Cours RTE 2022
Optimisation stochastique
◊
Méthodes de décomposition par blocs temporels

Mélange des techniques de décomposition

et de programmation dynamique par blocs

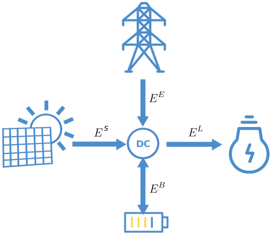
P. Carpentier, J.-Ph. Chancelier, M. De Lara , V. Leclère

École des Ponts ParisTech

ENSTA IP PARIS

Vendredi 22 avril 2022

A battery management problem over a long time horizon



We present a battery management problem over several years

- optimize long-term investment decisions
  — here the renewal of a battery in an energy system
- but the optimal long-term decisions highly depend on short-term operating decisions
  — here the way the battery is operated in real-time.

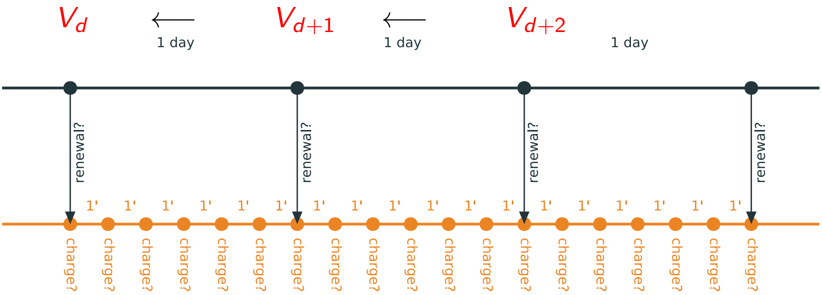
Battery management involves two time scales

- When to renew a battery (long term decision – day)?
- How to control the battery (short time decision – minute) and impact on aging?

Huge number of stages: $10,512,000 = \underbrace{7300}_{\text{days}} \times \underbrace{1440}_{\text{minutes}}$

Fortunately the problem displays a **two time scales** structure...

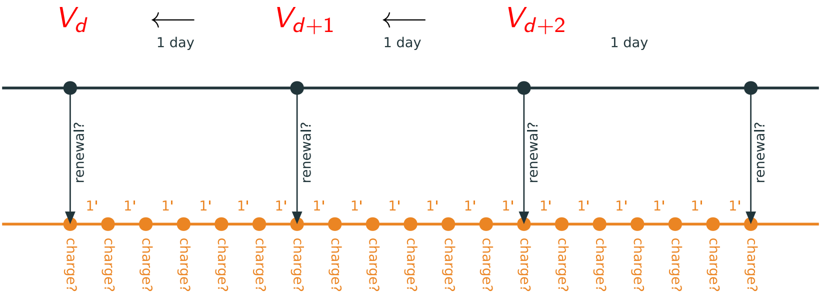
We will decompose the two time scales (day and minute) I



Fast time scale: minute (battery charge)
Slow time scale: day (battery renewal)

- What assumptions for a Bellman equation day by day?
- How to compute the daily Bellman value functions V_d, which involves an optimization problem at the fast time scale?

We will decompose the two time scales (day and minute) II

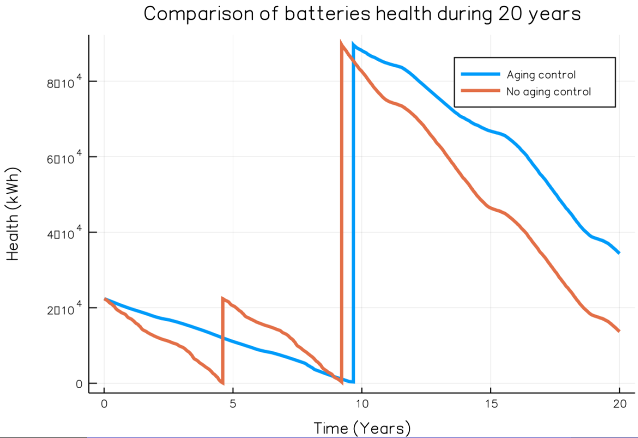


We propose numerical schemes providing upper and lower bounds on the family of daily Bellman value functions V_d

- Noise independence between days enables time decomposition (within a day, the fast time scale noises can be dependent)
- We resort to resource/price decomposition techniques to solve day by day subproblems.

It pays to control battery aging!

We simulate battery control policies over 20 years,
making an operating decision on the battery every minute



We introduce notations for two time scales

Time is described by to indices $(d, m) \in \mathbb{T}$

$$\mathbb{T} = \{0, \dots, D\} \times \{0, \dots, M\} \cup \{(D+1, 0)\}$$

1. Battery charge, decision every minute $m \in \{0, \dots, M\}$ of every day $d \in \{0, \dots, D\}$
   $\rightarrow$ Minutes in day d are $(d, 0)$, $(d, 1)$, $\dots$, (d, M)
2. Renewal of the battery, decision every day $d \in \{0, \dots, D+1\}$
   $\rightarrow$ Start of days are $(0, 0)$, $\dots$, $(d, 0)$, $\dots$, $(D+1, 0)$
3. Compatibility between days: $(d, M+1) = (d+1, 0)$
   It could be practical to add a (fictitious) time interval between $(d, M+1)$ and $(d+1, 0)$: we will not detail this point

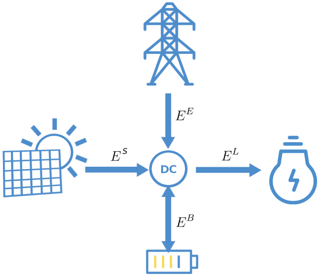
Equipped with the *lexicographical order*, $\mathbb{T}$ is a totally ordered set

$$(d, m) < (d', m') \iff (d < d') \vee (d = d' \wedge m < m')$$

Lecture outline

Outline of the presentation

Physical model: a home with load, solar panel and storage



- Two time scales uncertainties
  - $\mathbf{D}_{d,m}$: Net demand $(= \mathbf{E}^L_{d,m} - \mathbf{E}^S_{d,m})$
  - $\mathbf{P}^b_d$: Uncertain battery price
- Two time scales controls
  - $\mathbf{E}^B_{d,m}$: Battery charge/discharge
  - $\mathbf{E}^E_{d,m}$: National grid import
  - $\mathbf{V}_d$: Battery renewal
- Two time scales states
  - $\mathbf{S}_{d,m}$: Battery state of charge
  - $\mathbf{H}_{d,m}$: Battery health
  - $\mathbf{C}_d$: Battery capacity

Fast time scale: system operation

- The national grid import ensures energy balance

$$\mathbf{E}^{E}_{d,m} = \mathbf{D}_{d,m} + (\mathbf{E}^{B}_{d,m})^{+} - (\mathbf{E}^{B}_{d,m})^{-}$$

  and induces an operating cost

$$\pi^{e}_{d,m} \times \big(\mathbf{D}_{d,m} + (\mathbf{E}^{B}_{d,m})^{+} - (\mathbf{E}^{B}_{d,m})^{-}\big)$$

- The battery state of charge and health evolve at the fast time scale

$$\mathbf{S}_{d,m+1} = \mathbf{S}_{d,m} + \rho^{\mathrm{c}}(\mathbf{E}^{B}_{d,m})^{+} - \rho^{\mathrm{d}}(\mathbf{E}^{B}_{d,m})^{-}$$
$$\mathbf{H}_{d,m+1} = \mathbf{H}_{d,m} - (\mathbf{E}^{B}_{d,m})^{+} - (\mathbf{E}^{B}_{d,m})^{-}$$

  whereas the battery capacity remains unchanged at this scale

$$\mathbf{C}_{d,m+1} = \mathbf{C}_{d}$$

$$\longrightarrow \quad (\mathbf{S}_{d,m+1}, \mathbf{H}_{d,m+1}, \mathbf{C}_{d,m+1}) = \varphi(\mathbf{S}_{d,m}, \mathbf{H}_{d,m}, \mathbf{C}_{d}, \mathbf{E}^{B}_{d,m})$$

Slow time scale: renewal model

- At the end of every day d, we can buy a new battery at cost $\mathbf{P}_d^b \times \mathbf{V}_d$

$$\text{Storage capacity: } \mathbf{C}_{d+1} = \begin{cases} \mathbf{V}_d \ , & \text{if } \mathbf{V}_d > 0 \\ \mathbf{C}_d \ , & \text{otherwise} \end{cases}$$

- A new battery can make a maximum number of cycles $N_c(\mathbf{V}_d)$:

$$\text{Storage health: } \mathbf{H}_{d+1,0} = \begin{cases} 2 \times N_c(\mathbf{V}_d) \times \mathbf{V}_d \ , & \text{if } \mathbf{V}_d > 0 \\ \mathbf{H}_{d,M} \ , & \text{otherwise} \end{cases}$$

- A new battery is empty

$$\text{Storage state of charge: } \mathbf{S}_{d+1,0} = \begin{cases} 0 \ , & \text{if } \mathbf{V}_d > 0 \\ \mathbf{S}_{d,M} \ , & \text{otherwise} \end{cases}$$

$$\longrightarrow \quad (\mathbf{S}_{d+1,0}, \mathbf{H}_{d+1,0}, \mathbf{C}_{d+1}) = \psi(\mathbf{S}_{d,M}, \mathbf{H}_{d,M}, \mathbf{C}_d, \mathbf{V}_d)$$

We build an optimization problem at the daily scale

- Uncertainties

$$\mathbf{W}_d = \left(\mathbf{D}_{d,0}, \ldots, \mathbf{D}_{d,m}, \ldots, \mathbf{D}_{d,M-1}, \begin{pmatrix} \mathbf{D}_{d,M} \\ \mathbf{P}^b_d \end{pmatrix}\right)$$

- Controls

$$\mathbf{U}_d = \left(\mathbf{E}^B_{d,0}, \ldots, \mathbf{E}^B_{d,m}, \ldots, \mathbf{E}^B_{d,M-1}, \begin{pmatrix} \mathbf{E}^B_{d,M} \\ \mathbf{V}_d \end{pmatrix}\right)$$

- States and dynamics

$$\mathbf{X}_d = \left(\mathbf{S}_{d,0}, \mathbf{H}_{d,0}, \mathbf{C}_d\right) \quad \text{and} \quad \mathbf{X}_{d+1} = \overbrace{f_d\big(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d\big)}^{\text{composition of } \varphi \text{ and } \psi}$$

- Objective to be minimized

$$\mathbb{E}\bigg(\sum_{d=0}^{D} \underbrace{\Big(\mathbf{P}^b_d \times \mathbf{V}_d + \sum_{m=0}^{M} \pi^e_{d,m} \times \big(\mathbf{D}_{d,m} + (\mathbf{E}^B_{d,m})^+ - (\mathbf{E}^B_{d,m})^-\big)\Big)}_{L_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d)} + K(\mathbf{X}_{D+1})\bigg)$$

Two time scales non standard stochastic control problem

We now write the associated stochastic multistage optimization problem, whose optimal value is V_0

$$\mathcal{P}^e : \quad V_0 = \min_{(\mathbf{X}_{0:D+1}, \mathbf{U}_{0:D})} \mathbb{E}\left(\sum_{d=0}^{D} L_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d) + K(\mathbf{X}_{D+1}) \right)$$

$$\text{s.t} \quad \mathbf{X}_0 = x_0 \ , \ \ \mathbf{X}_{d+1} = f_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d)$$

$$\mathbf{U}_d = (\mathbf{U}_{d,0}, \ldots, \mathbf{U}_{d,m}, \ldots, \mathbf{U}_{d,M})$$

$$\mathbf{W}_d = (\mathbf{W}_{d,0}, \ldots, \mathbf{W}_{d,m}, \ldots, \mathbf{W}_{d,M})$$

$$\sigma(\mathbf{U}_{d,m}) \subset \sigma\Big(\mathbf{W}_{d',m'}, \ (d', m') \leq (d, m)\Big)`$$

It is a non standard SOC problem: the nonanticipativity constraint is **written every minute** whereas dynamics is written **every day**!

Bellman equation with daily time blocks

Daily Independence Assumption

$\{\mathbf{W}_d\}_{d=0,\dots,D}$ is a sequence of daily independent random vectors

We set $V^e_{D+1} = K$ and

$$V^e_d(x) = \min_{(\mathbf{X}_{d+1},\mathbf{U}_d)} \mathbb{E}\Big[L_d(x,\mathbf{U}_d,\mathbf{W}_d) + V^e_{d+1}(\mathbf{X}_{d+1})\Big]$$

$$\text{s.t} \quad \mathbf{X}_{d+1} = f_d(x,\mathbf{U}_d,\mathbf{W}_d)$$

$$\sigma(\mathbf{U}_{d,m}) \subset \sigma(\mathbf{W}_{d,0},\dots,\mathbf{W}_{d,m}) \; , \quad \forall m \in \{0,\dots,M\}$$

Proposition (see [Carpentier, Chancelier, De Lara and Rigaut, 2018])

Under Daily Independence Assumption, $V^e_0(x_0)$ is the value V_0 of Problem $\mathcal{P}^e$

We introduce price/resource daily decompositions

The main practical difficulty is the huge number of stages ($D \times M = 10,512,000$)! To overcome this, we appeal to decomposition methods.

Decomposition is done on the dynamics $\mathbf{X}_{d+1} = f_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d)$

1. Resource decomposition: we choose resources (targets) $\mathbf{R}_{d+1}$ and we split the dynamic constraints in

$$\mathbf{X}_{d+1} = \mathbf{R}_{d+1} \ , \quad \mathbf{R}_{d+1} = f_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d)$$

2. Price decomposition: we choose prices (weights) $\mathbf{\Lambda}_{d+1}$ and we dualize the dynamic constraints

$$\left\langle \mathbf{\Lambda}_{d+1} \ , f_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d) - \mathbf{X}_{d+1} \right\rangle$$

Relaxation of the stochastic control problem

A new difficulty then appears, linked to resource decomposition

The optimization subproblems in the resource decomposition method involve equality constraints between random variables

$$\mathbf{R}_{d+1} = f_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d)$$

which are almost always impossible to satisfy for a given resource

To solve this new difficulty, we relax the optimization problem $\mathcal{P}^e$ by writing the dynamic constraints as inequality constraints

$$\mathbf{X}_{d+1} \leq f_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d)$$

that is, we enlarge the admissible set of the problem

Relaxation of the stochastic control problem

We consider the following relaxed optimization problem

$$\mathcal{P}^i : \quad V_0^i = \min_{(\mathbf{X}_{0:D+1}, \mathbf{U}_{0:D})} \mathbb{E}\left(\sum_{d=0}^{D} L_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d) + K(\mathbf{X}_{D+1})\right)$$

$$\text{s.t} \quad \mathbf{X}_{d+1} \leq f_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d)$$

$$\sigma(\mathbf{U}_{d,m}) \subset \sigma\big(\mathbf{W}_{d',m'},\ (d',m') \leq (d,m)\big)$$

$$\sigma(\mathbf{X}_{d+1}) \subset \sigma\big(\mathbf{W}_{d',m'},\ (d',m') \leq (d,M)\big)$$

and the associated sequence of value functions

$$V_d^i(x) = \min_{(\mathbf{X}_{d+1}, \mathbf{U}_d)} \mathbb{E}\Big[L_d(x, \mathbf{U}_d, \mathbf{W}_d) + V_{d+1}^i(\mathbf{X}_{d+1})\Big]$$

$$\text{s.t} \quad \mathbf{X}_{d+1} \leq f_d(x, \mathbf{U}_d, \mathbf{W}_d)$$

$$\sigma(\mathbf{U}_{d,m}) \subset \sigma(\mathbf{W}_{d,0}, \ldots, \mathbf{W}_{d,m})$$

$$\sigma(\mathbf{X}_{d+1}) \subset \sigma(\mathbf{W}_{d,0}, \ldots, \mathbf{W}_{d,M})$$

Equivalence between the initial and the relaxed problem

Proposition (see [Carpentier, Chancelier, De Lara and Rigaut, 2022])

The battery management problem $\mathcal{P}^e$ displays a monotonicity property, that is, the value functions V_d^e are nonincreasing

$$x \leq x' \quad \Longrightarrow \quad V_d^e(x) \geq V_d^e(x')$$

Proposition (see [Carpentier, Chancelier, De Lara and Rigaut, 2022])

The monotonicity property of the value functions V_d^e of Problem $\mathcal{P}^e$ implies

$$V_d^i = V_d^e \ , \ \ \forall d \in \{0, \ldots, D+1\}$$

Lecture outline

1. Two time scales battery management problem

2. Resource and price decomposition methods
   - Time blocks and resource decomposition
   - Time blocks and price decomposition
   - Producing fast time scale policies

3. Numerical results

We introduce price/resource daily decompositions

We present two decomposition algorithms to compute upper and lower bounds of the daily value functions V_d

Decomposition is done on the dynamics $\mathbf{X}_{d+1} \leq f_d(x, \mathbf{U}_d, \mathbf{W}_d)$

1. Resource decomposition: choosing deterministic resources (targets) r_{d+1} and splitting the dynamic constraints in

$$\mathbf{X}_{d+1} = r_{d+1} \; , \;\; r_{d+1} \leq f_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d)$$

   gives an upper bound of Problem $\mathcal{P}^e$

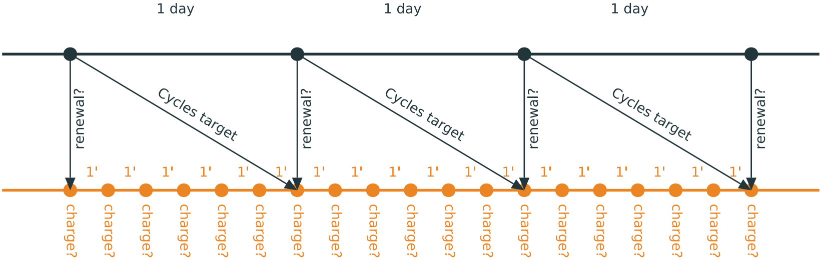
2. Price decomposition: choosing deterministic prices (weights) $\mathbf{\Lambda}_{d+1} \leq 0$ and dualizing the dynamic constraints

$$\langle \lambda_{d+1} \, , f_d(\mathbf{X}_d, \mathbf{U}_d, \mathbf{W}_d) - \mathbf{X}_{d+1} \rangle$$

   gives a lower bound of Problem $\mathcal{P}^e$

Lecture outline

Decomposing by imposing targets



Resource decomposition mechanism

$$
\begin{aligned}
V^e_d(x_d) &= \min_{(\mathbf{X}_{d+1},\mathbf{U}_d)} \mathbb{E}\Big[L_d(x_d,\mathbf{U}_d,\mathbf{W}_d) + V^e_{d+1}(\mathbf{X}_{d+1})\Big] \\
&\qquad \text{s.t} \quad \mathbf{X}_{d+1} = f_d(x_d,\mathbf{U}_d,\mathbf{W}_d) && \text{(Bellman equation)} \\
&= \min_{(\mathbf{X}_{d+1},\mathbf{U}_d)} \mathbb{E}\Big[L_d(x_d,\mathbf{U}_d,\mathbf{W}_d) + V^i_{d+1}(\mathbf{X}_{d+1})\Big] \\
&\qquad \text{s.t} \quad \mathbf{X}_{d+1} \leq f_d(x_d,\mathbf{U}_d,\mathbf{W}_d) && \text{(monotonicity)} \\
&= \min_{\mathbf{R}_{d+1}} \Big(\min_{\mathbf{U}_d} \mathbb{E}\Big[L_d(x_d,\mathbf{U}_d,\mathbf{W}_d) + V^i_{d+1}(\mathbf{R}_{d+1})\Big]\Big) \\
&\qquad \text{s.t} \quad \mathbf{R}_{d+1} \leq f_d(x_d,\mathbf{U}_d,\mathbf{W}_d) && \text{(stochastic resource)} \\
&\leq \min_{r_{d+1}} \Big(\min_{\mathbf{U}_d} \mathbb{E}\Big[L_d(x_d,\mathbf{U}_d,\mathbf{W}_d) + V^i_{d+1}(r_{d+1})\Big]\Big) \\
&\qquad \text{s.t} \quad r_{d+1} \leq f_d(x_d,\mathbf{U}_d,\mathbf{W}_d) && \text{(deterministic resource)} \\
&= \min_{r_{d+1}} \Big(\underbrace{\min_{\mathbf{U}_d} \Big(\mathbb{E}\Big[L_d(x_d,\mathbf{U}_d,\mathbf{W}_d)\Big] \text{ s.t } r_{d+1} \leq f_d(x_d,\mathbf{U}_d,\mathbf{W}_d)\Big)}_{L^{\mathrm{R}}_d(x_d,r_{d+1})} + V^i_{d+1}(r_{d+1})\Big)
\end{aligned}
$$

Relaxed deterministic resource decomposition

We introduce a relaxed deterministic resource intraday problem

$$L_d^{\mathrm{R}}(x_d, r_{d+1}) = \min_{\mathbf{U}_d} \ \mathbb{E}\Big[L_d(x_d, \mathbf{U}_d, \mathbf{W}_d)\Big]$$

$$\text{s.t} \quad f_d(x_d, \mathbf{U}_d, \mathbf{W}_d) \geq r_{d+1}$$

$$\sigma(\mathbf{U}_{d,m}) \subset \sigma(\mathbf{W}_{d,0:m})$$

and the associated Bellman recursion

$$\overline{V}_d^{\mathrm{R}}(x_d) = \min_{r_{d+1}} \ L_d^{\mathrm{R}}(x_d, r_{d+1}) + \overline{V}_{d+1}^{\mathrm{R}}(r_{d+1})$$

Proposition (see [Carpentier, Chancelier, De Lara and Rigaut, 2022])

Thanks to the monotonicity property, the value functions $\overline{V}_d^{\mathrm{R}}$ are upper bounds to the value functions V_d^e of Problem $\mathcal{P}^e$

$$\overline{V}_d^{\mathrm{R}} \geq V_d^e \ , \ \ \forall d \in \{0, \ldots, D+1\}$$

Efficiency of deterministic resource decomposition

$$\overbrace{\overline{V}_d^{\mathrm{R}}(x_d) = \min_{r_{d+1}} \underbrace{L_d^{\mathrm{R}}(x_d, r_{d+1})}_{\text{Hard to compute}} + \overline{V}_{d+1}^{\mathrm{R}}(r_{d+1})}^{\text{Easy to compute by dynamic programming}}$$

It is challenging to compute the intraday function value $L_d^{\mathrm{R}}(x_d, r_{d+1})$ for each couple (x_d, r_{d+1}) and each day d, but

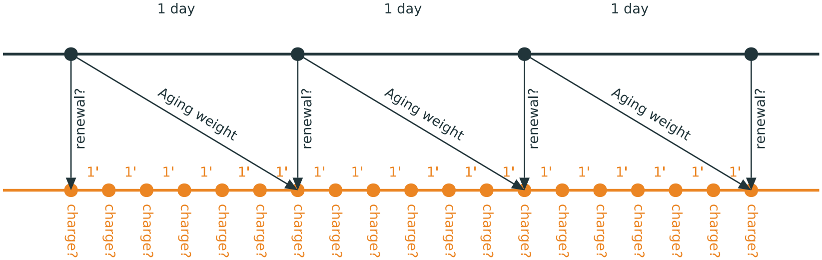
- we can exploit periodicity of the problem, e.g $L_d^{\mathrm{R}} = L_0^{\mathrm{R}}$
- for some components of the state, the intraday function L_d^{R} depends on $x_d - r_{d+1}$ rather than (x_d, r_{d+1})
- we can parallelize the computation of L_d^{R} on several days

Note that we can use any suitable method to solve the multistage intraday problems L_d^{R} (SDP, SDDP, scenario tree methods, PH,...)

Lecture outline

1. Two time scales battery management problem
2. Resource and price decomposition methods
   - Time blocks and resource decomposition
   - Time blocks and price decomposition
   - Producing fast time scale policies
3. Numerical results
   - Managing battery charge and health over 5 days
   - Managing battery charge, health and renewal over 20 years

Decomposing by applying weights



Price decomposition mechanism

$$
\begin{aligned}
V^e_d(x_d) = & \min_{(\mathbf{X}_{d+1},\mathbf{U}_d)} \mathbb{E}\Big[L_d(x_d,\mathbf{U}_d,\mathbf{W}_d) + V^e_{d+1}(\mathbf{X}_{d+1})\Big] \\
& \quad \text{s.t} \quad \mathbf{X}_{d+1} = f_d(x_d,\mathbf{U}_d,\mathbf{W}_d) \qquad \text{(Bellman equation)} \\
\geq & \max_{\mathbf{\Lambda}_{d+1}} \min_{(\mathbf{X}_{d+1},\mathbf{U}_d)} \mathbb{E}\Big[L_d(x_d,\mathbf{U}_d,\mathbf{W}_d) + V^e_{d+1}(\mathbf{X}_{d+1}) \\
& \qquad + \big\langle \mathbf{\Lambda}_{d+1}, f_d(x_d,\mathbf{U}_d,\mathbf{W}_d) - \mathbf{X}_{d+1}\big\rangle\Big] \qquad \text{(duality)} \\
= & \max_{\mathbf{\Lambda}_{d+1}} \min_{\mathbf{U}_d} \mathbb{E}\Big[L_d(x_d,\mathbf{U}_d,\mathbf{W}_d) + \big\langle \mathbf{\Lambda}_{d+1}, f_d(x_d,\mathbf{U}_d,\mathbf{W}_d)\big\rangle \\
& \qquad + \min_{\mathbf{X}_{d+1}} \big(- \langle \mathbf{\Lambda}_{d+1}, \mathbf{X}_{d+1}\rangle + V^e_{d+1}(\mathbf{X}_{d+1})\big)\Big] \qquad \text{(Fenchel)} \\
\geq & \max_{\lambda_{d+1}} \min_{\mathbf{U}_d} \mathbb{E}\Big[L_d(x_d,\mathbf{U}_d,\mathbf{W}_d) + \big\langle \lambda_{d+1}, f_d(x_d,\mathbf{U}_d,\mathbf{W}_d)\big\rangle\Big] \\
& \qquad - \big(V^e_{d+1}\big)^\star(\lambda_{d+1}) \qquad \text{(deterministic price)} \\
= & \max_{\lambda_{d+1}} \Bigg(\underbrace{\min_{\mathbf{U}_d} \mathbb{E}\Big[L_d(x_d,\mathbf{U}_d,\mathbf{W}_d) + \big\langle \lambda_{d+1}, f_d(x_d,\mathbf{U}_d,\mathbf{W}_d)\big\rangle\Big]}_{L^{\mathrm{P}}_d(x_d,\lambda_{d+1})} - \big(V^e_{d+1}\big)^\star(\lambda_{d+1})\Bigg)
\end{aligned}
$$

Relaxed deterministic price decomposition

We introduce a relaxed deterministic price intraday problem

$$L^{\mathrm{P}}_d(x_d, \lambda_{d+1}) = \min_{\mathbf{U}_d} \mathbb{E}\Big[L_d(x_d, \mathbf{U}_d, \mathbf{W}_d) + \langle \lambda_{d+1}, f_d(x_d, \mathbf{U}_d, \mathbf{W}_d)\rangle\Big]$$

$$\text{s.t.} \quad \sigma(\mathbf{U}_{d,m}) \subset \sigma(\mathbf{W}_{d,0:m})$$

and the associated Bellman recursion[1]

$$\underline{V}^{\mathrm{P}}_d(x_d) = \max_{\lambda_{d+1} \leq 0} L^{\mathrm{P}}_d(x_d, \lambda_{d+1}) - \big(\underline{V}^{\mathrm{P}}_{d+1}\big)^{\star}(\lambda_{d+1})$$

Proposition (see [Carpentier, Chancelier, De Lara and Rigaut, 2022])

The value functions $\underline{V}^{\mathrm{P}}_d$ are lower bounds to the value functions V^e_d of Problem $\mathcal{P}^e$

$$\underline{V}^{\mathrm{P}}_d \leq V^e_d \ , \ \ \forall d \in \{0, \ldots, D+1\}$$

[1] where $\phi^{\star}(\lambda) = \sup_x \langle \lambda, x\rangle - \phi(x)$ is the Fenchel transform of ϕ

Efficiency of deterministic price decomposition

$$\overbrace{\underline{V}_d^{\mathrm{P}}(x_d) = \max_{\lambda_{d+1} \leq 0} \underbrace{L_d^{\mathrm{P}}(x_d, \lambda_{d+1})}_{\text{Hard to compute}} - \left(\underline{V}_{d+1}^{\mathrm{P}}\right)^{\star}(\lambda_{d+1})}^{\text{Easy to compute by dynamic programming}}$$

It is challenging to compute the intraday function value $L_d^{\mathrm{P}}(x_d, \lambda_{d+1})$ for each couple (x_d, λ_{d+1}) and each day d, but

- we can exploit periodicity of the problem, e.g $L_d^{\mathrm{P}} = L_0^{\mathrm{P}}$
- we can parallelize the computation of L_d^{P} on several days
- we can use any suitable method to solve the multistage intraday problems L_d^{P} (SDP, SDDP, scenario tree methods, PH,...)

Lecture outline

1. Two time scales battery management problem
2. Resource and price decomposition methods
   - Time blocks and resource decomposition
   - Time blocks and price decomposition
   - Producing fast time scale policies
3. Numerical results
   - Managing battery charge and health over 5 days
   - Managing battery charge, health and renewal over 20 years

Value functions $\underline{V}_d^{\mathrm{P}}$ and $\overline{V}_d^{\mathrm{R}}$ yield admissible policies

We have obtained functions that are bounds for the "true" Bellman value functions V_d^e

$$\underline{V}_d^{\mathrm{P}} \leq V_d^e \leq \overline{V}_d^{\mathrm{R}}$$

Now we can solve the following subproblems on all days d

$$\min_{\mathbf{U}_d} \ \mathbb{E}\Big[L_d(x, \mathbf{U}_d, \mathbf{W}_d) + \widetilde{V}_{d+1}\big(f_d(x, \mathbf{U}_d, \mathbf{W}_d)\big)\Big]$$

$$\text{s.t} \quad \sigma(\mathbf{U}_{d,m}) \subset \sigma(\mathbf{W}_{d,0:m})$$

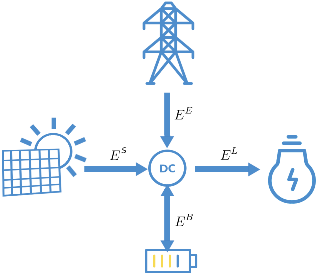
with $\widetilde{V}_{d+1} = \underline{V}_{d+1}^{\mathrm{P}}$ or $\widetilde{V}_{d+1} = \overline{V}_{d+1}^{\mathrm{R}}$, and obtain a resource and a price policies at the fast time scale

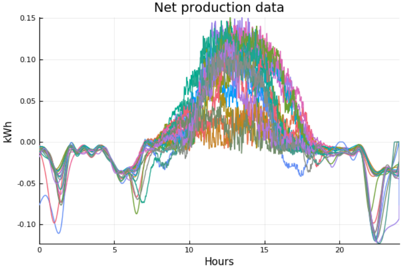
Lecture outline

We present numerical results associated to two use cases

Common data: load/production from a house with solar panels

1. Managing battery charge and health on 5 days
   to compare our algorithms to references on a "small" instance
2. Managing battery charge, health and renewal on 20 years
   to show that targets decomposition scales





Lecture outline

1. Two time scales battery management problem
2. Resource and price decomposition methods
   - Time blocks and resource decomposition
   - Time blocks and price decomposition
   - Producing fast time scale policies
3. Numerical results
   - Managing battery charge and health over 5 days
   - Managing battery charge, health and renewal over 20 years

Application 1: managing charge and aging of a battery

We control a battery

- capacity $c_0 = 13$ kWh
- $h_0 = 100$ kWh of exchangeable energy (4 cycles remaining)
- over $D = 5$ days, so that $D \times M = 7200$ minutes
- with 1 day periodicity

We compare 4 algorithms

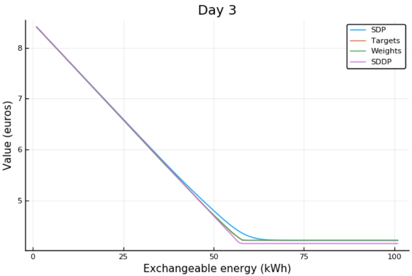
1. stochastic dynamic programming (that is, SDP alone)
2. stochastic dual dynamic programming (that is, SDDP alone)
3. resource decomposition (+ SDDP for intraday problems)
4. price decomposition (+ SDP for intraday problems)

Decomposition provide tighter bounds than S(D)DP

We know that

- $V_d^{\mathrm{SDDP}} \leq V_d \leq V_d^{\mathrm{SDP}}$
- $\underline{V}_d^{\mathrm{P}} \leq V_d \leq \overline{V}_d^{\mathrm{R}}$

We observe that $V_d^{\mathrm{SDDP}} \leq \underline{V}_d^{\mathrm{P}} \leq \overline{V}_d^{\mathrm{R}} \leq V_d^{\mathrm{SDP}}$



We beat SDP and SDDP (that cannot fully handle 7200 stages)

Computation times and convergence

	SDP	Price	SDDP	Resource
Total time	22.5 min	5.0 min	3.6 min	0.41 min
Gap	0.91 %	0.32 %	0.90 %	0.28 %

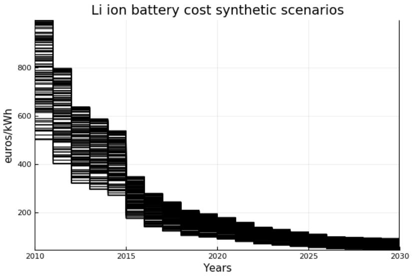
Gap: is between Monte Carlo simulation (upper bound) minus value function at time 0

- Decomposition algorithms display smaller gaps
- Resource decomposition is faster than SDDP
- Price decomposition is faster than SDP

Lecture outline

Application 2: Managing battery charge, health an renewal

- 20 years, 7300 days, 10, 512, 000 minutes, 1 day periodicity
- Battery capacity between 0 and 20 kWh
- Scenarios for batteries prices



SDP and SDDP fail to solve such a problem over millions of stages!

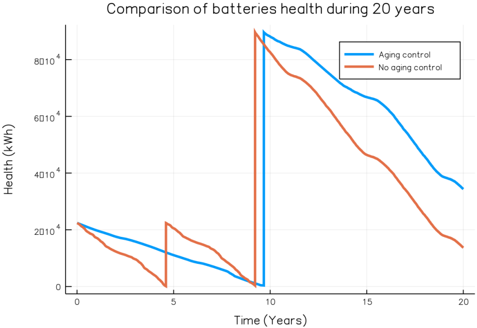
Resource decomposition can handle millions of stages

$$\overbrace{\overline{V}^{\mathrm{R}}_d(x_d) = \min_{r_{d+1}} \underbrace{L^{\mathrm{R}}_d(x_d, r_{d+1})}_{\text{Computing } L^{\mathrm{R}}_d(\cdot,\cdot) \text{ with SDDP takes 60 min}} + \overline{V}^{\mathrm{R}}_{d+1}(r_{d+1})}^{\text{Computing daily value functions by dynamic programming takes 45 min}}$$

- Complexity: 45 min + $D \times$ 60 min
- With periodicity: 45 min + $N \times$ 60 min, with $N \ll D$
- With parallelization: 45 min + 60 min

Does it pay to control aging?

We draw one battery price scenario and one solar/demand scenario over 20 years and simulate the policy obtained by resource decomposition



We make a simulation of 10, 512, 000 decisions in 45 minutes

We compare to a policy that does not control aging

- Without aging control: 3 battery purchases
- With aging control: 2 battery purchases

It pays to control aging with targets decomposition!

Conclusions

1. We have solved problems with millions of time steps using the resource decomposition algorithm
2. We have designed control strategies for charging/aging/renewing batteries
3. We have used our algorithm to improve results obtained with algorithms that are sensitive to the number of time steps (SDP, SDDP)

D. P. Bertsekas.
Dynamic Programming and Optimal Control, Vol. I.
Athena Scientific, Belmont, Massachusets, second edition, 2005.

P. Carpentier and G. Cohen.
Décomposition-coordination en optimisation déterministe et stochastique.
Springer-Verlag, Berlin, 2017.

P. Carpentier, J.-P. Chancelier, M. De Lara, and T. Rigaut.
Time blocks decomposition of multistage stochastic optimization problem.
arXiv:1804.01711, 2018.

P. Carpentier, J.-P. Chancelier, M. De Lara, and T. Rigaut.
Decomposition methods for dynamically monotone two time scales stochastic optimization problem.
Working paper, 2022.

B. Heymann and P. Martinon.
Optimal battery aging: an adaptive weights dynamic programming algorithm.
Journal of Optimization Theory and Applications, 179(3):1043–1053, 2018.

T. Rigaut.
Time decomposition methods for optimal management of energy storage under stochasticity.
Thèse de Doctorat, Université Paris-Est, 2019.