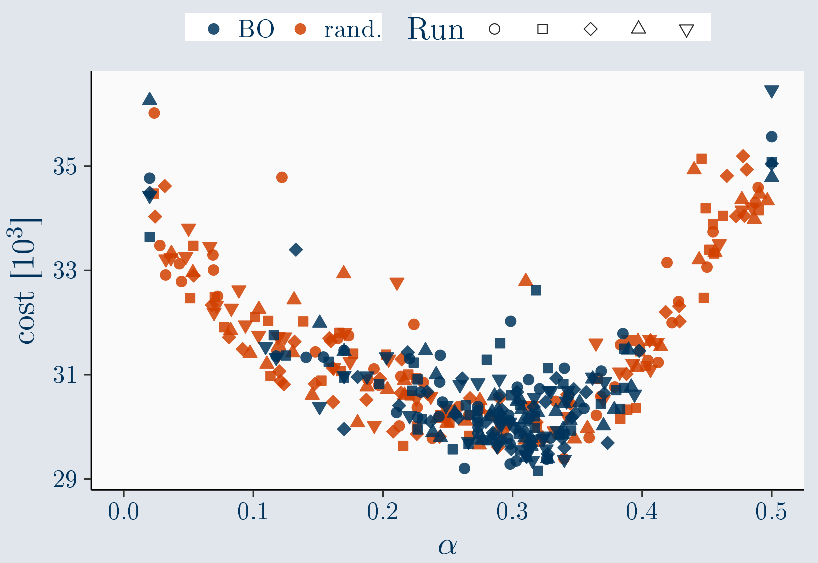
An upper bound on the probability of failure p_j^f can be estimated from data (using logistic regression).



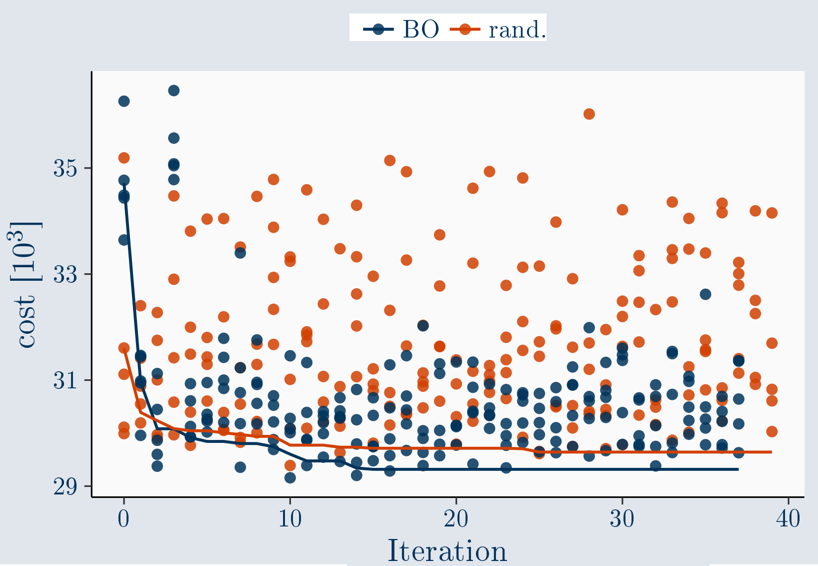
Saving time: data-driven approximations



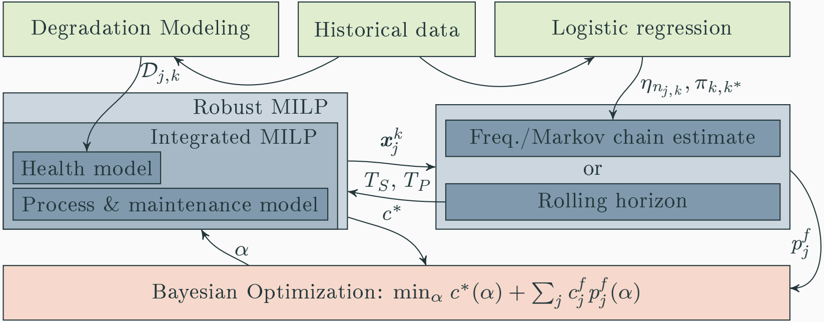
Bayesian Optimization



Bayesian Optimization

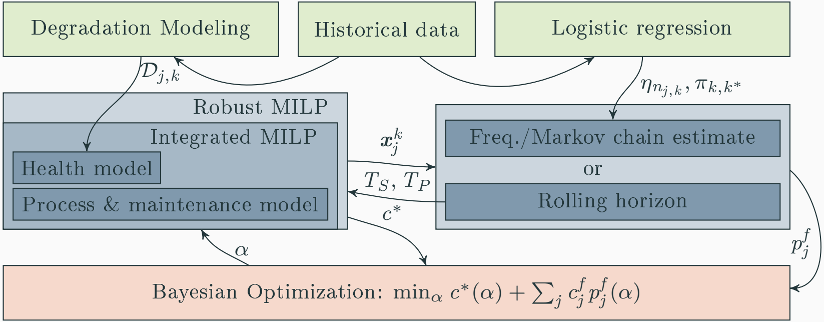


Conclusion



Degradation Modeling
Historical data
Logistic regression
$\mathcal{D}_{j,k}$
$\eta_{n_{j,k}}, \pi_{k,k^*}$
Robust MILP
Integrated MILP
Health model
Process & maintenance model
$\boldsymbol{x}_j^k$
T_S, T_P
c^*
Freq./Markov chain estimate
or
Rolling horizon
p_j^f
α
Bayesian Optimization: $\min_\alpha c^*(\alpha) + \sum_j c_j^f p_j^f(\alpha)$

Conclusion



Thank You!

Funding: EP/L016796/1, EP/R511961/1 no. 17000145, and EP/P016871/1

Imperial College London

computational optimisation group

HiPEDS Schlumberger

References

Aguirre, A. M. and Papageorgiou, L. G. (2018). Medium-term optimization-based approach for the integration of production planning, scheduling and maintenance. Computers and Chemical Engineering, 0:1–21.

Alaswad, S. and Xiang, Y. (2017). A review on condition-based maintenance optimization models for stochastically deteriorating system. Reliability Engineering & System Safety, 157:54–63.

Applebaum, D. (2004). Lévy processes-from probability to finance and quantum groups. Notices of the American Mathematical Society, 51(11):1336–1347.

Başçiftci, B., Ahmed, S., Gebraeel, N. Z., and Yildirim, M. (2018). Stochastic Optimization of Maintenance and Operations Schedules under Unexpected Failures. IEEE Transactions on Power Systems, 8950(c):1–1.

Biondi, M., Sand, G., and Harjunkoski, I. (2017). Optimization of multipurpose process plant operations: A multi-time-scale maintenance and production scheduling approach. Computers and Chemical Engineering, 99:325–339.

Ierapetritou, M. G. and Floudas, C. A. (1998). Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes. Industrial & Engineering Chemistry Research, 37(11):4341–4359.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient Global Optimization of Expensive Black-Box Functions. Journal of Global Optimization, 13:455–492.

Karimi, I. A. and McDonald, C. M. (1997). Planning and Scheduling of Parallel Semicontinuous Processes. 2. Short-Term Scheduling. Industrial & Engineering Chemistry Research, 36(7):2701–2714.

Kondili, E., Pantelides, C., and Sargent, R. (1993). A general algorithm for short-term scheduling of batch operations - I. MILP formulation. Computers and Chemical Engineering, 17(2):211–227.

Lappas, N. H. and Gounaris, C. E. (2016). Multi-stage adjustable robust optimization for process scheduling under uncertainty. AIChE Journal, 62(5):1646–1667.

Li, Z. and Li, Z. (2015). Optimal robust optimization approximation for chance constrained optimization problem. Computers and Chemical Engineering, 74:89–99.

Liao, H. and Tian, Z. (2013). A framework for predicting the remaining useful life of a single unit under time-varying operating conditions. IIE Transactions, 45(9):964–980.

Liu, S., Yahia, A., and Papageorgiou, L. G. (2014). Optimal Production and Maintenance Planning of Biopharmaceutical Manufacturing under Performance Decay. Industrial & Engineering Chemistry Research, 53(44):17075–17091.

Maravelias, C. T. and Grossmann, I. E. (2003). New general continuous-time state - Task network formulation for short-term scheduling of multipurpose batch plants. Industrial & Engineering Chemistry Research, 42(13):3056–3074.

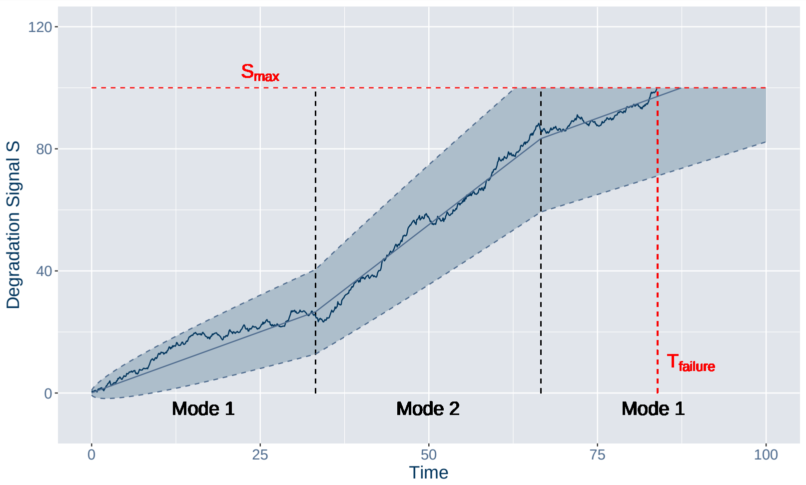
Ning, C. and You, F. (2017). A data-driven multistage adaptive robust optimization framework for planning and scheduling under uncertainty. AIChE Journal, 63(10):4343–4369.

Vassiliadis, C. and Pistikopoulos, E. (2001). Maintenance scheduling and process optimization under uncertainty. Computers and Chemical Engineering, 25(2-3):217–236.

Xenos, D. P., Kopanos, G. M., Cicciotti, M., and Thornhill, N. F. (2016). Operational optimization of networks of compressors considering condition-based maintenance. Computers and Chemical Engineering, 84:117–131.

Yildirim, M., Gebraeel, N. Z., and Sun, X. A. (2017). Integrated Predictive Analytics and Optimization for Opportunistic Maintenance and Operations in Wind Farms. IEEE Transactions on Power Systems, 32(6):4319–4328.

Degradation modelling with multiple operating modes



Saving time: a deterministic approximation

Assumption

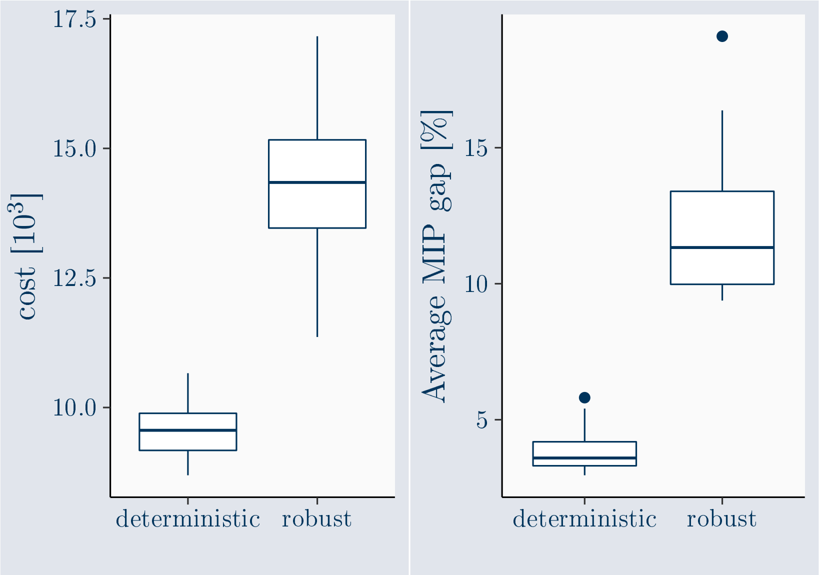
Only the health model depends on $\tilde{d}_{j,k}$ and $\tilde{d}_{j,k} \geq 0$.

Then we can prove that a solution to

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}} \quad & \text{cost}(\boldsymbol{x}, \boldsymbol{m}, \cancel{\boldsymbol{h}}) \\
\text{s.t.} \quad & \text{process model, maint. model}(\boldsymbol{x}, \boldsymbol{m}, \cancel{\boldsymbol{h}}) \\
& s_{j,t} \leq s_j^{max} && \forall t, j \in J \\
& s_{j,t} = \begin{cases} s_{j,t-1} + \sum_{k \in \mathcal{K}} x_{j,k,t} \cdot d_{j,k}^{max}, & \text{if } m_{j,t} = 0 \\ s_j^0, & \text{otherwise} \end{cases} && \forall t, j \in J
\end{aligned}
$$

with $d_{j,k}^{max} = \max_{\mathcal{U}} \tilde{d}_{j,k}$ is also feasible in the robust problem.

Saving time: a deterministic approximation



How does robust optimization work?

General idea

- Make constraints hold for all values in $\mathcal{U}$: $\sum_j \tilde{a}_{ij} x_j \leq b_i, \forall \tilde{a}_{ij} \in \mathcal{U}$
- Reformulate semi-infinite constraint: $\sum_j a_{ij} x_j + \text{protection}\,(\mathcal{U}) \leq b_i$
- How do we choose the right protection level?

Example: Soyster's method (worst case) [1973]

$$\begin{aligned} \max_{x_1, x_2} \quad & x_1 + x_2 \\ \text{s.t.} \quad & \tilde{a}_{11} x_1 + \tilde{a}_{12} x_2 \leq b_1, \\ & \forall \tilde{a}_{ij} \in \mathcal{U} \end{aligned}$$

$$\begin{aligned} \max_{x_1, x_2} \quad & x_1 + x_2 \\ \text{s.t.} \quad & a_{11} x_1 + a_{12} x_2 + \sum_j \hat{a}_{ij} |x_j| \leq b_1 \end{aligned}$$

Given: $[a_{11}, a_{12}] = [1, 2], [\hat{a}_{11}, \hat{a}_{12}] = [0.1, 0.2], [b_1] = [2]$

Formulation

Scheduling

$$M_{j,t}S_{j,0} \leq S_{j,t} \leq S_{j,max} + M_{j,t} \cdot (S_{j,0} - S_{j,max}) \qquad \forall t, j \in J, D \in \mathcal{D}$$

$$S_{j,t} \geq S_{j,t-\Delta t} + \sum_k Z_{j,k,t}D_{j,k,t} + M_{j,t} \cdot (S_{j,0} - S_{j,max}) \qquad \forall t, j \in J, D \in \mathcal{D}$$

$$S_{j,t} \leq S_{j,t-\Delta t} + \sum_k Z_{j,k,t}D_{j,k,t} \qquad \forall t, j \in J, D \in \mathcal{D}$$

Planning

$$S_{j,t} \leq S_{j,max} \qquad \forall t, j \in J$$

$$S_{j,t} \geq S_{j,t-\Delta t} + \sum_k N_{j,k,t}D_{j,k,t} + M_{j,t} \cdot (S_{j,0} - S_{j,max}) \qquad \forall t, j \in J$$

$$S_{j,t} \leq S_{j,t-\Delta t} + \sum_k N_{j,k,t}D_{j,k,t} \qquad \forall t, j \in J$$

Adjustable robust optimization

Affine decision rule

$$S_{j,t} = [S_{j,t}]_0 + \sum_{k} \sum_{t'=0}^{t} [S_{j,t}]_{k,t'} D_{j,k,t'}. \tag{1}$$

Deriving a robust counterpart

Replace $D_{j,k}$ by an uncertain parameter $\tilde{d}_{j,k}$ bounded by a set $\mathcal{U}$:

$$s_{j,t} \leq s_j^{max} \qquad \forall t, j \in J$$

$$s_{j,t} = \begin{cases} s_{j,t-1} + \sum_{k \in \mathcal{K}} x_{j,k,t} \cdot \tilde{d}_{j,k}, & \text{if } m_{j,t} = 0 \\ s_j^0, & \text{otherwise} \end{cases} \qquad \forall \tilde{d}_{j,k} \in \mathcal{U}, t, j \in J$$

Reformulate:

$$m_{j,t} s_j^0 \leq s_{j,t} \leq s_j^{max} + m_{j,t} \cdot (s_j^0 - s_{j,max}) \qquad \forall t, j \in J, \tilde{d}_{j,k} \in \mathcal{U}$$

$$s_{j,t} \geq s_{j,t-\Delta t} + \sum_k x_{j,k,t} \tilde{d}_{j,k} + m_{j,t} \cdot (s_j^0 - s_j^{max}) \qquad \forall t, j \in J, \tilde{d}_{j,k} \in \mathcal{U}$$

$$s_{j,t} \leq s_{j,t-\Delta t} + \sum_k x_{j,k,t} \tilde{d}_{j,k} \qquad \forall t, j \in J, \tilde{d}_{j,k} \in \mathcal{U},$$

Replace $s_{j,t}$ by linear decision rule $s_{j,t} = [s_{j,t}]_0 + \sum_k [s_{j,t}]_k \tilde{d}_{j,k}$.

Case study: model

Objective function:

$$
\begin{aligned}
\text{cost} = \sum_{j \in J} c_j^{maint} &\left(s_j^{fin} / s_j^{max} + \sum_{t \in T} m_{j,t} \right) \\
+ c_s^{storage} &\left(q_s^{fin} + \sum_{t \in T_p} q_{s,t} \right) \\
+ U &\left(\sum_{s \in S} \phi_s^d + \sum_{t \in T_S} \phi_{s,t}^q \right)
\end{aligned}
$$

Case study: model

Constraints scheduling horizon:

$$\sum_{k\in K_j}\sum_{i\in I_j}\sum_{t'=t-p_{i,j,k}+\Delta t_S}^{t} w_{i,j,k,t'} + \sum_{t'=t-\tau_j+\Delta t_s}^{t} m_{j,t'} \leq 1 \qquad \forall J, t \in T_S \tag{2a}$$

$$v_{i,j}^{min} w_{i,j,k,t} \leq b_{i,j,k,t} \leq v_{i,j}^{max} w_{i,j,k,t} \qquad \forall J, i \in I_j, k \in K_j, t \in T_S \tag{2b}$$

$$\begin{aligned} q_{s,t} = q_{s,t-1} &+ \sum_{i\in \bar{I}_s} \bar{\rho}_{i,s} \sum_{j\in J_i}\sum_{k\in K_j} b_{i,j,k,t-p_{i,j,k}} \\ &- \sum_{i\in I_s} \rho_{i,s} \sum_{j\in J_i}\sum_{k\in K_j} b_{i,j,k,t} \end{aligned} \qquad \forall s, t \in T_S \tag{2c}$$

$$0 \leq q_{s,t} - \phi_{s,t}^{q} \leq c_s \qquad \forall s, t \in T_S \tag{2d}$$

$$m_{j,t} s_j^0 \leq s_{j,t} \leq s_j^{max} + m_{j,t} \cdot (s_j^0 - s_j^{max}) \qquad \forall t, j \in J, D \in \mathcal{U} \tag{2e}$$

$$s_{j,t} \geq s_{j,t-\Delta t_S} + \sum_i \sum_k w_{i,j,k,t} \tilde{d}_{j,k} + m_{j,t} \cdot (s_j^0 - s_j^{max}) \qquad \forall t, j \in J, D \in \mathcal{U} \tag{2f}$$

$$s_{j,t} \leq s_{j,t-\Delta t_S} + \sum_i \sum_k w_{i,j,k,t} \tilde{d}_{j,k} \qquad \forall t, j \in J, D \in \mathcal{U}, \tag{2g}$$

Case study: model

Constraints planning horizon:

$$\sum_{i \in I_j} \sum_{k \in K_j} p_{i,j,k} n_{i,j,k,t} + \tau_j m_{j,t} \leq \Delta t_P \qquad \forall J, t \in T_P \backslash \{\bar{t}_P\} \tag{2a}$$

$$v_{i,j}^{min} \sum_{k \in K_j} n_{i,j,k,t} \leq a_{i,j,t} \leq v_{i,j}^{max} \sum_{k \in K_j} n_{i,j,k,t} \qquad \forall J, i \in I_j, k \in K_j, t \in T_P \tag{2b}$$

$$q_{s,t} = q_{s,t-1} + \sum_{i \in \bar{I}_s} \bar{\rho}_{i,s} \sum_{j \in J_i} a_{i,j,t} - \sum_{i \in I_s} \rho_{i,s} \sum_{j \in J_i} a_{i,j,t} - \delta_{s,t} \qquad \forall s, t \in T_P \backslash \{\bar{t}_P\} \tag{2c}$$

$$0 \leq q_{s,t} \leq c_s \qquad \forall s, t \in T_P \tag{2d}$$

$$n_{i,j,k,t} \leq U \cdot \omega_{j,k,t} \qquad \forall J, i \in I_j, k \in K_j, t \in T_P \tag{2e}$$

$$\sum_{k \in K_j} \omega_{j,k,t} = 1 \qquad \forall J, t \in T_P \tag{2f}$$

$$s_{j,t} \leq s_j^{max} \qquad \forall t, j \in J \tag{2g}$$

$$s_j^t \geq s_{j,t-\Delta t_P} + \sum_k n_{j,k,t} \tilde{d}_{j,k,t} + m_{j,t} \cdot (s_j^0 - s_j^{max}) \qquad \forall t, j \in J \tag{2h}$$

$$s_{j,t} \leq s_{j,t-\Delta t_P} + \sum_k n_{j,k,t} \tilde{d}_{j,k,t} \qquad \forall t, j \in J \tag{2i}$$

Case study: model

Constraints interface between scheduling and planning:

$$\sum_{k\in K_j}\sum_{i\in I_j}\sum_{t'=\bar{t}_S+2\Delta t_S-p_{i,j,k}}^{\bar{t}_S} w_{i,j,k,t'}\left[p_{i,j,k}-(\bar{t}_S-t'+\Delta t_S)\right]$$
$$+\sum_{t'=\bar{t}_S+2\Delta t_S-\tau_j}^{\bar{t}_S} m_{j,t'}\left[\tau_j-(\bar{t}_S-t'+\Delta t_S)\right] \qquad \forall J \tag{2a}$$
$$+\sum_{i\in I_j}\sum_{k\in K_j} p_{i,j,k}n_{i,j,k,\bar{t}_P}+\tau_j m_{j,\bar{t}_P}\leq \Delta t_P,$$

$$q_s^{fin}=q_{s,\bar{t}_S}+\sum_{i\in \bar{I}_s}\bar{\rho}_{i,s}\sum_{j\in J_i}\sum_{k\in K_j} b_{i,j,k,\bar{t}_S+1-p_{i,j,k}}$$
$$-d_{s,\bar{t}_S}+\phi_s^d \qquad \forall s \tag{2b}$$

$$0\leq q_s^{fin}\leq c_s \qquad \forall s \tag{2c}$$

$$q_{s,\bar{t}_P}=q_s^{fin}+\sum_{i\in \bar{I}_S}\bar{\rho}_{i,s}\sum_{j\in J_i}\sum_{k\in K_j}\sum_{t'=\bar{t}_s+2-p_{i,j,k}}^{\bar{t}_S} b_{i,j,k,t'}$$
$$+\sum_{i\in \bar{I}_S}\bar{\rho}_{i,s}\sum_{j,J_i} a_{i,j,\bar{t}_P} \qquad \forall s \tag{2d}$$
$$-\sum_{i,I_s}\rho_{i,s}\sum_{j\in J_i} a_{i,j,\bar{t}_P}-d_{s,\bar{t}_P}$$

Results: instances

Instance	Toy	P1	P2	P4	P6
Units	2	4	5	3	6
Tasks	3	5	3	4	8
Op. modes	2	3	3	2	2
Products	2	2	1	2	4
Discrete vars	518	2492	1930	1869	1993
Continuous vars	1033	3630	2371	2777	4084
Constraints	1860	7332	5705	5699	7994
Avg. MIP gap [%]	0.0	3.0	5.8	10.9	1.02

Table 1: Evaluated STN instances

P1: Kondili et al. (1993), P2: Karimi and McDonald (1997), P4: Maravelias and Grossmann (2003), P6: Ierapetritou and Floudas (1998)

Results: metrics data-driven approximation

$$\mathrm{rms}^2_{all} = \frac{1}{N \cdot |A|} \sum_{n \in \{1..N\}, \alpha \in A} \left(\left[p^f_j \right]_{n,\alpha} - \bar{p}^f_j \right)^2, \tag{3a}$$

$$p_{out} = \frac{1}{N \cdot |A|} \sum_{n \in \{1..N\}, \alpha \in A} \mathbb{1} \left(\left[p^f_j \right]_{n,\alpha} > \bar{p}^f_j \right), \text{ and} \tag{3b}$$

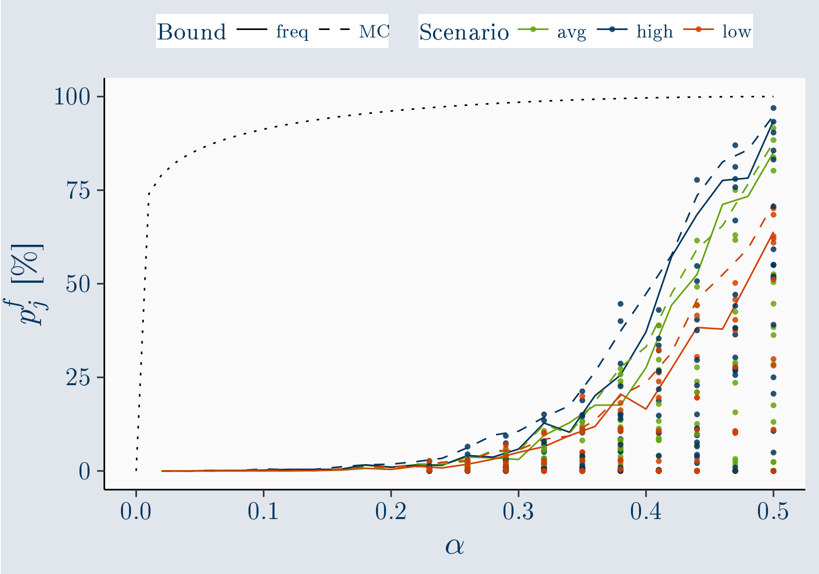
$$\mathrm{rms}^2_{out} = \frac{1}{p_{out} \cdot N \cdot |A|} \sum_{n \in \{1..N\}, \alpha \in A} \mathbb{1} \left(\left[p^f_j \right]_{n,\alpha} > \bar{p}^f_j \right) \left(\left[p^f_j \right]_{n,\alpha} - \bar{p}^f_j \right)^2, \tag{3c}$$

Results: metrics data-driven approximation

instance	bound	rms_all	rms_max	p_out
toy	freq	8.00	1.53	29.40
toy	mc	10.41	3.08	21.27
P1	freq	12.61	3.52	17.54
P1	mc	17.25	4.39	9.62
P2	freq	7.40	2.31	18.08
P2	mc	13.68	4.98	10.13
P4	freq	9.17	3.27	47.78
P4	mc	11.43	2.84	32.50
P6	freq	18.75	8.94	12.17
P6	mc	20.84	10.09	10.98
all	freq	11.19	3.91	24.99
all	mc	14.72	5.08	16.90

P1: Kondili et al. (1993), P2: Karimi and McDonald (1997), P4: Maravelias and Grossmann (2003), P6: Ierapetritou and Floudas (1998)

Results: metrics data-driven approximation



Results: metrics data-driven approximation