Data-driven optimization of processes with degrading equipment

Johannes Wiebe[1]

Wednesday 1st August, 2018

Supervisors: Ruth Misener[1], Ines Cecilio[2]

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

[2]Schlumberger Research Cambridge, Cambridge, UK London

First things first

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

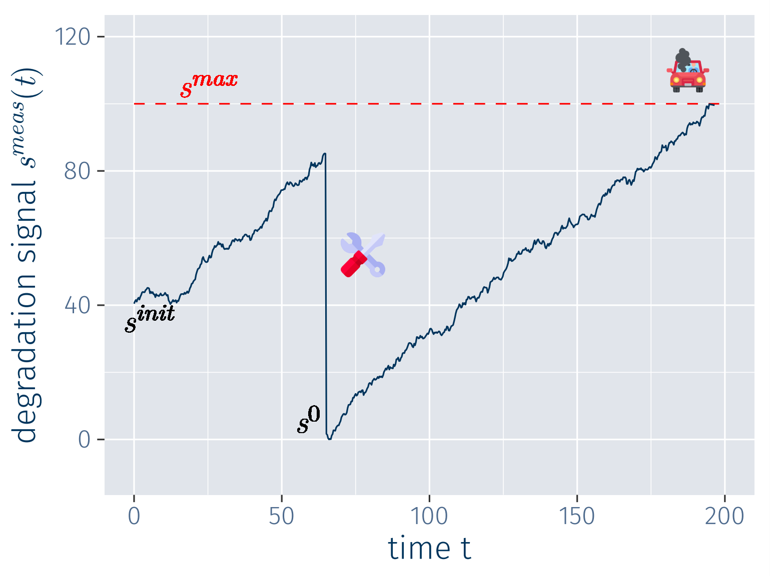
$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}} \quad & \text{cost}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}) & \\ \text{s.t.} \quad & \text{process model}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}) & \text{(eg. balance equations)} \\ & \text{maintenance model}(\boldsymbol{x},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

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

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 $s^{meas}(t)$ is often modelled by a stochastic process:

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

where S_t is a random variable.

What is degradation modelling?

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

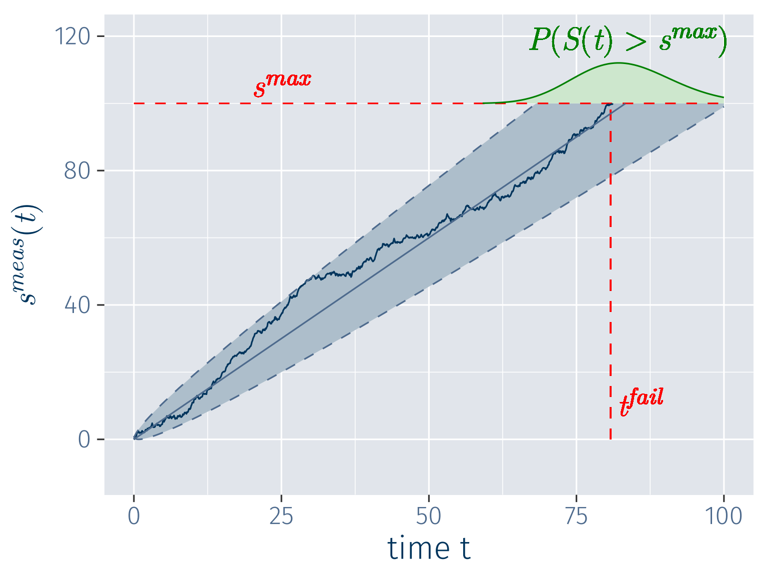
$$S(t) = \{S_t : t \in T\},$$

where S_t is a random variable.

Often used: Lévy type processes

- 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 $s < t$
- Continuity in probability: $\lim_{h \to 0} P(|S_{t+h} - S_t| > \epsilon) = 0$ for any $\epsilon > 0$, $t \geq 0$.

What is degradation modelling?



Second things second

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 $S_{j,t} - S_{j,t-\Delta t} = D_j \sim \mathcal{D}_j(\boldsymbol{\Theta}, \boldsymbol{x}, \Delta t)$.

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

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

Accounting for effects of process variables

Assumption [Liao & Tian 2013]

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

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

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

Accounting for effects of process variables

Assumption [Liao & Tian 2013]

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

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

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

Deriving a robust counterpart [Lappas & Gounaris 2016]

Random variables can be replaced by uncertain parameters

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

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

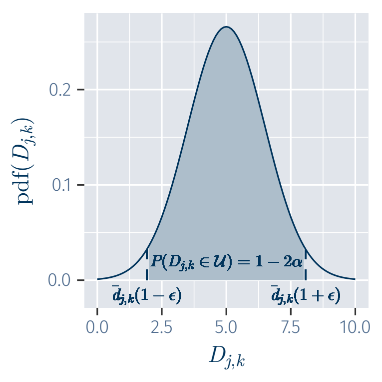
Assume a simple box uncertainty set:

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

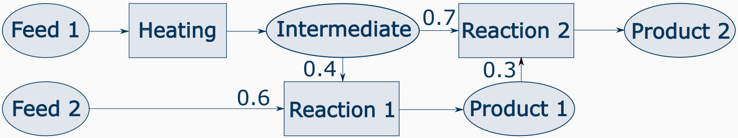
Choose $\epsilon_{j,k}$ from distribution $\mathcal{D}_{j,k}$:

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

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



Case study: State-Task-Network [Biondi et al 2017]



Objective function:

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

Case study: State-Task-Network [Biondi et al 2017]

Scheduling horizon:

$$\sum_{k\in K_j}\sum_{i\in I_j}\sum_{t'=t-p_{i,j,k}+1}^{t} w_{i,j,k,t'} + \sum_{t'=t-\tau_j+1}^{t} m_{j,t'} \leq 1 \quad \forall J, t \in T_S \qquad (1)$$

$$v_{i,j}^{min} w_{i,j,k,t} \leq b_{i,j,k,t} \leq v_{i,j}^{max} w_{i,j,k,t} \quad \forall J, i \in I_j, k \in K_j, t \in \qquad (2)$$

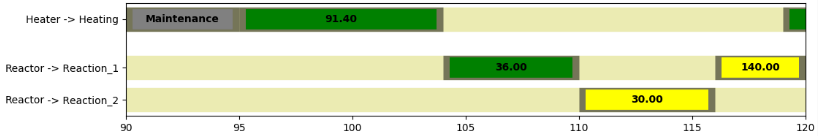
$$\begin{aligned} q_{s,t} = q_{s,t-1} &+ \sum_{i\in \bar{I}_s} \bar{\rho}_{i,s} \sum_{j\in J_i}\sum_{k\in K_j} b_{i,j,k,t-p_{i,j,k}} \\ &- \sum_{i\in I_s} \rho_{i,s} \sum_{j\in J_i}\sum_{k\in K_j} b_{i,j,k,t} \end{aligned} \quad \forall s, t \in T_S \qquad (3)$$

$$0 \leq q_{s,t} - \phi_{s,t}^{q} \leq c_s \quad \forall s, t \in T_S \qquad (4)$$

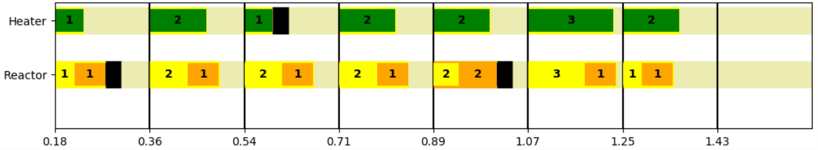
(5)

Case study: State-Task-Network [Biondi et al 2017]

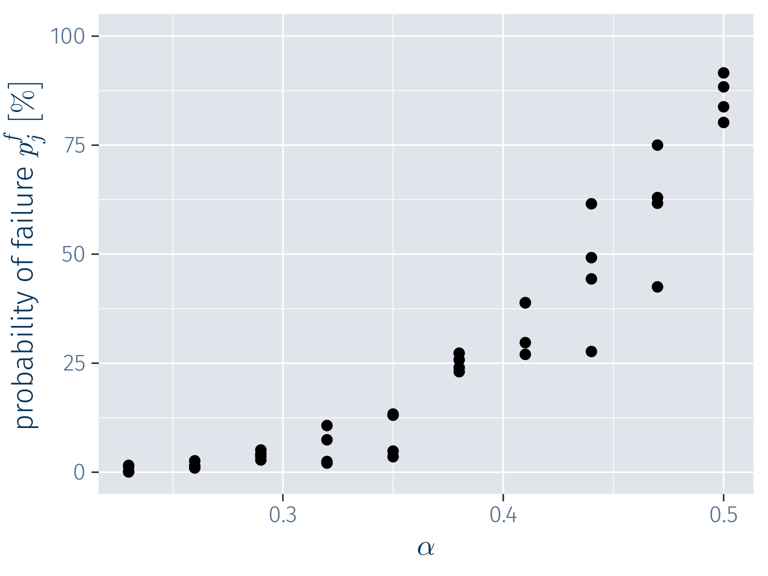
Scheduling horizon:



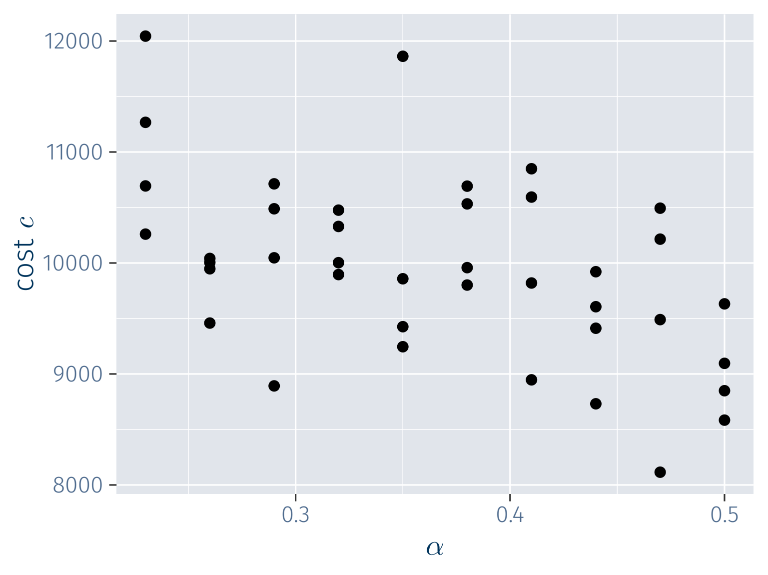
Planning horizon:



Case study: State-Task-Network [Biondi et al 2017]



Case study: State-Task-Network [Biondi et al 2017]



What is left to do?

What should the value of α be?

There is a trade-off between cost c and probability of failure p_j^f that is governed by α (the size of the uncertainty set $\mathcal{U}$).

How can we save time?

Integrated MILP model can be very big - e.g 22500 variables (2400 discrete) and 12700 constraints. Obtaining a good solution over a long time horizon takes a long time (≈ 2 hours for a reasonable instance of the STN)

Optimizing α

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 for a given α.
- $p_j^f(\alpha)$ is the corresponding probability of failure (of unit j) evaluated by Monte-Carlo simulation.
- c_j^f is the cost of an unexpected failure.

Since evaluating c^* and p_j^f is expensive, we propose using Bayesian optimization (good for black box optimization with expensive function evaluations).

Saving time

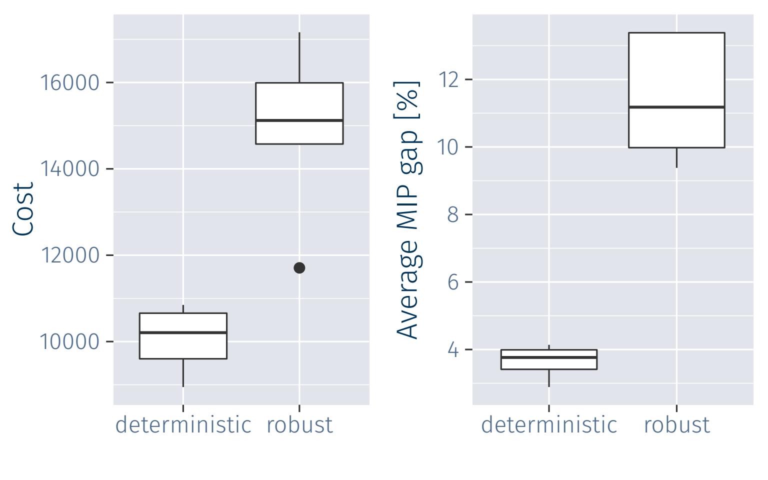
An equivalent deterministic problem

Assume that $\tilde{d}_{j,k}$ only appears in health model constraints and that $\tilde{d}_{j,k} \geq 0, \forall \tilde{d}_{j,kt} \in \mathcal{U}$. Then we can prove that a solution to

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{m}} \quad & \text{cost}(\boldsymbol{x}, \boldsymbol{m}) \\
\text{s.t} \quad & \text{process, maint. model}(\boldsymbol{x}, \boldsymbol{m}) \\
& m_{j,t} s_j^0 \leq s_{j,t}, && \forall t \\
& s_{j,t} \leq s_j^{max} + m_{j,t}(s_j^0 - s_j^{max}), && \forall t \\
& s_{j,t} \geq s_{j,t-1} + \sum_k x_{j,k,t} d_{j,k}^{max} + m_{j,t}(s_j^0 - s_j^{max}), && \forall t \\
& s_{j,t} \leq s_{j,t-1} + \sum_k x_{j,k,t} d_{j,k}^{max}, && \forall t
\end{aligned}$$

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

Saving time



Cost
16000
14000
12000
10000
deterministic
robust
Average MIP gap [%]
12
10
8
6
4
deterministic
robust

Avoiding rolling horizon

Estimating p_j^f from data

Instead of solving over long time horizon using rolling horizon we can estimate an upper bound $\bar{p}_j^f$ on the probability of failure p_j^f from data.

Frequency approach

1. Estimate the distribution of the frequencies of occurrence N_k of the operating modes from data.
2. Randomly select n_k frequencies from this distribution.
3. Arrange the selected number of operating modes in a random order.
4. Insert maintenance at the latest possible points in time.
5. Calculate p_j^f for the generated sequence.
6. Repeat steps 2 through 5 N_{mc} times and set $\bar{p}_j^f = \max p_j^f$

Avoiding rolling horizon

Markov-chain approach

1. Model sequence of operating modes as Markov chain and estimate transition probabilities

$$\pi_{k,k^*} = P(X_n = k^* | X_{n-1} = k)$$

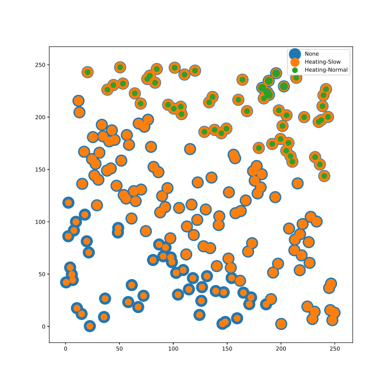
   from data.

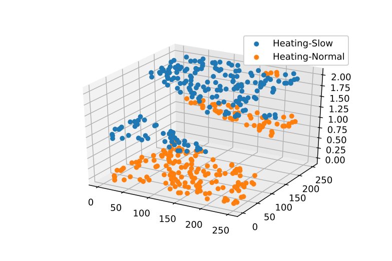
2. Randomly draw N_{mc} sequences of operating modes from Markov chain (inserting maintenance at latest point).
3. Evaluate p_j^f for each
4. $\bar{p}_j^f = \max p_j^f$

Estimating frequencies and transition probabilities

Covariate dependencies

Frequencies and transition probabilities depend on product demand $\boldsymbol{p}$.





Estimating frequencies and transition probabilities

Covariate dependencies

Frequencies and transition probabilities depend on product demand $\boldsymbol{p}$.

Logistic regression [Paton et al 2014]

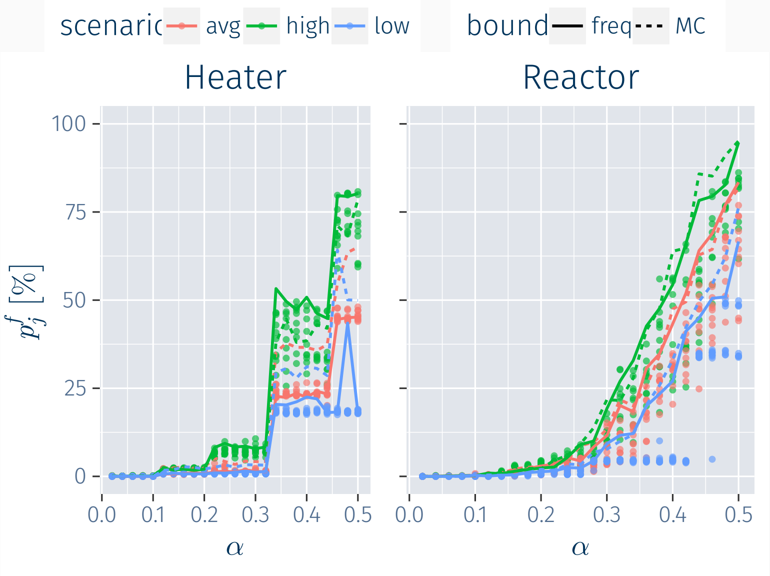
Predict the probability η_{n_k} that mode k occurs n_k times using logistic regression:

$$\eta_{n_k}(\boldsymbol{p}(t)) = P(N_k = n_k, \boldsymbol{p}) = \frac{\exp(\boldsymbol{\beta}_{n_k}\boldsymbol{p})}{\sum_{n_k'} \exp(\boldsymbol{\beta}_{n_k'}\boldsymbol{p})}$$

Similarly predict transition probabilities based on demand:

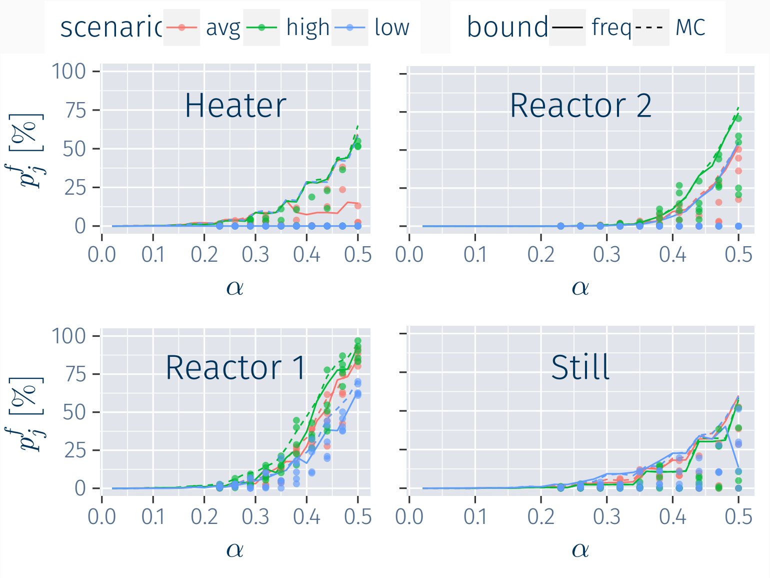
$$\pi_{k,k^*}(\boldsymbol{p}(t)) = P(X_n = k^* | X_{n-1} = k, \boldsymbol{p}) = \frac{\exp(\boldsymbol{\beta}_{n_k}\boldsymbol{p})}{\sum_{n_k'} \exp(\boldsymbol{\beta}_{n_k'}\boldsymbol{p})}$$

Results



scenario
avg
high
low
bound
freq
MC
Heater
Reactor
100
75
50
25
0
p_j^f [%]
0.0
0.1
0.2
0.3
0.4
0.5
α

Results



scenario avg high low
bound freq MC
Heater
Reactor 2
Reactor 1
Still
p_j^f [%]
α
100
75
50
25
0
0.0
0.1
0.2
0.3
0.4
0.5

Results

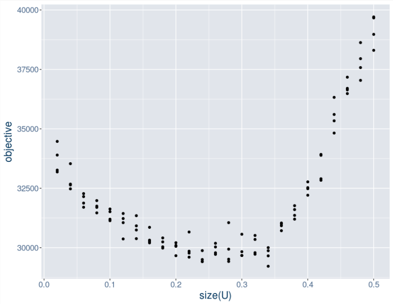
instance	bound	rms_all	rms_max	p_out
toy	freq	8.00	1.53	29.40
toy	mc	10.41	3.08	21.27
P1	freq	12.61	3.52	17.54
P1	mc	17.25	4.39	9.62
P2	freq	7.40	2.31	18.08
P2	mc	13.68	4.98	10.13
P4	freq	9.17	3.27	47.78
P4	mc	11.43	2.84	32.50
P6	freq	18.75	8.94	12.17
P6	mc	20.84	10.09	10.98
all	freq	11.19	3.91	24.99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

Table 1: Average performance metrics for probability estimates - all instances

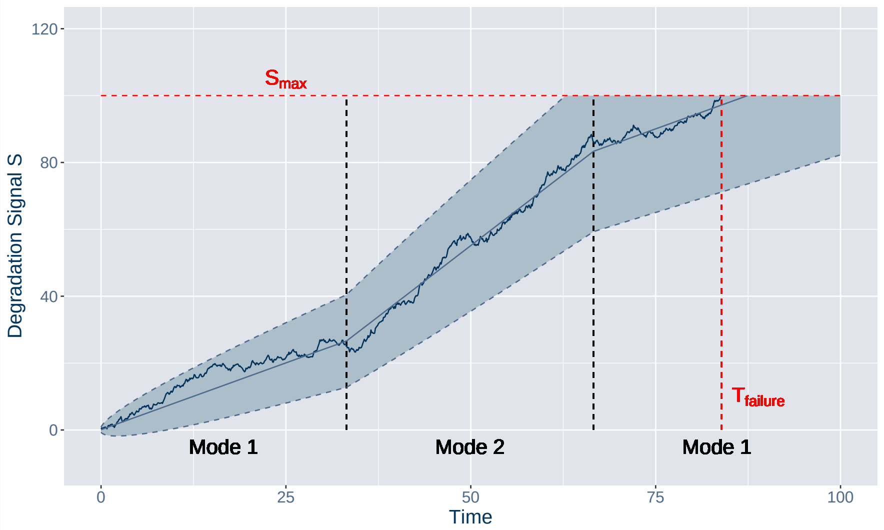
Outlook

Optimizing α

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



Degradation modelling with multiple operating modes



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]

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

$$\max_{x_1, x_2} \quad x_1 + x_2$$

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

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

Size of toy problem

	deterministic	robust $D \neq f(t)$	robust $D =$
# vars	913	3011	27719
# binaries	338	338	338
# constraints	1198	2356	13300
time to solve [s]	2	0.3-10	0.3-10
gap [%]	0	0	0
scheduling periods	30	30	30
planning periods	8	8	8
task-unit-op. mode combinations	6	6	6

Size of realistic problem

	deterministic	robust $D \neq f(t)$	robust $D =$
# vars	5389		397361
# binaries	2492		2492
# constraints	6798		180858
time to solve [s]	7883		16756
gap [%]	3.62		31.02
scheduling periods	56	56	56
planning periods	24	24	24
task-unit-op. mode combinations	24	24	24

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

Choose $\mathcal{U}$ from distribution

- Choose parameter α
- Choose D_{min} such that $P(D \leq D_{min}) = \alpha$
- Choose D_{max} such that $P(D \geq D_{max}) = \alpha$
- $\mathcal{U} = \{D | D_{min} \leq D \leq D_{max}\}$