Stochastic Optimization of Maintenance Scheduling: Blackbox Methods and Decomposition Approaches

Thomas BITTAR

Ph.D. Defense

February 5, 2021

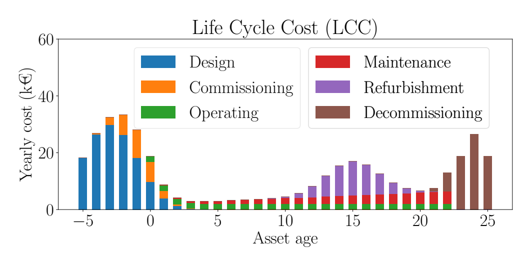
Supervised by:

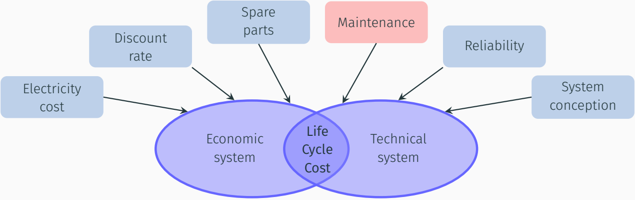
Supervised by:	P. Carpentier	*ENSTA Paris*
	J-Ph. Chancelier	*CERMICS, École des Ponts ParisTech*
	J. Lonchampt	*EDF R&D, PRISME Department*

École des Ponts ParisTech

EDF

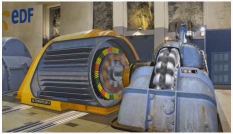
Short introduction to engineering asset management



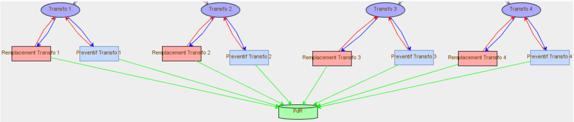


SYSTEMS OF INTEREST IN THIS WORK

- From 2 to 80 components of a hydroelectric power plant: turbines, generators, transformers

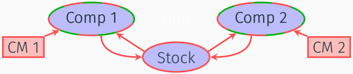


- Common stock of spares, initial stock with a low number of parts

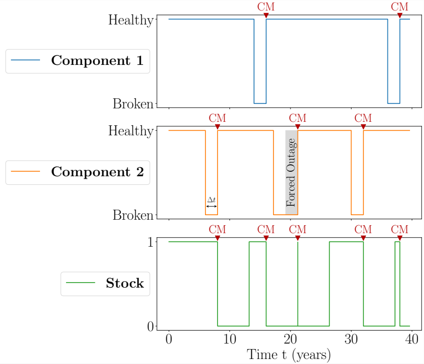


- Horizon of study: 40 years

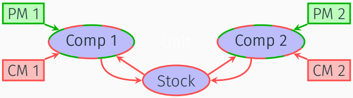
Maintenance strategies and dynamics of the industrial system



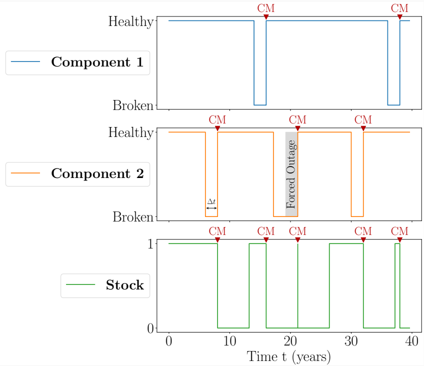
Reference strategy
Corrective maintenance only

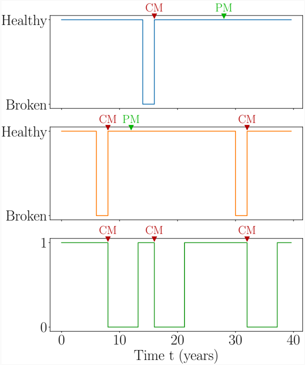


Maintenance strategies and dynamics of the industrial system



Reference strategy	Preventive strategy
Corrective maintenance only	Corrective and Preventive maintenance





Order of magnitude of costs

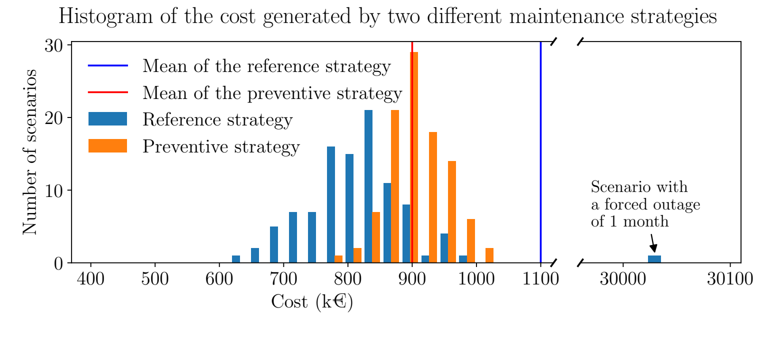
- Costs of maintenance and forced outage have different order of magnitude:
  - Preventive maintenance: ~ 100 k€
  - Corrective maintenance: ~ 500 k€
  - Forced outage: ~ 30000 k€/month

Order of magnitude of costs

- Costs of maintenance and forced outage have different order of magnitude:
  - Preventive maintenance: ~ 100 k€
  - Corrective maintenance: ~ 500 k€
  - Forced outage: ~ 30000 k€/month
- Failures of the components are random events $\Rightarrow$ LCC is a random variable

Order of magnitude of costs

- Costs of maintenance and forced outage have different order of magnitude:
  - Preventive maintenance: ~ 100 k€
  - Corrective maintenance: ~ 500 k€
  - Forced outage: ~ 30000 k€/month
- Failures of the components are random events $\Rightarrow$ LCC is a random variable
- Expected cost of a strategy estimated with Monte Carlo scenarios



Main goal and challenges

Industrial goal

For a given system, find the deterministic (open loop) maintenance strategy that minimizes the expectation of the LCC.

Main goal and challenges

Industrial goal

For a given system, find the deterministic (open loop) maintenance strategy that minimizes the expectation of the LCC.

Optimization challenges:

- Large-scale optimization problem (up to 80 components)
- Expected LCC computed with the simulation model VME: blackbox objective function

- VME uses Monte-Carlo simulations to estimate the expected LCC: no access to the true value of the objective but only noisy evaluations
- Evaluations of the objective function are expensive
  Industrial case with 80 components: $\sim$ 30 min for one evaluation

1 Blackbox methods for optimal maintenance scheduling

2 A decomposition by prediction for the maintenance problem

3 Contributions on the stochastic Auxiliary Problem Principle

4 Conclusion

Submitted papers

[BCCL20] T. Bittar, P. Carpentier, J-Ph. Chancelier, and J. Lonchampt.
A Decomposition Method by Interaction Prediction for the Optimization of Maintenance Scheduling.
Submitted to Annals of Operations Research, 2020.

[BCCL21] T. Bittar, P. Carpentier, J-Ph. Chancelier, and J. Lonchampt.
The stochastic Auxiliary Problem Principle in Banach spaces: measurability and convergence.
Submitted to SIAM Journal on Optimization, 2021.

Outline

1 Blackbox methods for optimal maintenance scheduling

1.1 Review of kriging and the EGO algorithm

1.2 The EGO-FSSF algorithm

1.3 The MADS algorithm

1.4 Computational results

1.5 Conclusion on blackbox methods

2 A decomposition by prediction for the maintenance problem

3 Contributions on the stochastic Auxiliary Problem Principle

4 Conclusion

Overview of kriging a.k.a. Gaussian process regression I

Goal

Predict the values of a function $f : U^{\mathrm{ad}} \to \mathbb{R}$ on U^{ad} from its values on an initial design of experiments $U^{\mathrm{o}} = \{u_1, \ldots, u_l\} \in U^{\mathrm{ad}}$, where U^{ad} is a subset of a Hilbert space $\mathbb{U}$.

Assumption

f is the realization of a Gaussian process $\boldsymbol{Z} = \{\boldsymbol{Z}_u : \Omega \to \mathbb{R}\}_{u \in U^{\mathrm{ad}}}$ characterized by:

- its mean function $\mu : u \in U^{\mathrm{ad}} \mapsto \mathbb{E}(\boldsymbol{Z}_u) \in \mathbb{R}$
- its covariance function $k : (u, v) \in U^{\mathrm{ad}} \times U^{\mathrm{ad}} \mapsto \mathrm{Cov}(\boldsymbol{Z}_u, \boldsymbol{Z}_v) \in \mathbb{R}$

Overview of kriging a.k.a. Gaussian process regression II

Consider the event $\mathcal{A}_l : \left\{ \boldsymbol{Z}_{u_1} = f(u_1), \ldots, \boldsymbol{Z}_{u_l} = f(u_l) \right\}$, then:

$$[\boldsymbol{Z}_u \mid \mathcal{A}_l] \sim \mathcal{N}(m_l(u), s_l^2(u)), \quad u \in U^{\mathrm{ad}}$$

The kriging mean $m_l(u)$ and the kriging variance $s_l^2(u)$ can be computed analytically.

Overview of kriging a.k.a. Gaussian process regression II

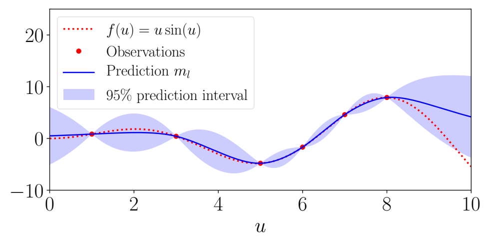
Consider the event $\mathcal{A}_l : \{\boldsymbol{Z}_{u_1} = f(u_1), \ldots, \boldsymbol{Z}_{u_l} = f(u_l)\}$, then:

$$[\boldsymbol{Z}_u \mid \mathcal{A}_l] \sim \mathcal{N}(m_l(u), s_l^2(u)), \quad u \in U^{\mathrm{ad}}$$

The kriging mean $m_l(u)$ and the kriging variance $s_l^2(u)$ can be computed analytically.

The kriging prediction for $f(u)$ is $m_l(u)$ with a confidence interval of level α given by:

$$[m_l(u) - \Phi^{-1}(1-\alpha/2)s_l(u)\ ,\ m_l(u) + \Phi^{-1}(1-\alpha/2)s_l(u)]$$



The conditional Gaussian process characterized by m_l and s_l is the kriging metamodel.

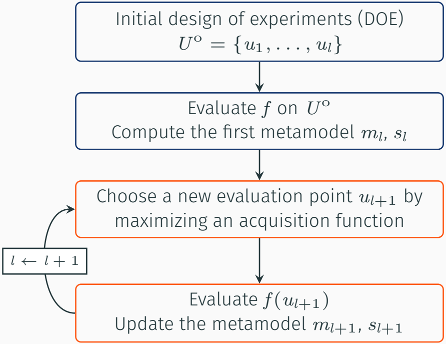
The Efficient Global Optimization (EGO) algorithm [JSW98]

Goal: Solve the minimization problem:

$$\min_{u \in U^{\mathrm{ad}}} f(u)$$

Idea

Take advantage of the kriging prediction to smartly choose the successive evaluation points of f .



Common acquisition function:
Expected Improvement

$$EI_l(u) = \mathbb{E}[I_l(u) | \mathcal{A}_l]$$

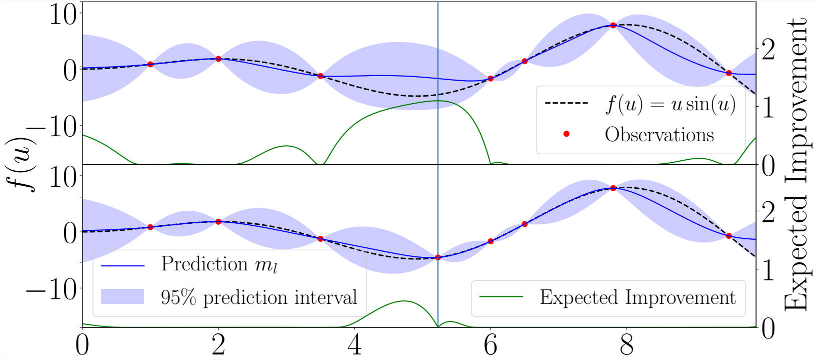
with

$$I_l(u) = \left(\min_{1 \leq i \leq l} f(u_i) - \boldsymbol{Z}_u \right)^+$$

We set:

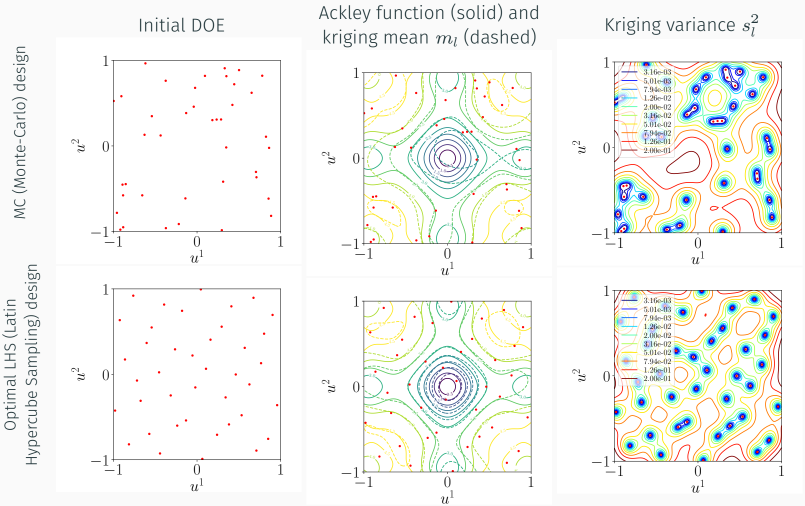
$$u_{l+1} \in \arg\max_{u \in U^{\mathrm{ad}}} EI_l(u)$$

Illustration of an EGO iteration



On the importance of the initial design of experiments (DOE) for kriging

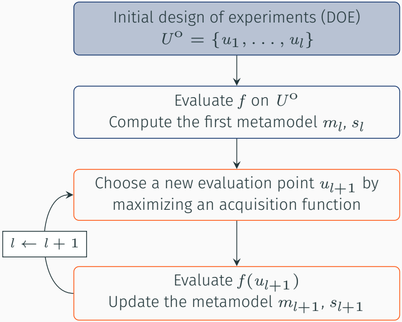
Accuracy of kriging depends on the initial DOE: Example with the Ackley function



The initial DOE in EGO

In the literature, the initial DOE for EGO is a fixed-size space-filling design:

- Minimax or maximin designs
- Optimal LHS designs



Characteristics:

1. The size l of the initial DOE only depends on the dimension of the input space.
2. The location of the l points is determined simultaneously.
3. The design does not depend on the underlying function f we minimize.

$\Rightarrow$ No guarantee on the accuracy of the initial metamodel in the EGO algorithm.

Outline

1 Blackbox methods for optimal maintenance scheduling
1.1 Review of kriging and the EGO algorithm
1.2 The EGO-FSSF algorithm
1.3 The MADS algorithm
1.4 Computational results
1.5 Conclusion on blackbox methods

2 A decomposition by prediction for the maintenance problem

3 Contributions on the stochastic Auxiliary Problem Principle

4 Conclusion

Introduction to FSSF (Fully Sequential Space-Filling) designs

Contributions

- Use a FSSF (Fully Sequential Space-Filling) initial design [SA20] that is adapted to the difficulty of the underlying optimization problem
- Ensure that the metamodel is accurate before launching the infill step of EGO

$\longrightarrow$ The EGO-FSSF algorithm

Introduction to FSSF (Fully Sequential Space-Filling) designs

Contributions

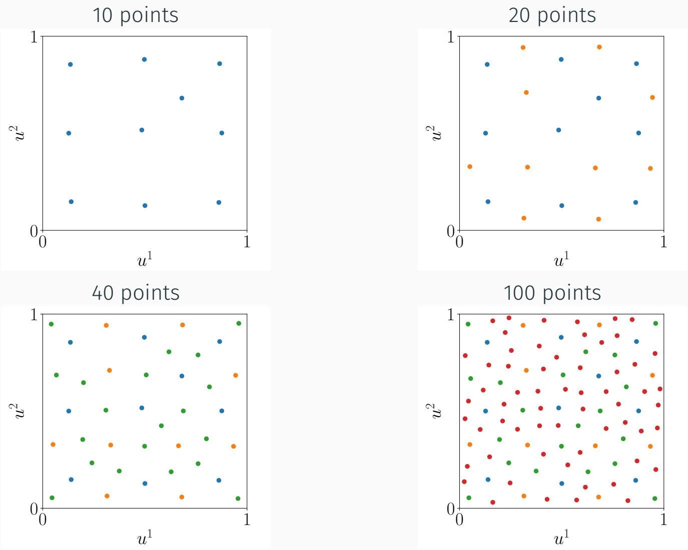
- Use a FSSF (Fully Sequential Space-Filling) initial design [SA20] that is adapted to the difficulty of the underlying optimization problem
- Ensure that the metamodel is accurate before launching the infill step of EGO

$\longrightarrow$ The EGO-FSSF algorithm

Characteristics of FSSF designs:

- Fully sequential: Points are added one-at-a-time. For $m < n$, the design with m points is a subset of the design with n points.
- Space-filling: At each new added point, the design retains good space-filling properties.

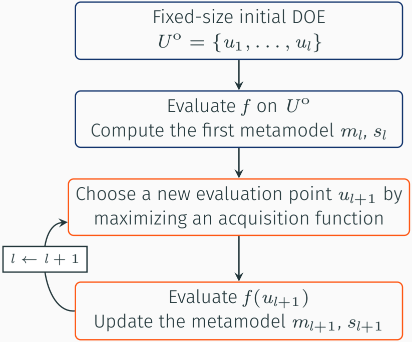
Examples of FSSF designs



Description of the EGO-FSSF algorithm

Goal: Solve the minimization problem $\min_{u \in U^{\mathrm{ad}}} f(u)$

Recall the EGO algorithm:

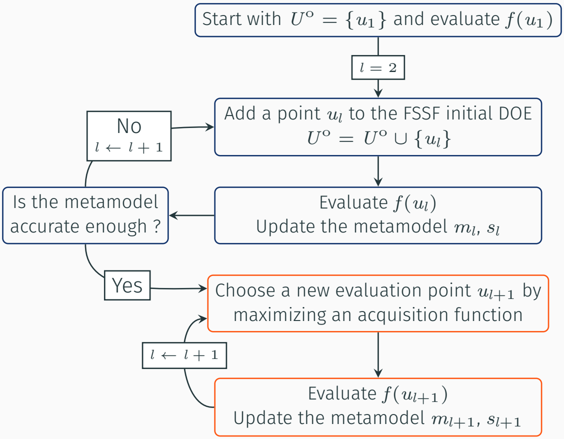


Description of the EGO-FSSF algorithm

Goal: Solve the minimization problem $\min_{u \in U^{\text{ad}}} f(u)$

Contribution

The EGO-FSSF algorithm

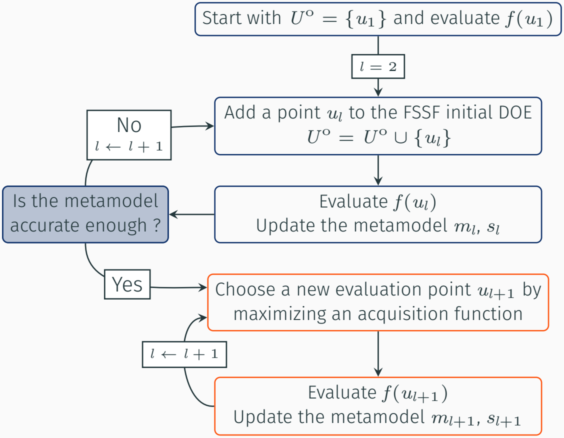


Description of the EGO-FSSF algorithm

Goal: Solve the minimization problem $\min_{u \in U^{ad}} f(u)$

Contribution

The EGO-FSSF algorithm



Metamodel validation

Introduce $\{v_1, \dots, v_p\} \subset U^{\text{ad}}$: Test sample disjoint from the DOE $U^{\text{o}} = \{u_1, \dots, u_l\}$

- The predictivity coefficient Q^2 (the higher, the better):

$$Q^2 = 1 - \frac{\sum_{i=1}^{p} \left(f(v_i) - m_l(v_i)\right)^2}{\sum_{i=1}^{p} \left(f(v_i) - \frac{1}{p}\sum_{j=1}^{p} f(v_j)\right)^2}$$

$\rightarrow$ Quantifies the predictive performance of the metamodel

Metamodel validation

Introduce $\{v_1, \dots, v_p\} \subset U^{\text{ad}}$: Test sample disjoint from the DOE $U^{\text{o}} = \{u_1, \dots, u_l\}$

- The predictivity coefficient Q^2 (the higher, the better):

$$Q^2 = 1 - \frac{\sum_{i=1}^{p} \left(f(v_i) - m_l(v_i)\right)^2}{\sum_{i=1}^{p} \left(f(v_i) - \frac{1}{p}\sum_{j=1}^{p} f(v_j)\right)^2}$$

$\rightarrow$ Quantifies the predictive performance of the metamodel

- The Predictive Variance Adequacy PVA (the lower, the better):

$$\text{PVA} = \left| \log_{10}\left(\frac{1}{p} \sum_{i=1}^{p} \frac{\left(f(v_i) - m_l(v_i)\right)^2}{s_l^2(v_i)} \right) \right|$$

$\rightarrow$ Quantifies the accuracy of the prediction intervals given by the metamodel

Metamodel validation

Introduce $\{v_1, \dots, v_p\} \subset U^{\mathrm{ad}}$: Test sample disjoint from the DOE $U^{\mathrm{o}} = \{u_1, \dots, u_l\}$

- The predictivity coefficient Q^2 (the higher, the better):

$$Q^2 = 1 - \frac{\sum_{i=1}^{p} \left(f(v_i) - m_l(v_i)\right)^2}{\sum_{i=1}^{p} \left(f(v_i) - \frac{1}{p}\sum_{j=1}^{p} f(v_j)\right)^2}$$

$\rightarrow$ Quantifies the predictive performance of the metamodel

- The Predictive Variance Adequacy PVA (the lower, the better):

$$\mathrm{PVA} = \left| \log_{10}\left(\frac{1}{p} \sum_{i=1}^{p} \frac{\left(f(v_i) - m_l(v_i)\right)^2}{s_l^2(v_i)} \right) \right|$$

$\rightarrow$ Quantifies the accuracy of the prediction intervals given by the metamodel

In the EGO-FSSF algorithm

- User-defined thresholds $Q^2_{\mathrm{min}} < 1$ and $\mathrm{PVA}_{\mathrm{max}} > 0$
- Metamodel is accurate enough if $Q^2 > Q^2_{\mathrm{min}}$ and $\mathrm{PVA} < \mathrm{PVA}_{\mathrm{max}}$
- Q^2 and PVA computed by cross-validation

Outline

1 Blackbox methods for optimal maintenance scheduling

1.1 Review of kriging and the EGO algorithm

1.2 The EGO-FSSF algorithm

1.3 The MADS algorithm

1.4 Computational results

1.5 Conclusion on blackbox methods

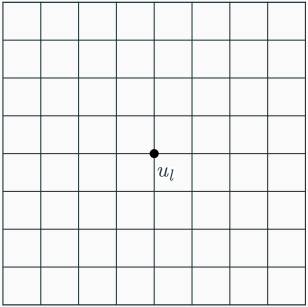
2 A decomposition by prediction for the maintenance problem

3 Contributions on the stochastic Auxiliary Problem Principle

4 Conclusion

The blackbox algorithm MADS (Mesh Adaptive Direct Search) [AD06]

Goal: Solve the minimization problem $\min_{u \in U^{\mathrm{ad}}} f(u)$



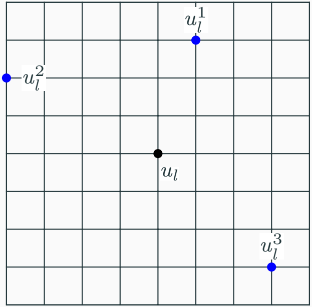
At iteration l:

- Current iterate u_l
- Mesh M_l

- Mesh M_l: Points defined by the intersection of the lines

The blackbox algorithm MADS (Mesh Adaptive Direct Search) [AD06]

Goal: Solve the minimization problem $\min_{u \in U^{\text{ad}}} f(u)$

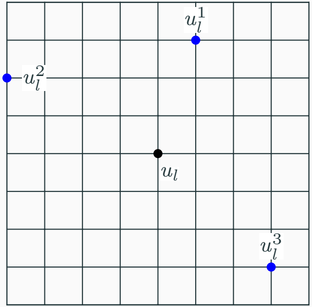


1. Search step (exploration): Evaluate f on a finite number of points $\{u_l^1, \cdots, u_l^n\} \subset M_l$ chosen with any user-defined strategy.

- Mesh M_l: Points defined by the intersection of the lines
- Search points are in blue

The blackbox algorithm MADS (Mesh Adaptive Direct Search) [AD06]

Goal: Solve the minimization problem $\min_{u \in U^{\mathrm{ad}}} f(u)$



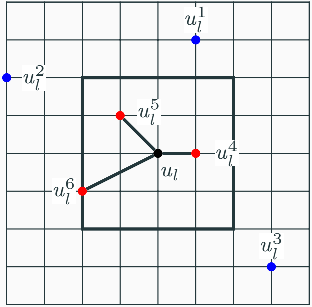
1. Search step (exploration): Evaluate f on a finite number of points $\{u_l^1, \cdots, u_l^n\} \subset M_l$ chosen with any user-defined strategy.

- If there exists $1 \leq i \leq n$ such that $f(u_l^i) < f(u_l)$, then $u_{l+1} = u_l^i$ and increase the mesh size parameter.
- Else, go to poll step.

- Mesh M_l: Points defined by the intersection of the lines
- Search points are in blue

The blackbox algorithm MADS (Mesh Adaptive Direct Search) [AD06]

Goal: Solve the minimization problem $\min_{u \in U^{\text{ad}}} f(u)$

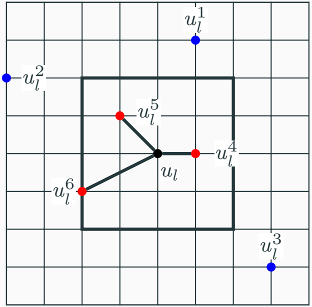


2. Poll step (exploitation): Evaluate f on $P_l = \{u_l^{n+1}, \cdots, u_l^p\}$. The points in P_l are in the neighbourhood of the current iterate u_l.

- Mesh M_l: Points defined by the intersection of the lines
- Search points are in blue
- Poll points are in red

The blackbox algorithm MADS (Mesh Adaptive Direct Search) [AD06]

Goal: Solve the minimization problem $\min_{u \in U^{\mathrm{ad}}} f(u)$



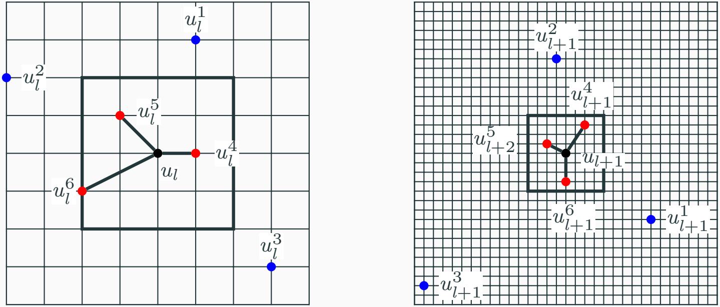
2. Poll step (exploitation): Evaluate f on $P_l = \{u_l^{n+1}, \cdots, u_l^p\}$. The points in P_l are in the neighbourhood of the current iterate u_l.

- If there exists $n+1 \leq i \leq p$ such that $f(u_l^i) < f(u_l)$, then $u_{l+1} = u_l^i$ and increase the mesh size parameter.
- Else $u_{l+1} = u_l$ and decrease the mesh size parameter.

- Mesh M_l: Points defined by the intersection of the lines
- Search points are in blue
- Poll points are in red

The blackbox algorithm MADS (Mesh Adaptive Direct Search) [AD06]

Goal: Solve the minimization problem $\min_{u \in U^{\mathrm{ad}}} f(u)$



Convergence result

Under mild assumptions, MADS converges to a stationary point of f.

Outline

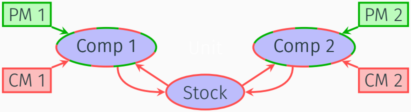
1 Blackbox methods for optimal maintenance scheduling
- 1.1 Review of kriging and the EGO algorithm
- 1.2 The EGO-FSSF algorithm
- 1.3 The MADS algorithm
- 1.4 Computational results
- 1.5 Conclusion on blackbox methods

2 A decomposition by prediction for the maintenance problem

3 Contributions on the stochastic Auxiliary Problem Principle

4 Conclusion

Description of the industrial system



Parameter	Value		
Number of components n	$2, 3, 5$ or 10		
Initial number of spare parts	$\lfloor \frac{n}{5} \rfloor$		
Horizon	40 years		
Forced outage cost	30000 k€/ month		
	Comp. 1	Comp. 2	Comp. $i \geq 3$
PM cost	50 k€	50 k€	50 k€
CM cost	100 k€	250 k€	200 k€
Failure distribution	$\mathrm{Weib}(2.3, 10)$	$\mathrm{Weib}(4, 20)$	$\mathrm{Weib}(3, 10)$
Mean time to failure	8.85 years	18.13 years	8.93 years

The industrial maintenance optimization problem

Industrial problem

Find the periodic maintenance strategy that minimizes the expected Life Cycle Cost (LCC) of the system:

$$\min_{u \in U^{\text{ad}}} \mathbb{E}\big(j(u, W)\big)$$

- $u = (u_1, \dots, u_n) \in U^{\text{ad}} = [0, T]^n$, where T is the time horizon (40 years)
- W: random variable on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$, models the failures of the components
- $j : U^{\text{ad}} \times \Omega \to \mathbb{R}$: LCC of the system

The industrial maintenance optimization problem

Industrial problem

Find the periodic maintenance strategy that minimizes the expected Life Cycle Cost (LCC) of the system:

$$\min_{u \in U^{\mathrm{ad}}} \mathbb{E}\big(j(u, W)\big)$$

- $u = (u_1, \dots, u_n) \in U^{\mathrm{ad}} = [0, T]^n$, where T is the time horizon (40 years)
- W: random variable on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$, models the failures of the components
- $j : U^{\mathrm{ad}} \times \Omega \to \mathbb{R}$: LCC of the system

- We solve a Monte-Carlo approximation of the problem:

$$\min_{u \in U^{\mathrm{ad}}} \frac{1}{p} \sum_{i=1}^{p} j(u, w_i)$$

with $w_1, \dots, w_p$ being realizations of the random variable W

The industrial maintenance optimization problem

Industrial problem

Find the periodic maintenance strategy that minimizes the expected Life Cycle Cost (LCC) of the system:

$$\min_{u \in U^{\mathrm{ad}}} \mathbb{E}\big(j(u, W)\big)$$

- $u = (u_1, \ldots, u_n) \in U^{\mathrm{ad}} = [0, T]^n$, where T is the time horizon (40 years)
- W: random variable on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$, models the failures of the components
- $j : U^{\mathrm{ad}} \times \Omega \to \mathbb{R}$: LCC of the system

- We solve a Monte-Carlo approximation of the problem:

$$\min_{u \in U^{\mathrm{ad}}} \frac{1}{p} \sum_{i=1}^{p} j(u, w_i)$$

  with $w_1, \ldots, w_p$ being realizations of the random variable W
- Objective function evaluated with the blackbox software VME

The industrial maintenance optimization problem

Industrial problem

Find the periodic maintenance strategy that minimizes the expected Life Cycle Cost (LCC) of the system:

$$\min_{u \in U^{\mathrm{ad}}} \mathbb{E}\big(j(u, W)\big)$$

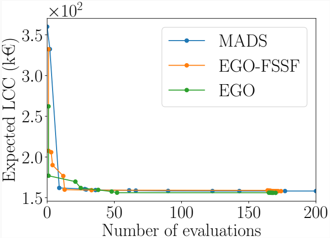
- $u = (u_1, \ldots, u_n) \in U^{\mathrm{ad}} = [0, T]^n$, where T is the time horizon (40 years)
- W: random variable on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$, models the failures of the components
- $j : U^{\mathrm{ad}} \times \Omega \to \mathbb{R}$: LCC of the system

- We solve a Monte-Carlo approximation of the problem:

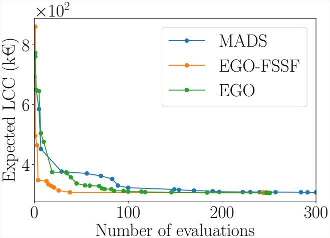
$$\min_{u \in U^{\mathrm{ad}}} \frac{1}{p} \sum_{i=1}^{p} j(u, w_i)$$

  with $w_1, \ldots, w_p$ being realizations of the random variable W
- Objective function evaluated with the blackbox software VME
- MADS, EGO and EGO-FSSF plugged on VME to perform optimization

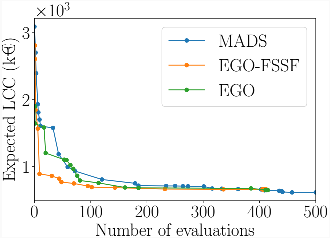
Results on the industrial problem



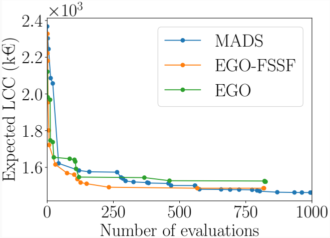
(a) Case with 2 components



(b) Case with 3 components



(c) Case with 5 components



(d) Case with 10 components

Conclusion of the computational tests

- EGO-FSSF more efficient than MADS in the first iterations
- MADS eventually outputs a better solution than EGO with more evaluations
- Striking difference in running times (given for the 10-component case):

	EGO	MADS
Running time	$\sim$ 10h	$\sim$ 1min

Outline

1 Blackbox methods for optimal maintenance scheduling

1.1 Review of kriging and the EGO algorithm

1.2 The EGO-FSSF algorithm

1.3 The MADS algorithm

1.4 Computational results

1.5 Conclusion on blackbox methods

2 A decomposition by prediction for the maintenance problem

3 Contributions on the stochastic Auxiliary Problem Principle

4 Conclusion

Conclusion on the optimization with blackbox methods

Contributions

1. Improvement of the initial design step within EGO: the EGO-FSSF algorithm
2. Comprehensive benchmark of EGO, EGO-FSSF and MADS (not in the talk)
3. Benchmark of solvers for the EI maximization within EGO (not in the talk)
4. Application of EGO-FSSF and MADS on a maintenance optimization problem

We have tackled periodic maintenance problems with up to 10 components:

→ Industrial application for common maintenance operations (e.g. lubrication of the components)

Towards maintenance optimization for the most demanding cases at EDF

What we have done:

- Periodic maintenance strategies
- Up to 10 components

Ultimate goal:

- General maintenance strategies
- Up to 80 components:
  most demanding cases at EDF

Limits of blackbox methods:

- Large instances intractable with EGO
- MADS may not be able to efficiently explore the search space in high dimension

Decomposition method !

OUTLINE

1 Blackbox methods for optimal maintenance scheduling

2 A decomposition by prediction for the maintenance problem

2.1 Formalization of the maintenance optimization problem

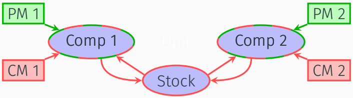
2.2 A decomposition method component by component

2.3 Computational results on the 80-component industrial case

3 Contributions on the stochastic Auxiliary Problem Principle

4 Conclusion

Philosophy behind the decomposition approach



1. The industrial system is a structured physical system.
    - Several similar components
    - Coupled by a common stock of spare part

Step 1

Analytical formulation of the dynamics $\longrightarrow$ We open the blackbox!

2. Take advantage of the structure of the system to efficiently perform the maintenance optimization.

Step 2

Design of a decomposition-coordination method

Outline

1 Blackbox methods for optimal maintenance scheduling

2 A decomposition by prediction for the maintenance problem

2.1 Formalization of the maintenance optimization problem

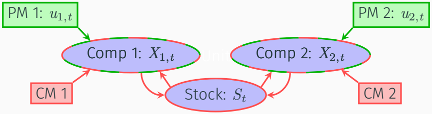
2.2 A decomposition method component by component

2.3 Computational results on the 80-component industrial case

3 Contributions on the stochastic Auxiliary Problem Principle

4 Conclusion

Some important variables to characterize the system



Component i at time t characterized by $X_{i,t} = (E_{i,t}, A_{i,t})$ where:

$E_{i,t} \in \{0,1\}$: Regime of the component

- $E_{i,t} = 1$: Healthy
- $E_{i,t} = 0$: Broken

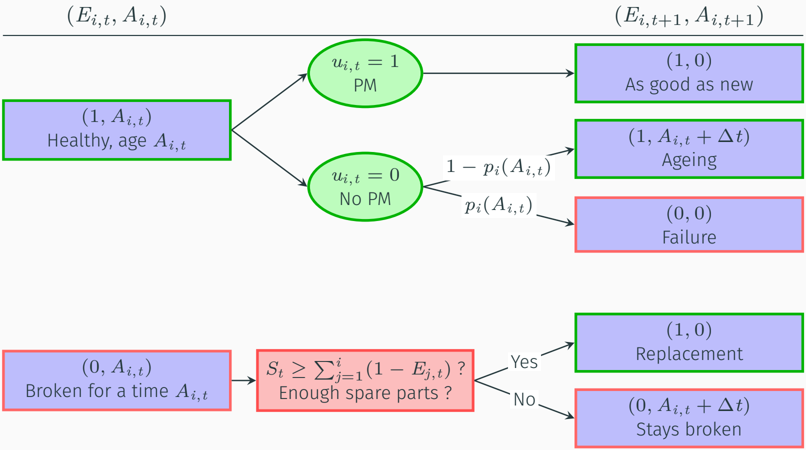
$A_{i,t} \in \mathbb{R}$:

- Age for a healthy component
- Time since last failure for a broken component

- $S_t \in \mathbb{N}$: Number of spare parts in the stock at time t
- $u_{i,t} \in \{0,1\}$: Control on component i at time t
  - $u_{i,t} = 0$: No preventive maintenance
  - $u_{i,t} = 1$: Preventive maintenance

Dynamics of the components

- Time discretized with time step Δt
- Dynamics of a component from time step t to $t+1$:



Maintenance optimization problem

$$\min_{(X,S,u)\in\mathcal{X}\times\mathcal{S}\times\mathbb{U}} \mathbb{E}\Bigg(\sum_{i=1}^{n} \overbrace{\sum_{t=0}^{T} j_{i,t}(X_{i,t}, u_{i,t})}^{\text{Maintenance cost } j_i(X_i,u_i)} + \overbrace{\sum_{t=0}^{T} j_t^{FO}(X_{1,t},\dots,X_{n,t})}^{\text{Forced outage cost } j^{FO}(X_1,\dots,X_n)}\Bigg)$$

$$\text{s.t.} \quad \underbrace{X_{i,t+1} = f_i^X(X_{i,t}, S_t, u_{i,t}, W_{i,t+1})}_{\text{Dynamics of component } i}, \quad X_{i,0} = x_i \quad \forall t, \forall i$$

$$\underbrace{S_{t+1} = f^S(X_{1,t},\dots,X_{n,t}, S_t)}_{\text{Dynamics of the stock}}, \quad S_0 = s \quad \forall t$$

where $(W_{i,t})_{t=1,\dots,T}^{i=1,\dots,n}$ are random variables that model failure scenarios.

- Maintenance cost: additive in time and components
- Forced outage cost: additive in time, coupling the components

Outline

1 Blackbox methods for optimal maintenance scheduling

2 A decomposition by prediction for the maintenance problem

2.1 Formalization of the maintenance optimization problem

2.2 A decomposition method component by component

2.3 Computational results on the 80-component industrial case

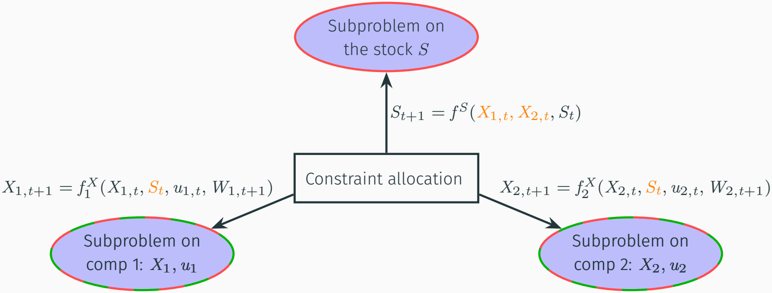
3 Contributions on the stochastic Auxiliary Problem Principle

4 Conclusion

Sketch of a decomposition method by component

General idea

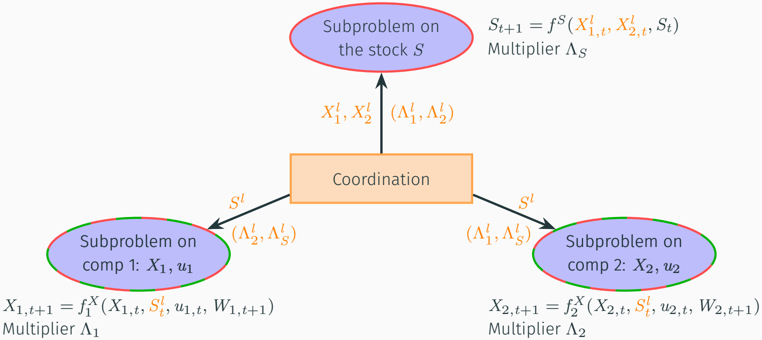
Use a decomposition by interaction prediction [MMT70] to iteratively find the best maintenance policy separately for each component and coordinate the components.



Sketch of a decomposition method by component

General idea

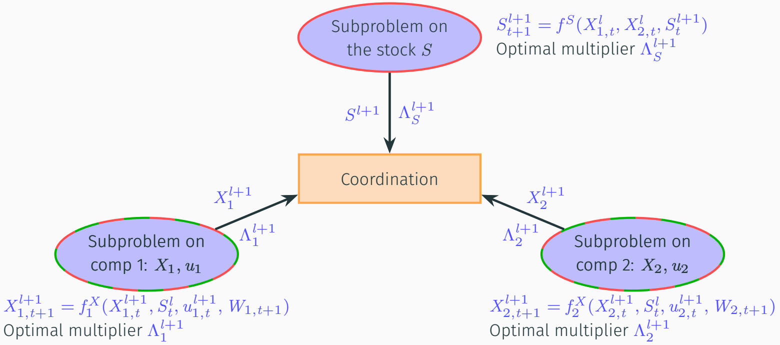
Use a decomposition by interaction prediction [MMT70] to iteratively find the best maintenance policy separately for each component and coordinate the components.



Sketch of a decomposition method by component

General idea

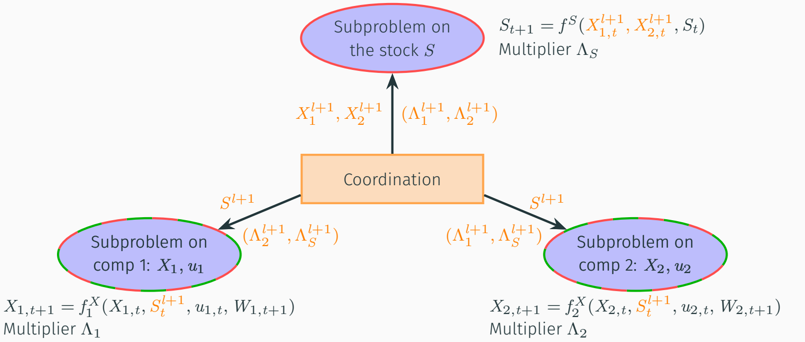
Use a decomposition by interaction prediction [MMT70] to iteratively find the best maintenance policy separately for each component and coordinate the components.



Sketch of a decomposition method by component

General idea

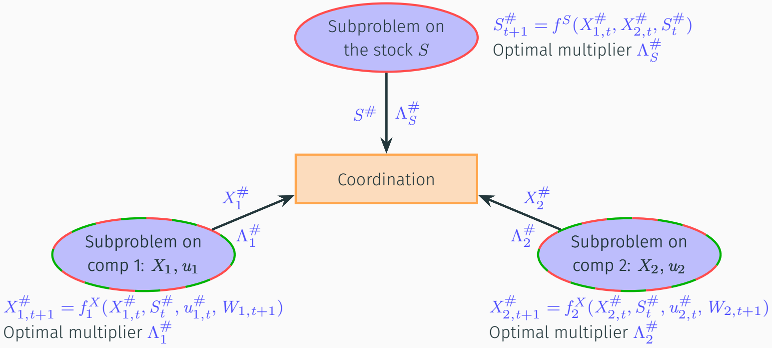
Use a decomposition by interaction prediction [MMT70] to iteratively find the best maintenance policy separately for each component and coordinate the components.



Sketch of a decomposition method by component

General idea

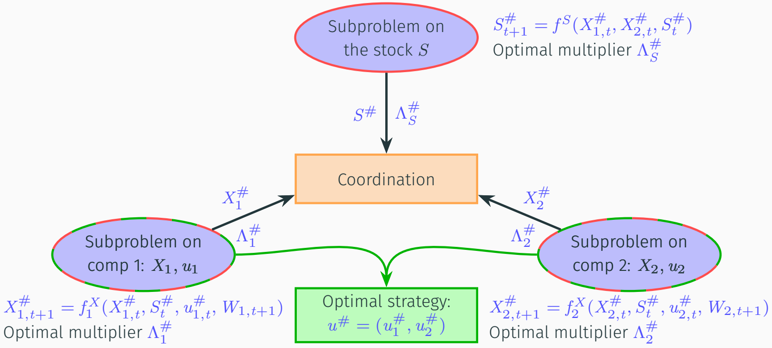
Use a decomposition by interaction prediction [MMT70] to iteratively find the best maintenance policy separately for each component and coordinate the components.



Sketch of a decomposition method by component

General idea

Use a decomposition by interaction prediction [MMT70] to iteratively find the best maintenance policy separately for each component and coordinate the components.



The auxiliary problem principle [Coh80] for a decomposition by component

- Choice of an auxiliary problem that is decomposable into independent subproblems

- Subproblem on component i at iteration $l+1$:

$$\min_{(X_i,u_i)\in\mathcal{X}_i\times\mathcal{U}_i} \mathbb{E}\left(j_i(X_i,u_i) + j^{FO}(X_1^l,\dots,X_i,\dots,X_n^l)\right) + \text{coordination terms}\left(\Lambda_1^l,\dots,\Lambda_{i-1}^l,\Lambda_{i+1}^l,\dots,\Lambda_n^l,\Lambda_S^l\right)$$

$$\text{s.t. } X_{i,t+1} = f_i^X(X_{i,t}, S_t^l, u_{i,t}, W_{i,t+1}), \quad \forall t$$

- Subproblem on the stock S:

$$\min_{S\in\mathcal{S}} \text{coordination terms}\left(\Lambda_1^l,\dots,\Lambda_n^l\right)$$

$$\text{s.t. } S_{t+1} = f^S(X_{1,t}^l,\dots,X_{n,t}^l,S_t), \quad \forall t$$

Fixed point algorithm for the decomposition by component

- Original problem: dimension nT
- Decomposition: n problems of dimension T per iteration

Fixed point algorithm for the decomposition by component

- Original problem: dimension nT
- Decomposition: n problems of dimension T per iteration

Algorithm 2 Fixed point algorithm

Start with (X^0, S^0, u^0) and Λ^0, set $l = 0$

At iteration $l+1$:

- For component $i = 1, \ldots, n$ do:
  - Solve

$$\min_{(X_i,u_i)\in\mathcal{X}_i\times\mathcal{U}_i} \mathbb{E}\Big(j_i(X_i,u_i) + j^{FO}(X_1^l,\ldots,X_i,\ldots,X_n^l)\Big) + \text{coordination terms}$$

$$\text{s.t. } X_{i,t+1} = f_i^X(X_{i,t}, S_t^l, u_{i,t}, W_{i,t+1}), \quad \forall t$$

    with any method (here with the blackbox optimization algorithm MADS [AD06]), solution (X_i^{l+1}, u_i^{l+1})
  - Compute an optimal multiplier Λ_i^{l+1} for the constraint using the adjoint state
- Similarly for the stock, solution S^{l+1} and optimal multiplier Λ_S^{l+1}

Stop if max number L of iterations reached, else $l \leftarrow l+1$ and start new iteration

Using a variational method in a discrete case

The fixed point algorithm is based on variational techniques:

- Gradient of the system dynamics appears in the coordination terms
- Gradient of the cost appears in the multiplier update step

Using a variational method in a discrete case

The fixed point algorithm is based on variational techniques:

- Gradient of the system dynamics appears in the coordination terms
- Gradient of the cost appears in the multiplier update step

But the system is characterized by integer variables, they are relaxed:

- Regime of the component: $E_{i,t} \in [0, 1]$
- Number of spare parts: $S_t \in \mathbb{R}_+$
- Controls: $u_{i,t} \in [0, 1]$

Using a variational method in a discrete case

The fixed point algorithm is based on variational techniques:

- Gradient of the system dynamics appears in the coordination terms
- Gradient of the cost appears in the multiplier update step

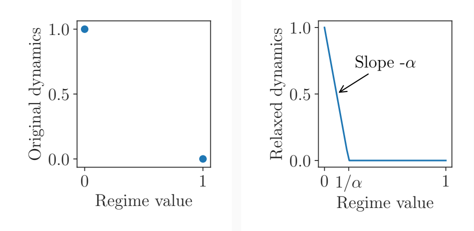
But the system is characterized by integer variables, they are relaxed:

- Regime of the component: $E_{i,t} \in [0, 1]$
- Number of spare parts: $S_t \in \mathbb{R}_+$
- Controls: $u_{i,t} \in [0, 1]$

The dynamics is non-smooth, it is also relaxed:

- Relaxation controlled by a parameter α

Example for the assertion:
`If the component is broken`



Parameter tuning: procedure description

Industrial goal

Apply the decomposition method on a maintenance problem with 80 components

Parameter tuning: procedure description

Industrial goal

Apply the decomposition method on a maintenance problem with 80 components

Some parameters need to be tuned:

- Relaxation controlled at iteration l by a parameter α^l
- Update of the relaxation parameter at each iteration: $\alpha^{l+1} = \alpha^l + \Delta\alpha$
  As $\alpha \to \infty$, the relaxed dynamics converges to the real one.
- Need to tune α^0 and $\Delta\alpha$
- Other parameters to tune: $\gamma^0, \Delta\gamma, r_x, r_s$ (not detailed in the talk)

Parameter tuning: procedure description

Industrial goal

Apply the decomposition method on a maintenance problem with 80 components

Some parameters need to be tuned:

- Relaxation controlled at iteration l by a parameter α^l
- Update of the relaxation parameter at each iteration: $\alpha^{l+1} = \alpha^l + \Delta\alpha$
  As $\alpha \to \infty$, the relaxed dynamics converges to the real one.
- Need to tune α^0 and $\Delta\alpha$
- Other parameters to tune: $\gamma^0, \Delta\gamma, r_x, r_s$ (not detailed in the talk)

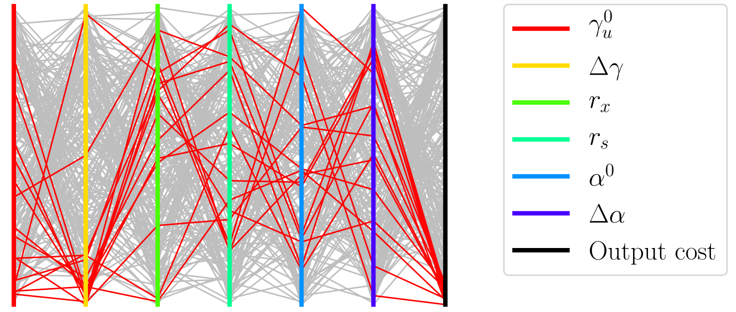
Tuning procedure for the vector of parameters $p = (\alpha^0, \Delta\alpha, \gamma^0, \Delta\gamma, r_x, r_s)$:

- Define bounds for the values of the parameters: $\alpha^0 \in [2, 200]$, $\Delta\alpha \in [0, 200]$, ...
- Draw 200 values of p with an optimized Latin Hypercube Sampling [DCI13]
- Optimization with each of the sampled values (i.e. 200 runs) on a smaller test case (10 components): computation time $\sim$ 4h

Using sensitivity analysis to tune an optimization algorithm I

Qualitative approach: Cobweb plots

$\rightarrow$ Visualize the best combinations of parameters for the optimization



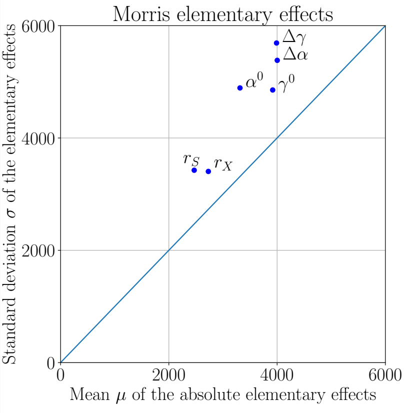
Conclusion

No clear result, except for $\Delta\gamma$ and r_x

Using sensitivity analysis to tune an optimization algorithm II

Quantitative approach: the Morris method [Mor91]

→ Screening method: sensitivity of the optimization quantified by elementary effects



- Mean of the elementary effects μ:

→ Quantifies the influence of a parameter on the result of the optimization

- Standard deviation σ of the elementary effects:

→ Measures the non-linear effects and the interactions between parameters on the result of the optimization

Conclusion

No screening possible, all inputs are influential with non linear/interaction effects

Parameter tuning: conclusion

Tuning procedure for the vector of parameters $p = (\alpha^0, \Delta\alpha, \gamma^0, \Delta\gamma, r_X, r_S)$:

- Define bounds for the values of the parameters: $\alpha^0 \in [2, 200]$, $\Delta\alpha \in [0, 200]$, ...
- Draw 200 values of p with an optimized Latin Hypercube Sampling
- Optimization with each of the sampled values (i.e. 200 runs) on a smaller test case (10 components): computation time $\sim$ 4h

Final choice

For the 80-component case, we use the value of p that gives the best results on the 10-component case.

Outline

1 Blackbox methods for optimal maintenance scheduling

2 A decomposition by prediction for the maintenance problem

2.1 Formalization of the maintenance optimization problem

2.2 A decomposition method component by component

2.3 Computational results on the 80-component industrial case

3 Contributions on the stochastic Auxiliary Problem Principle

4 Conclusion

Description of the industrial case

Parameter	Value		
Number of components n	80		
Initial number of spare parts S_0	16		
Horizon T	40 years		
Time of supply for the spare parts	2 years		
Discount factor	0.08		
Yearly forced outage cost	10000 k€/ year		
	Comp. 1	Comp. 2	Comp. $i \geq 3$
PM cost	50 k€	50 k€	50 k€
CM cost	100 k€	250 k€	200 k€
Failure distribution	$\text{Weib}(2.3, 10)$	$\text{Weib}(4, 20)$	$\text{Weib}(3, 10)$
Mean time to failure	8.85 years	18.13 years	8.93 years

1 maintenance decision each year for each component:

$$\Rightarrow \text{Problem in dimension } 80 \times 40 = 3200$$

Reference algorithm : MADS applied directly to the original optimization problem

Sample Average Approximation

Original problem:

$$\min_{(X,S,u)\in\mathcal{X}\times\mathcal{S}\times\mathbb{U}} \mathbb{E}\left(j(X,u)\right)$$
$$\text{s.t. } \Theta\left(X, S, u, W\right) = 0$$

- $j(X,u)$ represents the overall maintenance and forced outage costs
- $\Theta(X,S,u,W)$ represents the dynamics of the system

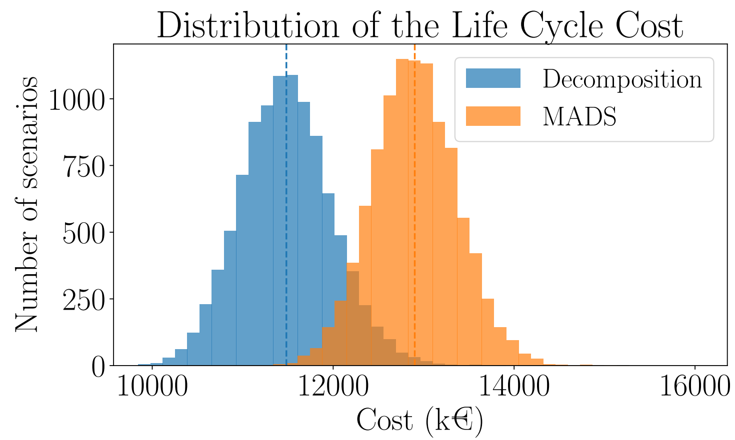
Sample Average Approximation with p Monte-Carlo scenarios $\omega_1,\dots,\omega_p$:

$$\min_{(X,S,u)\in\mathcal{X}\times\mathcal{S}\times\mathbb{U}} \frac{1}{p}\sum_{k=1}^{p} j(X(\omega_k),u)$$
$$\text{s.t. } \Theta\left(X(\omega_k), S(\omega_k), u, W(\omega_k)\right) = 0 \quad \forall k$$

Comparison of the Life Cycle Cost

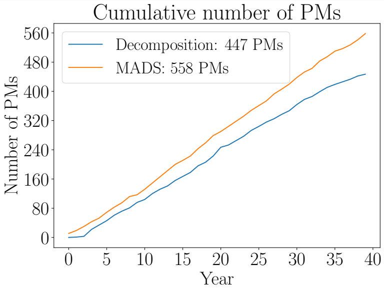
	Only CMs	MADS	Decomposition
Expected cost (k€)	46316	12902	11483

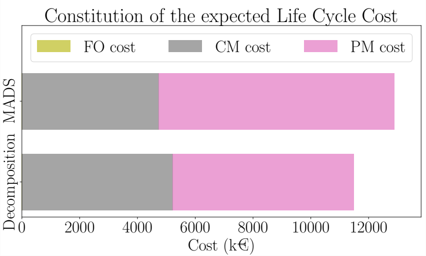
Gap MADS / Decomposition: 11%



Analysis of the maintenance strategies

	Decomposition	MADS
Mean number of PMs/component	5.6	7.0
Mean time between PMs	6.1 years	5.0 years
Mean number of failures/component	1.40	1.18
Number of forced outages	63/10000	1/10000





OUTLINE

1 Blackbox methods for optimal maintenance scheduling

2 A decomposition by prediction for the maintenance problem

3 Contributions on the stochastic Auxiliary Problem Principle

4 Conclusion

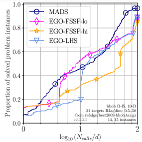
WORK SUMMARY: BLACKBOX METHODS AND DECOMPOSITION APPROACH

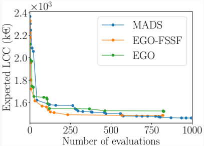
Blackbox optimization

- Use the simulation model VME: blackbox

- Contributions:

1. The EGO-FSSF algorithm: EGO with a sequential initial design and metamodel validation
2. Comparison with MADS:
   - On an academic benchmark
   - On an industrial maintenance problem





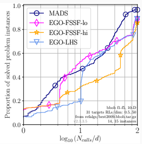
+ Plug and play: no modelling effort
- Small space of maintenance strategies
- System with few components

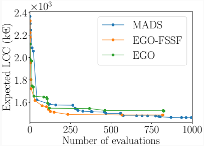
Work summary: Blackbox methods and decomposition approach

Blackbox optimization

- Use the simulation model VME: blackbox
- Contributions:

1. The EGO-FSSF algorithm: EGO with a sequential initial design and metamodel validation
2. Comparison with MADS:
   - On an academic benchmark
   - On an industrial maintenance problem



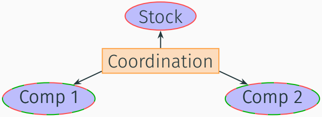


\+ Plug and play: no modelling effort
\- Small space of maintenance strategies
\- System with few components

Stochastic optimal control

- Analytical expression of the dynamics
  $\rightarrow$ We open the blackbox!
- Contributions in [BCCL20]:

1. Modelling of the maintenance problem
2. Resolution with a decomposition method



\- Problem-specific: modelling required
\+ General maintenance strategies
\+ Large-scale systems

Different optimization methods for different use cases

Blackbox optimization

- Periodic maintenance strategies
- Up to 10 components

→ Adapted for small systems when considering common maintenance operations such as lubrication

Decomposition method

- General maintenance strategies
- Scalable method

→ Adapted for large systems when considering exceptional maintenance operations (replacement of a large-size, expensive component)

Conclusion

Perspectives:

- Combine MADS and EGO for a more efficient blackbox method
- Solve more complex problems: add a control for the stock management strategy, consider degraded states for the component
- Try a stochastic approximation algorithm: the stochastic APP
- Could we apply the decomposition methodology in a robust optimization framework?

[AD06] C. Audet and J. E. Dennis, Jr.
Mesh Adaptive Direct Search Algorithms for Constrained Optimization.
SIAM Journal on Optimization, 17(1):188–217, 2006.

[Coh80] G. Cohen.
Auxiliary problem principle and decomposition of optimization problems.
Journal of Optimization Theory and Applications, 32(3):277–305, 1980.

[DCI13] G. Damblin, M. Couplet, and B. Iooss.
Numerical studies of space-filling designs: optimization of Latin Hypercube Samples and subprojection properties.
Journal of Simulation, 7(4):276–289, 2013.

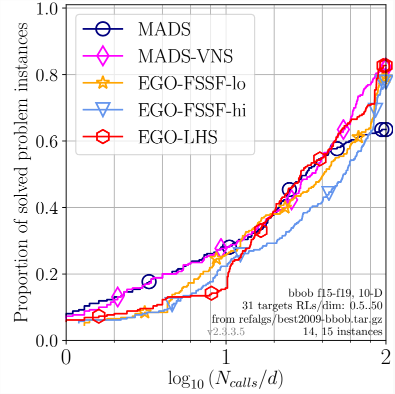
[JSW98] Donald R. Jones, Matthias Schonlau, and William J. Welch.
Efficient Global Optimization of Expensive Black-Box Functions.
Journal of Global optimization, 13(4):455–492, 1998.

[MMT70] M. D. Mesarović, D. Macko, and Y. Takahara.
Theory of Hierarchical, Multilevel Systems, volume 68 of Mathematics in Science and Engineering.
Academic Press, 1970.

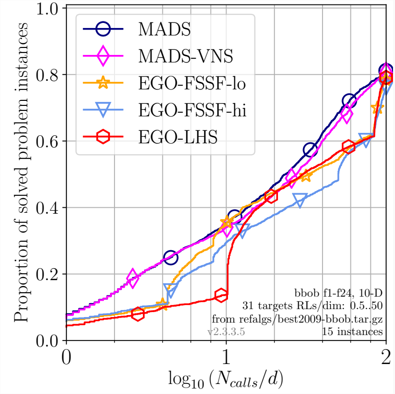
[Mor91] M. D. Morris.
Factorial Sampling Plans for Preliminary Computational Experiments.
Technometrics, 33(2):161–174, 1991.

Thank you for your attention!

Benchmark of EGO and MADS on the COCO platform



Results on multimodal functions



Results over all functions of the benchmark

Morris method cheat sheet

Denote by $p = (p_1, \ldots, p_l)$ the vector of parameters

- n randomized one-at-a-time experiments
- Elementary effect while perturbating p_i in experiment j:

$$d_i^{(j)}(p^{(j)}) = \frac{\mathcal{A}(p^{(j)} + \delta e_i) - \mathcal{A}(p^{(j)})}{\delta}$$

  with $p^{(j)}$ the value of the vector of parameters in the j-th experiment, $\mathcal{A}$ the model output (the optimization output in our case) and e_i the i-th vector of the canonical basis of $\mathbb{R}^l$.

We define two indices for each parameter p_i:

- Mean index:

$$\mu_i = \mathbb{E}\left(|d_i^{(j)}|\right) \simeq \frac{1}{n}\sum_{j=1}^{n}|d_i^{(j)}|$$

- Standard deviation index:

$$\sigma_i = \sqrt{\mathrm{Var}\left(d_i^{(j)}\right)} \simeq \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(d_i^{(j)} - \frac{1}{n}\sum_{j=1}^{n} d_i^{(j)}\right)^2}$$