Data-driven optimization of processes with degrading equipment

Johannes Wiebe[1]

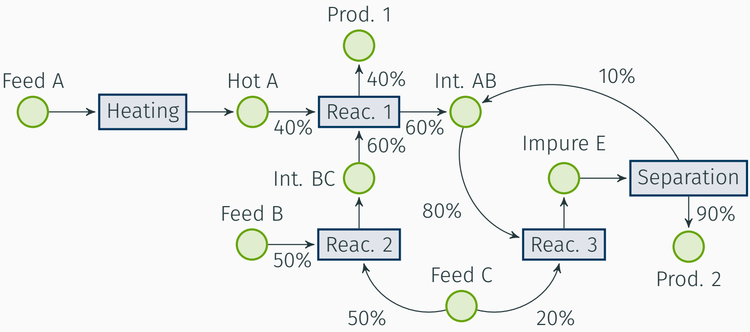
Wednesday 1st August, 2018

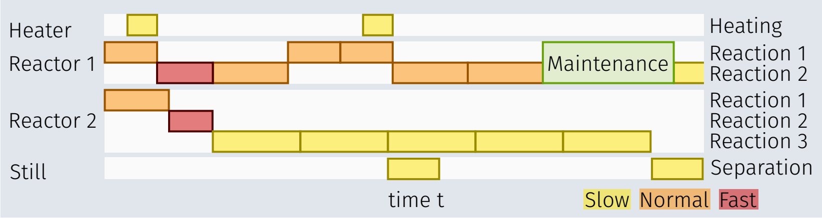
Supervisors: Ruth Misener[1], Ines Cecilio[2]

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

[2]Schlumberger Research Cambridge, Cambridge, UK London

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

?

Starting point: Process level MI(N)LP model

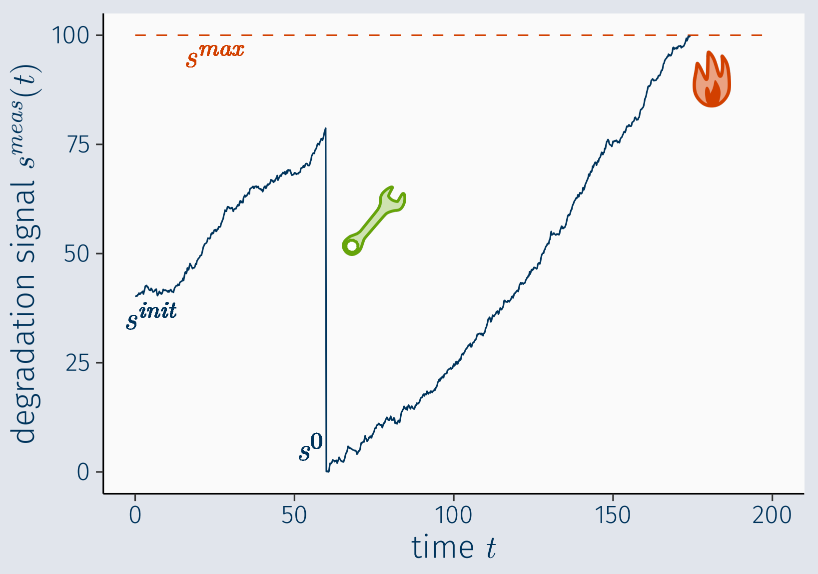
$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{m},\boldsymbol{h}} \quad & \text{cost}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}) & \\
\text{s.t.} \quad & \text{process model}(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{h}) & \text{(eg. balance equations)} \\
& \text{maintenance model}(\boldsymbol{x},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

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:

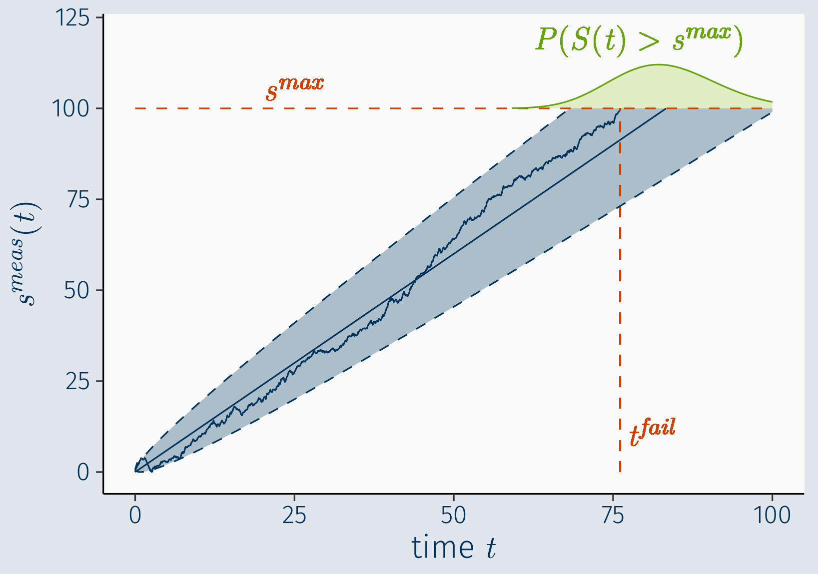
$$S(t) = \{S_t : t \in T\},$$

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?



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 $k \in K$.

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

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 $k \in K$.

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

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

Deriving a robust counterpart [Lappas & Gounaris 2016]

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

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

$\forall \tilde{d}_{j,k} \in \mathcal{U}$. Approximate $s_{j,t}\left(\tilde{d}_{j,k}\right)$ by linear decision rule.

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 $\mathcal{U}$?

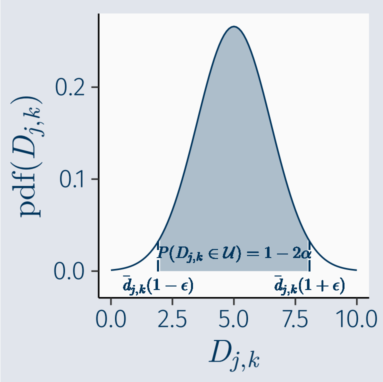
Assumption: $\mathcal{U}$ is a box uncertainty set

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

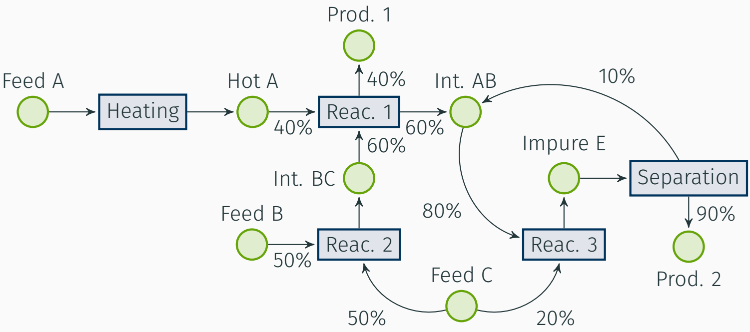
Choose $\epsilon_{j,k}$ from distribution $\mathcal{D}_{j,k}$:

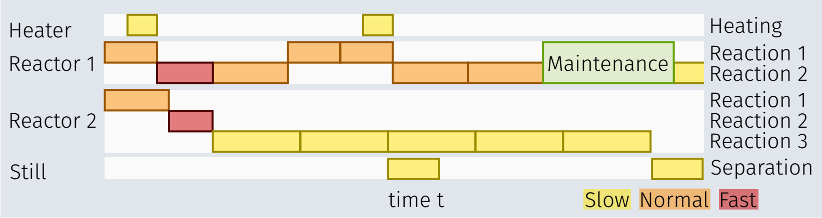
$$\epsilon_{j,k} = 1 - F^{-1}(\alpha)/\bar{d}_{j,k}$$

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



Case study: State-Task-Network [Kondili et al. 1993]

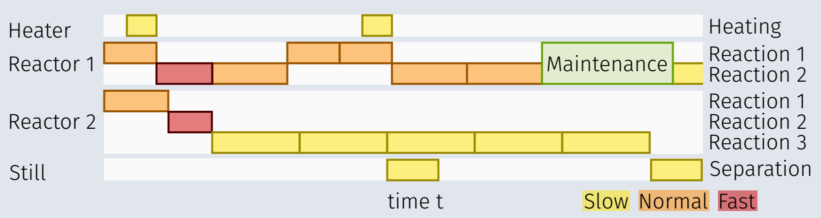




Case study: State-Task-Network [Kondili et al. 1993]

Biondi et al. (2017) extend the STN to include...

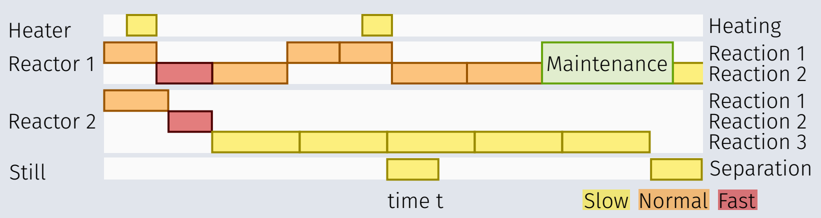
- ...unit health and maintenance scheduling
- ...integrated scheduling and planning
- ...multiple operating modes per task



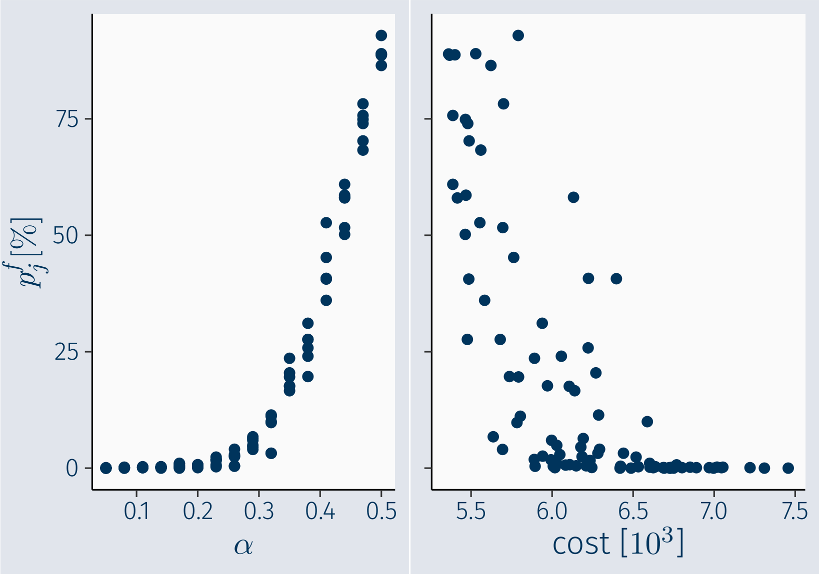
Case study: State-Task-Network [Kondili et al. 1993]

This work…

- …replaces their deterministic health model by the proposed degradation modeling based model.
- …utilizes robust optimization to obtain a solution that is likely to remain feasible.



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.

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.

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.

Saving time: a deterministic approximation

Assumption

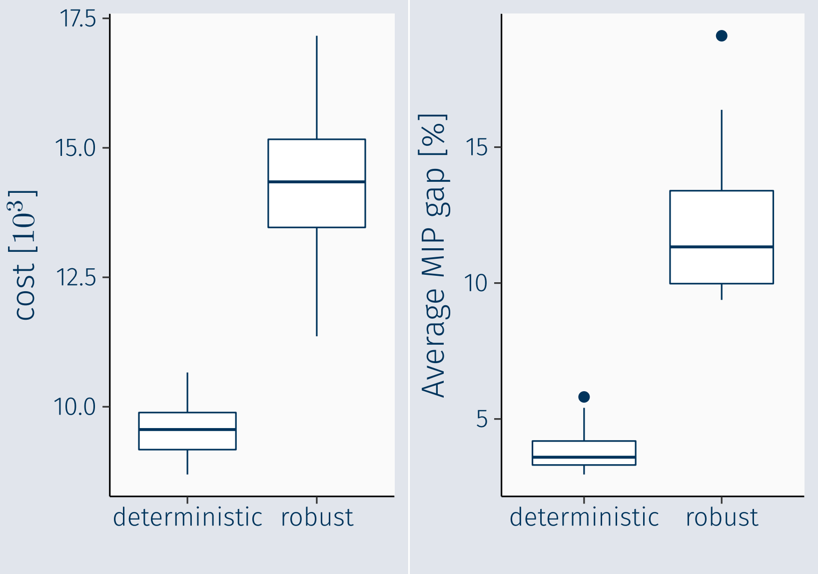
Only the health model depends on $\tilde{d}_{j,k}$ and $\tilde{d}_{j,k} \geq 0$.

Then we can prove that a solution to

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

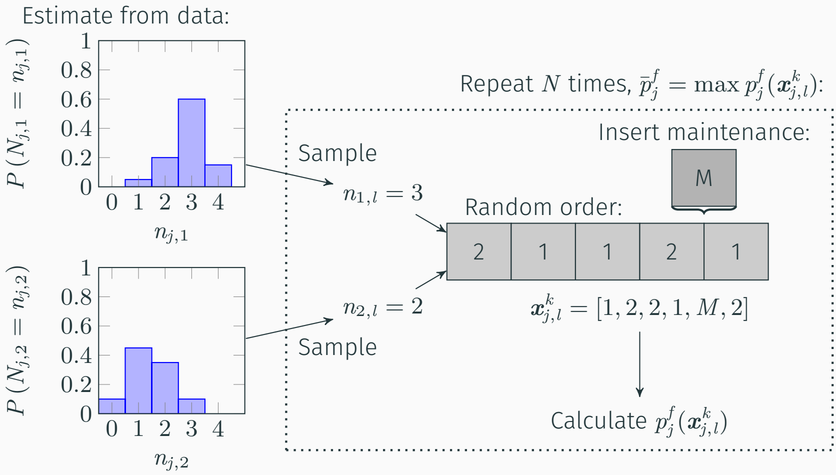
with $d_{j,k}^{max} = \max_{\mathcal{U}} \tilde{d}_{j,k}$ is also feasible in the robust problem.

Saving time: a deterministic approximation

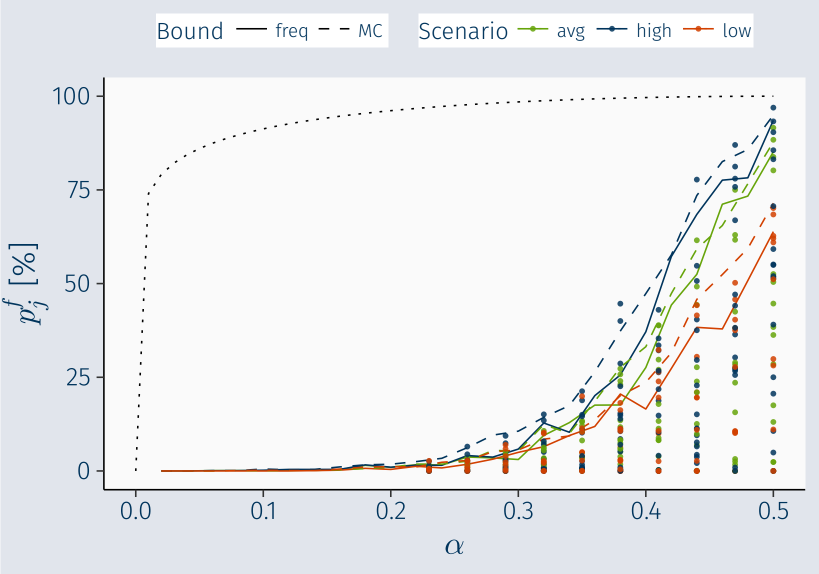


Saving time: data-driven approximations

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

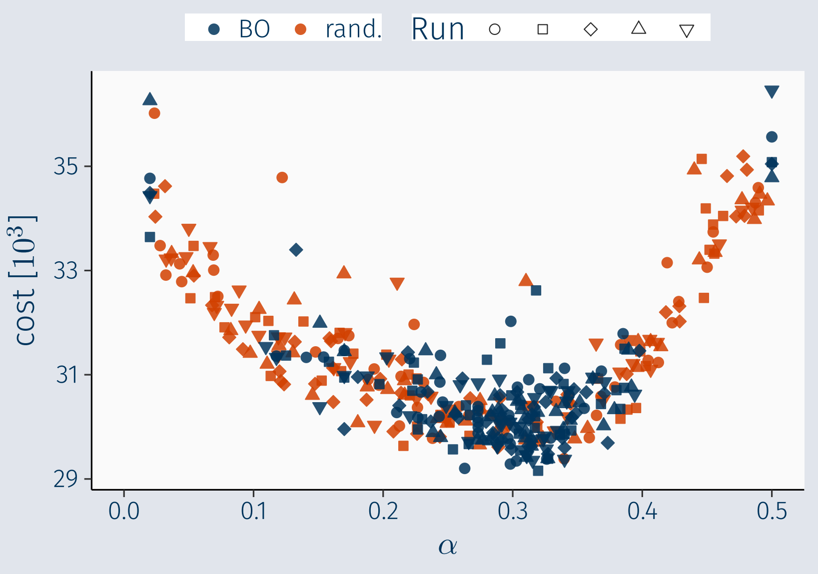


Results

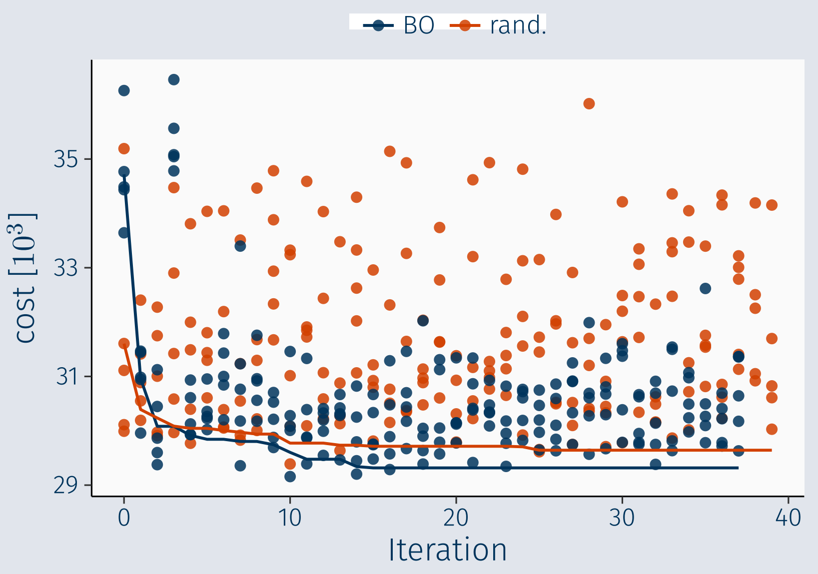


Bound
freq
MC
Scenario
avg
high
low
100
75
50
25
0
p_j^f [%]
0.0
0.1
0.2
0.3
0.4
0.5
α

Bayesian Optimization



Bayesian Optimization



Conclusion

- Something smart.
- Something really smart.
- Something really really smart.

Imperial College London

Conclusion

- Something smart.
- Something really smart.
- Something really really smart.

Imperial College London

Thank You!

computational optimisation group

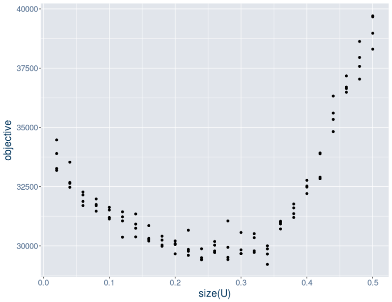
HiPEDS

Schlumberger

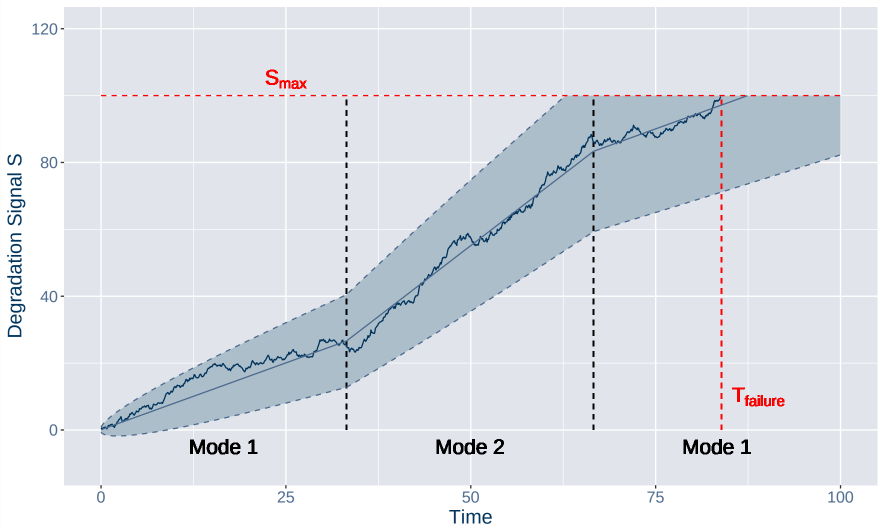
Outlook

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]

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

$$\text{s.t.} \quad \tilde{a}_{11} x_1 + \tilde{a}_{12} x_2 \leq b_1,$$

$$\forall \tilde{a}_{ij} \in \mathcal{U}$$

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

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

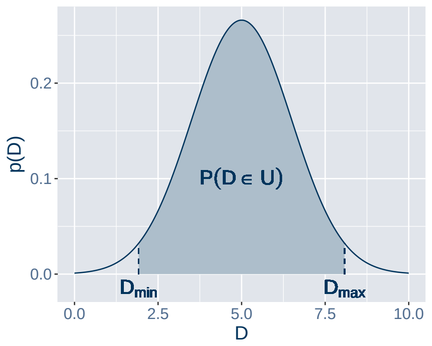
Size of realistic problem

	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 $P(D \geq D_{max}) = \alpha$
- $\mathcal{U} = \{D | D_{min} \leq D \leq D_{max}\}$



Robust optimization:deriving a robust counterpart [lappas & gounaris 2016]

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

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