

Optimization Project in Energy

PGE 306

Laurent Pfeiffer

Inria and CentraleSupélec, Université Paris-Saclay

Ensta-Paris
Institut Polytechnique de Paris
January, 2022



3 Dynamic programming

Control strategies

Random demand and decision process.

Two additional difficulties:

- The demand $d(t)$ is **random**.
- No available **mathematical model** for $d(t)$.

Adaptativity of the decision process.

- At the beginning of the time interval 1, $d(1)$ is revealed.
- Then: decision of the variables $a(1)$ and $v(1)$.
- At the beginning of the time interval 2, $d(2)$ is revealed.
- Then: decision of the variables $a(2)$ et $v(2)$.
- Etc.

Control strategies

Therefore, we can allow the following dependences:

- $a(1)$ and $v(1)$ as a function of $d(1)$
- $a(2)$ and $v(2)$ as a function of $d(1)$ and $d(2)$
- $a(3)$ and $v(3)$ as a function $d(1)$, $d(2)$, and $d(3)$
- Etc.

The number of possibilities increase exponentially with the number of time steps!

Control strategies

Demand scenarios.

We call **demand scenario** a vector $(D(s))_{s=1,\dots,T}$.

Two set of scenarios are available:

- **Training set** D_T : history of N_T demand scenarios.
Used to **build** a probabilistic model for the demand and an appropriate *control strategy*.
- **Simulation set** D_S : history of N_S demand scenarios.
Used to **test** the control strategies. Avoid to build biased strategies.

Control strategies

Shifting of the time index.

The two available histories of demand scenarios contain T_0 values of the demand from the “previous day”, corresponding to the time intervals $0, -1, -2, \dots, -(T_0 - 1)$.

On the computer: a demand scenario is a vector of size $T + T_0$. The training and simulation sets are matrices with $(T + T_0)$ columns and respectively N_T and N_S rows.

We “get access” to the demand at time t , for the scenario ℓ with

$$D_T(\ell, t + T_0) \quad D_S(\ell, t + T_0).$$

Control strategies

Online and offline phases.

We compute the decision variables in two steps.

1. **Offline phase.** We compute a variable \mathcal{I} which synthesizes all the available information, depending only on D_T and the global parameters (x_{\max}, P_a, P_v) . For example, \mathcal{I} can contain statistical data for D_T and coefficients describing some value function.

Control strategies

2. **Online phase.** Given a demand scenario $D \in \mathbb{R}^{T+T_0}$, the buying and selling decisions are taken at any time $s = 1, \dots, T$ with the help of some function ϕ in the following way:

$$(a(s), v(s)) = \phi\left(s, x(1), \dots, x(s), D(1), \dots, D(T_0+s), \mathcal{I}\right). (*)$$

Here the variables $x(1), \dots, x(s)$ denote the state-of-charge of the battery at times $1, \dots, s$.

We call **control strategy** the pair (\mathcal{I}, ϕ) .

Control strategies

Remarks.

- The mechanism is **non-anticipative**. At time s , we only use the revealed values of the demand (those until time s) and our a priori knowledge of the demand process, represented by the \mathcal{I} .
- **Feasibility**. The function ϕ must be such that

$$x(s+1) = x(s) + a(s) - v(s) - D(T_0 + s) \in [0, x_{\max}],$$

for any possible demand scenario.

Control strategies

Cost and evaluation of a control strategy.

Let us fix \mathcal{I} and ϕ . Given a demand scenario $D \in \mathbb{R}^{T+T_0}$, we denote

$$J_{\mathcal{I},\phi}(D) = \sum_{s=1}^T \left(P_a(s)a(s) - P_v(s)v(s) \right),$$

where $(a(s))_{s=1,\dots,T}$ and $(v(s))_{s=1,\dots,T}$ are computed with the help of $(*)$.

We set

$$J_{\mathcal{I},\phi} = \frac{1}{N_S} \sum_{\ell=1}^{N_S} J_{\mathcal{I},\phi}(D_S(\ell, \cdot)).$$

This number measure the efficiency of the strategy. Remember that the history D_S is used only for evaluating the control strategy.

Control strategies

We program a control strategy in three steps:

- **Offline phase:** we program \mathcal{I} .
We use D_T .
- **Online phase:** we program ϕ and $J_{\mathcal{I},\phi}$.
We use \mathcal{I} .
- **PEvaluation phase:** we compute $J_{\mathcal{I},\phi}$.
We use $J_{\mathcal{I},\phi(D)}$ and D_S .

Control strategies

A lower bound for the cost

Given a demand scenario $D \in \mathbb{R}^{T+T_0}$, we denote $J_{\text{anti}}(D)$ the optimal cost obtained, assuming that D is entirely known. We denote

$$J_{\text{anti}} = \frac{1}{N_S} \sum_{\ell=1}^{N_S} J_{\text{anti}}(D_S(\ell, \cdot)).$$

The number J_{anti} is a lower bound for the evaluation cost of any (feasible and non-anticipative) strategy.

Exercise 6

Write a function `lower_bound` which computes J_{anti} . To this purpose, use the functions already written in exercise 1. Pay attention to the shifting of time indices.

Control strategies

1. The naive strategy.

- Offline phase: $\mathcal{I} = \emptyset$. We do not exploit D_T .
- Online phase: at time s , given the demand $d(s)$, we chose

$$(a(s), v(s)) = \begin{cases} (d(s), 0), & \text{si } d(s) \geq 0, \\ (0, -d(s)), & \text{si } d(s) \leq 0. \end{cases}$$

Exercise 7

Verify that the naive strategy is non-anticipative and feasible. Write a function `naive_online` which computes the decision variables and the cost associated with a demand scenario (given in input). Write a function `naive_eval` which computes the cost of the cost of the strategy.

Control strategies

2. The reasonable strategy

- Offline phase: $\mathcal{I} = \emptyset$. Again, we do not exploit D_T .
- Online phase: at time s , given the demand $d(s)$ and the state of charge $x(s)$:
 - if $d(s) \geq 0$: we dip into the reserve $x(s)$ and we buy electricity if $d(s) \geq x(s)$.
 - If $d(s) \leq 0$: we stock energy in the battery as much as possible; if $d(s) \leq x(s) - x_{\max}$, the surplus is sold.

Exercise 8

Verify that the strategy is non-anticipative and feasible. Write two function `raisonnable_online` and `raisonnable_eval` implementing and testing this strategy.

1 Control strategies

2 Autoregressive processes

3 Dynamic programming

Autoregressive processes

Generalities.

- We look for a stochastic model describing **faithfully** the evolution of the demand with respect to time.
- This model should be of **reasonable complexity**, so that it can be exploited numerically.
- We are interested in **autoregressive processes**, for which an approach by dynamic programming can be implemented.

Autoregressive processes

Processes of order 0.

We suppose that the demands $d(1), d(2), \dots, d(T)$, are T **independent** random variables. Thus we do not need to identify any correlation between them, but we need to identify the probability distribution of each random variable.

Given t , we approximate $d(t)$ with a random variable which can take N_E different values with probability $p := 1/N_E$. These values are obtained by **sampling**.

Autoregressive processes

Sampling.

Let $h \in \mathbb{R}^{N_T}$ be a given vector, that we need to sample with n_E values. The result of the procedure is a vector $z \in \mathbb{R}^{N_E}$.

- To simplify, we will assume that $q := N_T/N_E$ is an integer.
- Let \tilde{h} be the vector obtained by sorting the values of h , from the smallest value to the largest one.
- We define z as follows:

$$z(1) = \frac{1}{q} \sum_{\ell=1}^q \tilde{h}(\ell), \quad z(2) = \frac{1}{q} \sum_{\ell=q+1}^{2q} \tilde{h}(\ell), \quad \dots$$

$$z(N_E) = \frac{1}{q} \sum_{\ell=N_T-q+1}^{N_T} \tilde{h}(\ell).$$

Autoregressive processes

Exercise 9

- Write a function `sample` realising the sampling of an arbitrary vector h in N_E values. Use the function `sort` of Matlab.
- Write a function `sample_training_set` with output a matrix $E \in \mathbb{R}^{N_E \times T}$ such that each column contains the sampled values of the vectors

$$D_T(:, T_0 + 1), \quad D_T(:, T_0 + 2), \dots \quad D_T(:, T_0 + T).$$

Autoregressive processes

Définition

We call white noise a sequence of independent random variables $(\varepsilon(t))_t$ with null expectation.

Définition

We call the process $d(t)$ an autoregressive process of order $I \in \mathbb{N}$ if there exist deterministic coefficients $\gamma(t), \beta_1(t), \dots, \beta_I(t)$ and a white noise $(\varepsilon(t))_t$ such that:

$$d(t) = \gamma(t) + \beta_1(t)d(t-1) + \dots + \beta_I(t)d(t-I) + \varepsilon(t).$$

Autoregressive processes

Numerical approximation.

We propose the following method to approximate an autoregressive process $d(t)$ of order I . We proceed in two steps:

- For all $t = 1, \dots, T$, compute the solution $(\bar{\gamma}, \bar{\beta}_1, \dots, \bar{\beta}_I)$ to

$$\inf_{\gamma, \beta_1, \dots, \beta_I \in \mathbb{R}} \sum_{\ell=1}^{N_T} \left(D_T(t + T_0) - \left(\gamma + \sum_{i=1}^I \beta_i D_T(t + T_0 - i) \right) \right)^2$$

We set $\gamma(t) = \bar{\gamma}$, $\beta_1(t) = \bar{\beta}_1(t), \dots, \beta_I(t) = \bar{\beta}_I(t)$.

- We sample the variable $\varepsilon(t)$, given by

$$\varepsilon(t) = D_T(t + T_0) - \left(\gamma + \sum_{i=1}^I \beta_i D_T(t + T_0 - i) \right).$$

Autoregressive processes

Exercise 10

Write a function `auto_reg_1` realizing the approximation of $d(t)$ as an autoregressive process of order 1

Output variables: $\gamma \in \mathbb{R}^T$, $\beta_1 \in \mathbb{R}^T$, $E \in \mathbb{R}^{N_E \times T}$.

Optional. Write a function `auto_reg` which realizes the approximation of $d(t)$ by an autoregressive process of arbitrary order (given as input variable).

Autoregressive processes

Predictive model.

Phase offline. Approximation of $d(t)$ with an autoregressive process of order 1, with the help of coefficients γ and β_1 .

Phase online. Let t be the current time step. Let x_t denote the current state-of-charge of the battery and let d_t denote the demand at time t .

1. Prediction. Compute $(D_p(s))_{s=t,\dots,T}$ as follows:

$$\begin{aligned}D_p(t) &= d_t, \\D_p(t+1) &= \gamma(t+1) + \beta_1(t+1)D_p(t), \\D_p(t+2) &= \gamma(t+2) + \beta(t+2)D_p(t+1), \\&\dots \\D_p(T) &= \gamma(T) + \beta(T)D_p(T).\end{aligned}$$

Predictive method

2. Optimization. We solve:

$$\inf_{\substack{x(t), \dots, x(T+1) \\ a(t), \dots, a(T) \\ v(t), \dots, v(T)}} \sum_{s=t}^T P_a(s)a(s) - P_v(s)v(s)$$
$$\text{s.t.} \quad \begin{cases} x(s+1) = x(s) + a(s) - v(s) - D_p(s), & s = t, \dots, T \\ x(t) = x_t \\ a(s) \geq 0, & s = t, \dots, T \\ v(s) \geq 0, & s = t, \dots, T \\ 0 \leq x(s) \leq x_{\max}, & s = t, \dots, T \end{cases}$$

Let $\bar{x}(t), \dots, \bar{x}(T+1)$, $\bar{a}(t), \dots, \bar{a}(T)$, $\bar{v}(t), \dots, \bar{v}(T)$ be a solution. We take:

$$a(t) = \bar{a}(t), \quad v(t) = \bar{v}(t).$$

Predictive method

Exercise 11

Implement the predictive method described above.

1 Control strategies

2 Autoregressive processes

3 Dynamic programming

Dynamic programming

Case of an autoregressive process of order 0.

We suppose that the demande $d(t)$ is described by an autoregressive process of order 0, that is, all the random variables $d(1), \dots, d(T)$ are independent.

We suppose that a matrix $(D(j, t))_{\substack{j=1, \dots, N_E \\ t=1, \dots, T}}$ is given and that

$$\mathbb{P}[d(t) = D(j, t)] = \frac{1}{N_E},$$

for all $j = 1, \dots, N_E$ and for all $t = 1, \dots, T$.

Dynamic programming

From now on, we need to work with two value functions:

- $V(t, x)$: the expectation of the optimal cost (from t to T), with initial state-of-charge x at time t , before the demand $d(t)$ is revealed.
- $\tilde{V}(t, x, d_t)$: the expectation of the optimal cost (from t to T), with initial state-of-charge x at time t , conditionally to $d(t) = d$.

Dynamic programming

Theorem

The following holds true.

- For all $x \in [0, x_{\max}]$, $V(T+1, x) = 0$.
- For all $t = 1, \dots, T$, for all $x \in [0, x_{\max}]$,

$$V(t, x) = \frac{1}{N_E} \sum_{j=1}^{N_E} \tilde{V}(t, x, E(j, t)).$$

- For all $t = 1, \dots, T$, for all $x \in [0, x_{\max}]$,

$$\tilde{V}(t, x, d) = \inf_{(z, a, v) \in \mathbb{R}^3} P_a(t)a - P_v(t)v + V(t+1, z), \quad (DP(t, x, d))$$

$$\text{sous la contrainte : } \begin{cases} z = x + a - v - d, \\ 0 \leq x \leq x_{\max}, \\ a \geq 0, v \geq 0. \end{cases}$$

Dynamic programming

Phase offline: numerical approximation of $V(\cdot, \cdot)$.

The mechanism is similar to the one seen in the deterministic framework.

Let $t \in \{1, \dots, T\}$. Let us suppose $V(t+1, \cdot)$ that is known and represented as a polynomial function.

- We calculate $\tilde{V}(t, x_j, D(k, t))$ for all $j = 1, \dots, J$ and for all $k = 1, \dots, N_E$, by solving $(DP(t, x_j, D(k, t)))$.
- We calculate $V(t, x_j)$ for all $j = 1, \dots, J$.
- We approximate the full function $V(t, \cdot)$ by approximation.

Phase online : at time t , when the demand $d(t)$ has been revealed, we solve $(DP(t, x, d))$, with x the current state-of-charge at time t and $d = d(t)$.

Dynamic programming

Exercise 12

Implement the control strategy induced by the dynamic programming principle with the auto-regressive model of order zero.

Dynamic programming

Case of a first-order autoregressive process.

We suppose that the demand $d(t)$ is described by a first-order autoregressive process, that is:

$$d(t) = \gamma(t) + \beta_1(t)d(t-1) + \varepsilon(t),$$

where $(\varepsilon(t))_{t=1,\dots,T}$ is a white noise.

We suppose that a matrix $(E(k, t))_{\substack{k=1,\dots,N_E \\ t=1,\dots,T}}$ is given and

$$\mathbb{P}[\varepsilon(t) = E(k, t)] = \frac{1}{N_E},$$

for all $k = 1, \dots, N_E$, and for all $t = 1, \dots, T$.

Dynamic programming

We consider two value functions:

- $V(t, x, d_{t-1})$: the optimal expected cost (from t to T), with state-of-charge x at time t , knowing that $d(t-1) = d_{t-1}$, before that $d(t)$ is revealed.
- $\tilde{V}(t, x, d_t)$: the optimal expected cost, with state-of-charge x at time t , knowing that $d(t) = d_t$.

Dynamic programming

Theorem

The following holds true.

- For all $x \in [0, x_{\max}]$, $V(T+1, x, d_T) = 0$.
- For all $t = 1, \dots, T$, for all $x \in [0, x_{\max}]$,

$$V(t, x, d_{t-1}) = \frac{1}{N_E} \sum_{k=1}^{N_E} \tilde{V}(t, x, \gamma(t) + \beta_1(t)d_{t-1} + E(k, t)).$$

- For all $t = 1, \dots, T$, for all $x \in [0, x_{\max}]$,

$$\tilde{V}(t, x, d_t) = \inf_{(z, a, v) \in \mathbb{R}^3} P_a(t)a - P_v(t)v + V(t+1, z, d_t),$$

$$\text{subject to: } \begin{cases} z = x + a - v - d_t, \\ 0 \leq x \leq x_{\max}, \\ a \geq 0, v \geq 0. \end{cases} \quad (DP(t, x, d_t))$$

Dynamic programming

Remark. The value function (at time t) depends on two variables. We can seek for an approximation with a second-order polynomial of the form:

$$\begin{aligned} V(t, x, d_{t-1}) = & \alpha_1(t) + \alpha_2(t)x + \alpha_3(t)d_{t-1} \\ & + \alpha_4(t)x^2 + \alpha_5(t)xd_{t-1} + \alpha_6(t)d_{t-1}^2. \end{aligned}$$