G-SCOP
SCIENCES POUR LA CONCEPTION, L'OPTIMISATION ET LA PRODUCTION

Scheduling with Component Health Index: Complexity Study

E. Foussard   M.-L. Espinouse   G. Mounié   M. Nattaf

Univ. Grenoble Alpes, CNRS, Grenoble INP, G-SCOP, LIG, 38000 Grenoble, France

April 2022

PERSYVAL-Lab

UMR 5272

GRENOBLE INP UGA

cnrs

UGA Université Grenoble Alpes

Who am I?

2nd year PhD student

- G-SCOP ROSP with M.-L. Espinouse and M. Nattaf
- LIG Datamove with G. Mounié: one day per week, usually Tuesday (office 428)

Research interests:

- Planning and Scheduling problems
- MILP, Constraint Programming, Heuristics and Approximations, Complexity theory...
- Application to industrial maintenance planning

General problem

Machine with G components
$\rightarrow$ Health index $H_g \in [0, 100]$ per component

Set of jobs $\mathcal{N}$
$\rightarrow$ Duration p_i
$\rightarrow$ Wear w_{ig}
$\rightarrow$ Health thresholds s_{ig}

Maintenance Operations $\mathcal{M}$
$\rightarrow$ Duration p_m
$\rightarrow$ Regeneration reg_{gm} on component g

Objectives: C_{max}, $\sum C_j$...

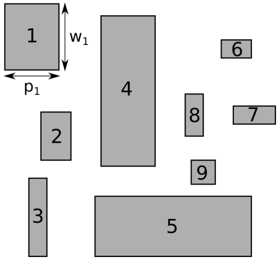
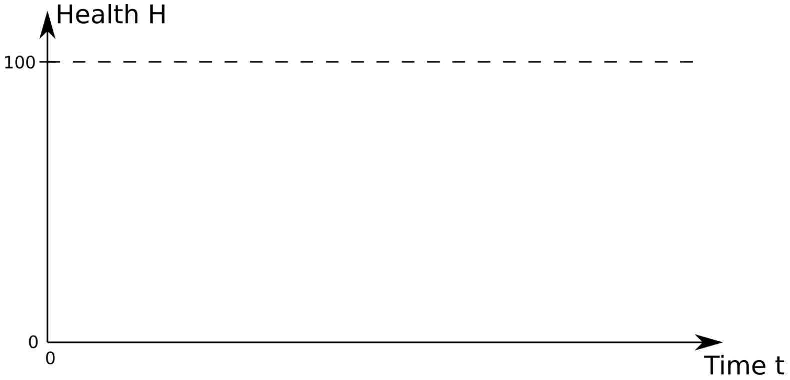
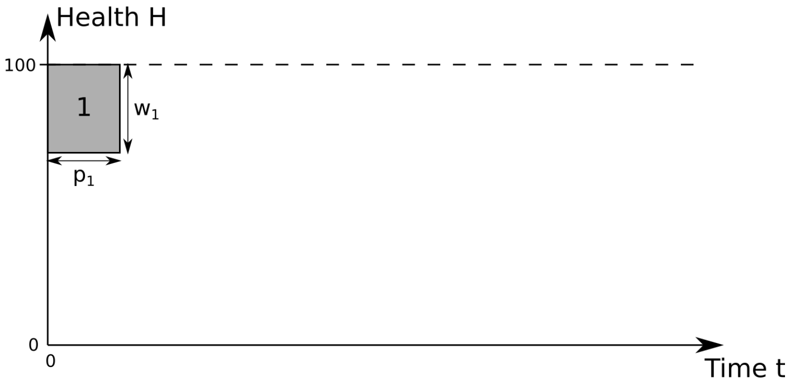


Figure: Example for one component without thresholds



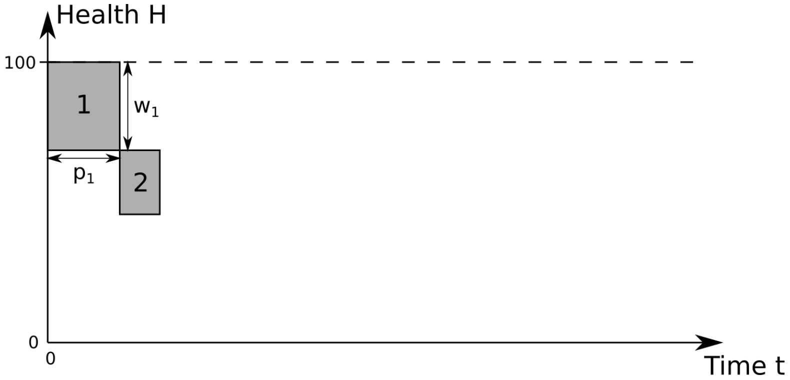
Hypotheses: 1 component, perfect maintenance, no thresholds

Greedy example



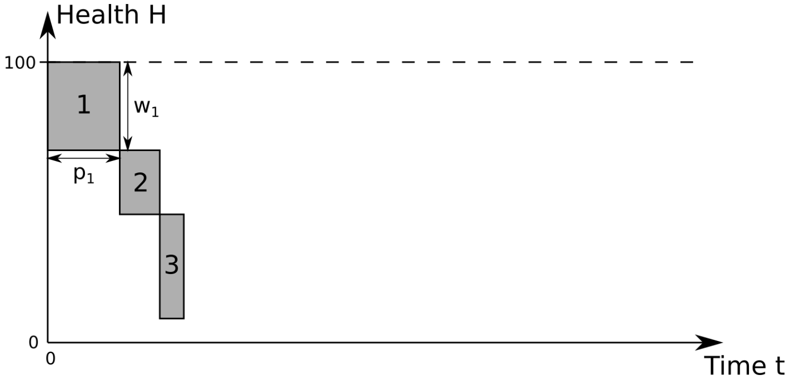
Hypotheses: 1 component, perfect maintenance, no thresholds

Greedy example



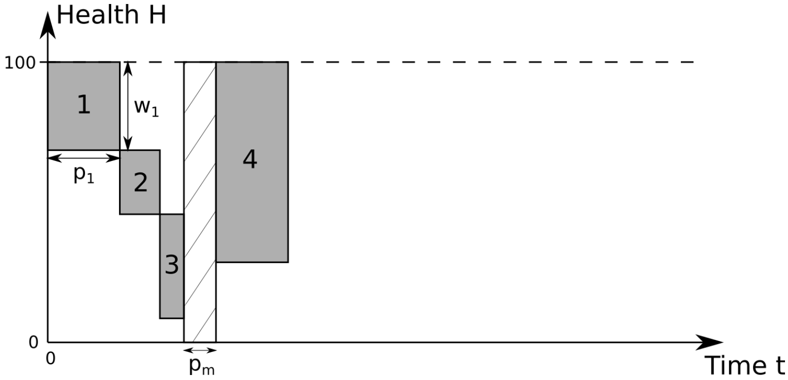
Hypotheses: 1 component, perfect maintenance, no thresholds

Greedy example



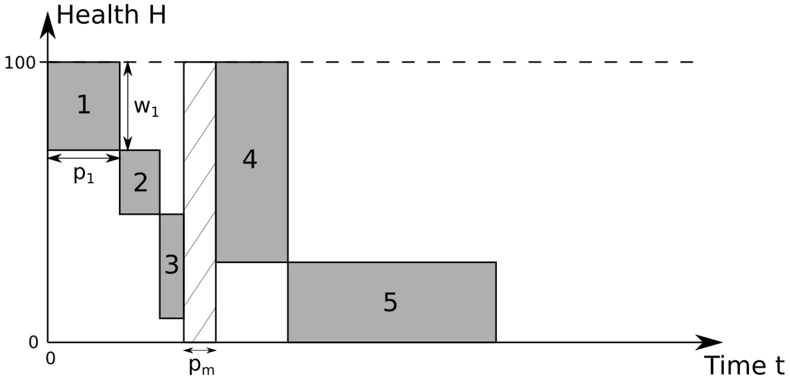
Hypotheses: 1 component, perfect maintenance, no thresholds

Greedy example



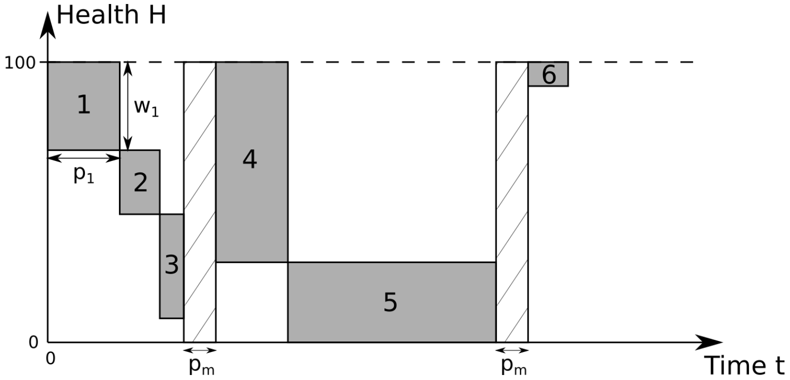
Hypotheses: 1 component, perfect maintenance, no thresholds

Greedy example



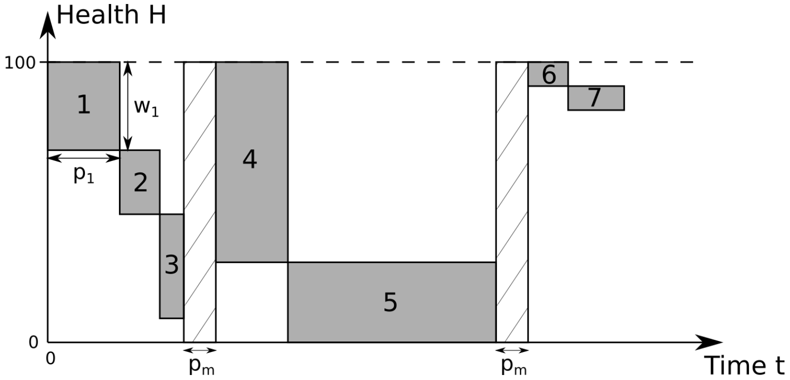
Hypotheses: 1 component, perfect maintenance, no thresholds

Greedy example



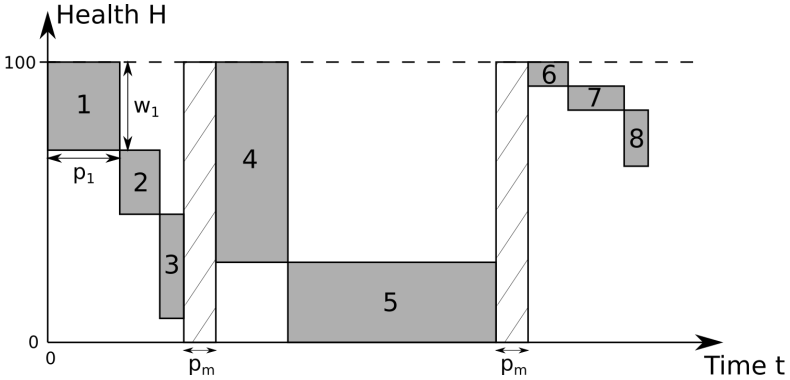
Hypotheses: 1 component, perfect maintenance, no thresholds

Greedy example



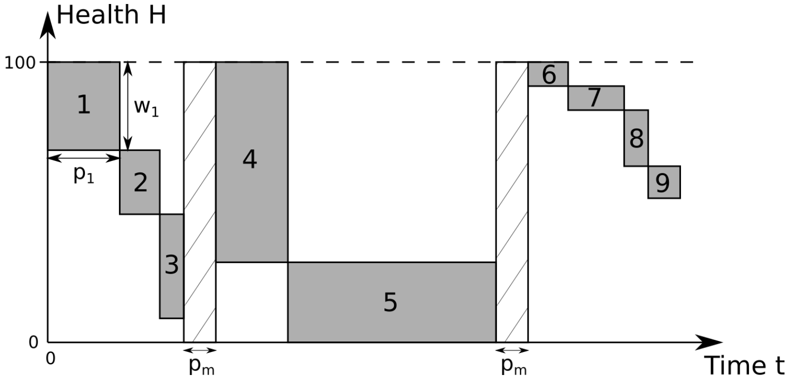
Hypotheses: 1 component, perfect maintenance, no thresholds

Greedy example



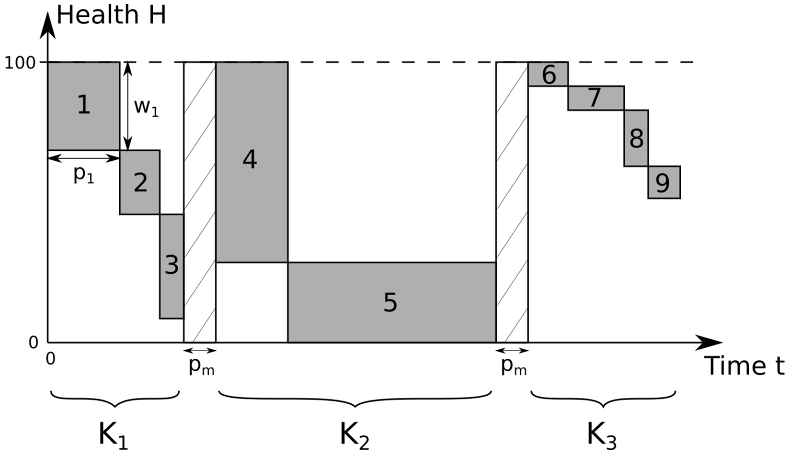
Hypotheses: 1 component, perfect maintenance, no thresholds

Greedy example



Hypotheses: 1 component, perfect maintenance, no thresholds

Greedy example



Hypotheses: 1 component, perfect maintenance, no thresholds

Scheduling with unavailabilties

Literature reviews: Kaabi and Harath (2014), Ma et al. (2010)

Non-resumable case:[1]:

- Fixed unavailabilities: Lee (1996), Sadfi et al. (2005)
- Periodic unavailabilities: Qi et al. (1997)

⇒ Temporal approach of maintenance scheduling

Main originalities

- Multi-component problem
- Time/wear decoupling ⇒ Predictive approach

[1] tasks may not be interrupted by maintenance

Degradation models

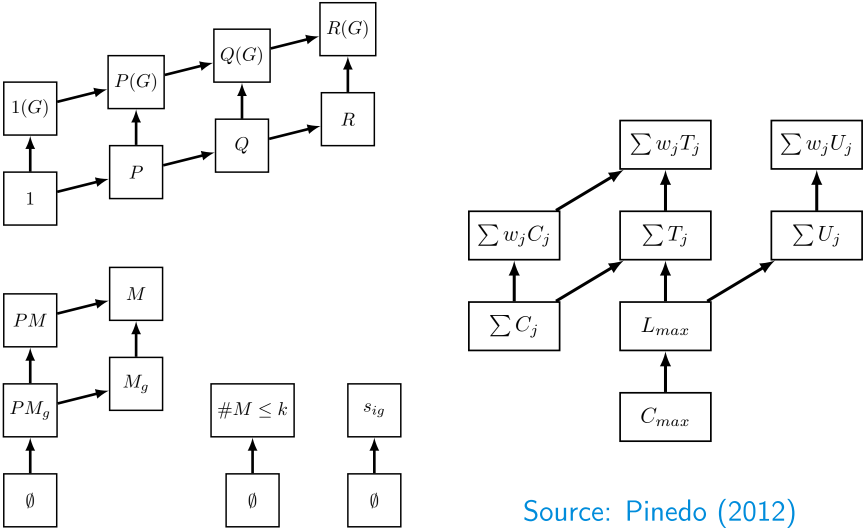
Classical approaches:

- Virtual Age-based stochastic approaches: de Jonge and Scarf (2020), Moghaddam and Usher (2011)
- Rate-modifying activities: Strusevich and Rustogi (2017)

Equipment Health Index:

- Kao et al. (2018)
- Master thesis of L. Penz (Prix Master ROADEF 2021)

Reductions



Source: Pinedo (2012)

Complexity study

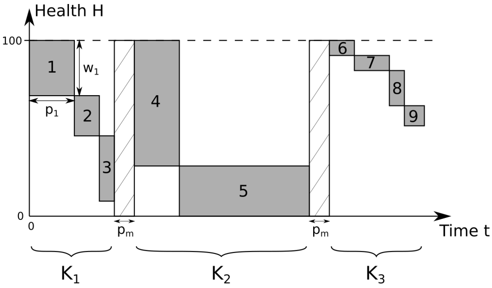
Machine with ~~G~~ 1 composant
→ Health index $H \in [0, 100]$ ~~per component~~

Set of jobs $\mathcal{N}$
→ Duration p_i
→ Wear w_i
→ ~~Health thresholds s_{ig}~~

Maintenance operations $\mathcal{M}$
→ Duration p_m
→ ~~Regeneration reg_{gm} on component g~~ Perfect maintenance on the unique composant

Existence of a feasible solution

No limits on the number of maintenance operations ⇒ Greedy algorithm



Bounded amount of maintenance operations ⇒ Strong sense NP-complete

Reduction from 3-PARTITION

3-PARTITION (Garey & Johnson 1990)

- Set A, $|A| = 3m$
- Size $s(a) \in \mathbb{Z}$ per object
- Bound B, such that $B/4 < s(a) < B/2$ and $\sum_{a \in A} s(a) = mB$.

Is there a partition $A_1 \ldots A_m$ such that for all $1 < i < m$, $\sum_{a \in A_i} s(a) = B$?

Reduction:

1 component, perfect maintenance of duration $p_m = 1$

3m tasks:

- duration $p_a = 1$
- wear $w_a = \frac{100 \cdot s(a)}{B}$

Is there a feasible schedule with at most $m - 1$ maintenance operations?

Example

Instance of 3-PARTITION:

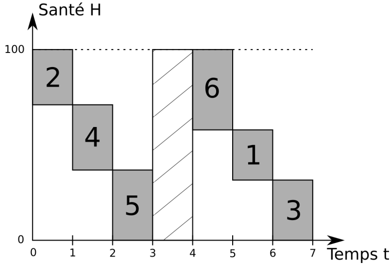
a	1	2	3	4	5	6
s(a)	13	14	15	16	20	22

avec B = 50

Corresponding instance:

a	1	2	3	4	5	6
p_a	1	1	1	1	1	1
w_a	26	28	30	32	40	44

with a single maintenance operation of duration $p_m = 1$



$A_1 = \{2, 4, 5\}$ and $A_2 = \{1, 3, 6\}$ is a valid partition.

Proof

$\Rightarrow$ If $A_1 ... A_m$ valid partition, reproduce the partition by affecting the tasks to the m corresponding batches ($m-1$ maintenance operations)

$$\forall i \in \{1 ... m\} \quad \sum_{a \in A_i} w_a = \frac{100}{B} \cdot \sum_{a \in A_i} s(a) = 100$$

$\Leftarrow$ If there exists a feasible schedule with $m-1$ maintenance operations (thus m batches).

$$\begin{cases} B/4 < s(a) < B/2 \\ \sum_{a \in A} s(a) = mB \end{cases} \Rightarrow \text{Batches of 3 jobs st} \sum_{a} w_a = 100$$

A valid partition can be extracted from the batches.

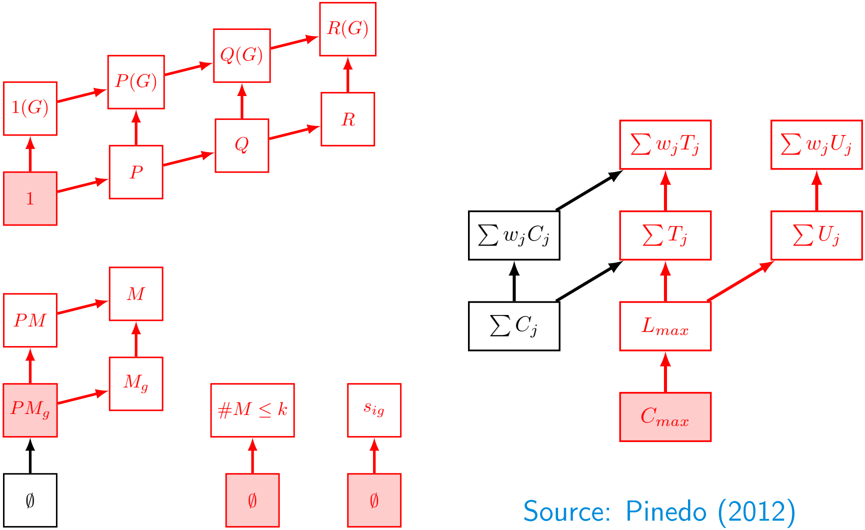
C_{max} (#maint. unbounded)

Corresponding decision problem equivalent to the existence of feasible schedule with limited maintenance => Strong sense NP-hard even with 1 component.

Relations with Bin-Packing $\Rightarrow$ Bin-Packing approximation algorithms have a performance ratio $1 + \rho_{BP}$ if $\sum p_i \geq p_m$

Maintenance operations minimization $\Rightarrow$ $(2 - \epsilon)$-inapproximable

Reductions



Source: Pinedo (2012)

$\sum C_j$ (#maint. unbounded)

Strong sense NP-hard even with 1 component (reduction from Qi et al. 1997).

One known polynomial sub case: constant wear $\rightarrow$ SPT optimal

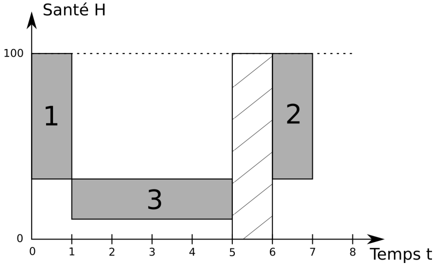
Known properties on the shape of the solutions:

- SPT order within each batch
- Batches ordered by non-decreasing $\frac{p_m + p(K_i)}{|K_i|}$

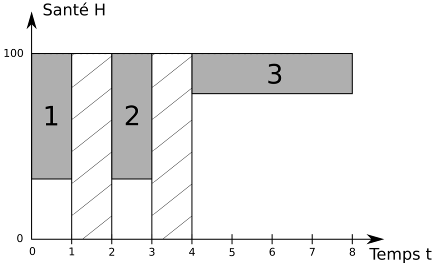
The "hard part" is defining the batches

$\sum C_j$ (#maint. unbounded)

- Number of scheduled maintenance operations not necessarily minimal $\Rightarrow$ not the same as C_{max}



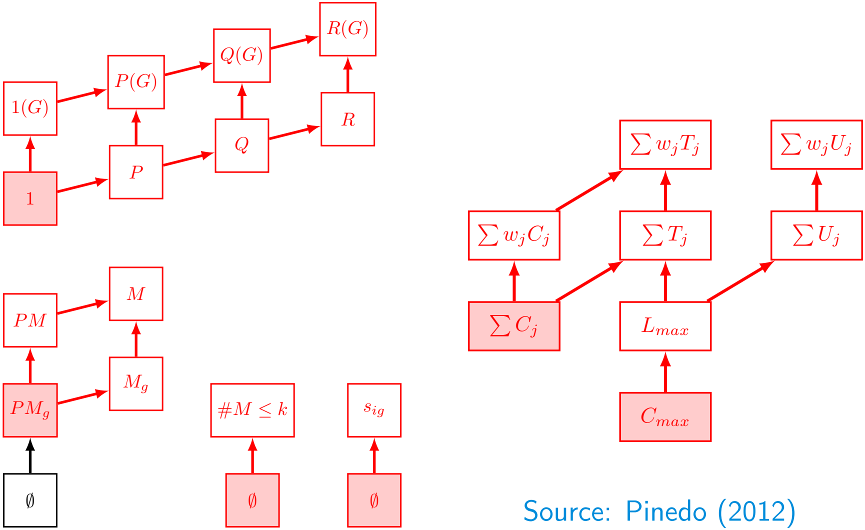
$\sum C_j = 13$



$\sum C_j = 12$

- Exchanging two tasks may break the schedule

Reductions



Source: Pinedo (2012)

Perspectives and ongoing projects

- Branch&Bound with column generation for the C_{max} problem with thresholds, study of classical BP heuristics (Gérémi Bridonneau's internship)
- Exact methods: study of MILP formulations and valid inequalities, Constraint Programming models...
- More results on approximation algorithms and performance of online algorithms

Thanks you for your attention!
Any questions?