Optimal control and large-scale optimization
Energy applications

Laurent Pfeiffer
Inria-Saclay, Laboratory of Signals and Systems,
Federation of Mathematics of CentraleSupélec
Paris-Saclay University

ACSYON course - University of Limoges - November 8, 2024

Introduction

Curriculum:

- PhD in Applied Mathematics (2013)
- Assistant professor in Graz (Austria), from 2013 to 2019
- Since 2019: researcher at Inria, working on optimal control (the optimization of dynamical systems).

About Inria:

- a national research institute in computer science and applied mathematics (8 regional centers).
- Organization in research teams dedicated to specialized topics, in partnership with faculty members of the local universities
- Contracts with companies, other research institutes.
- Many opportunities for master internships and PhDs (JobIn).

A motivating situation

Consider a **hydro-electric valley** consisting of:

- **dams**, storing water
- **turbines**, producing electricity, in function of the amount of water flowing out of every dam
- connections between the dams.

General objective: optimizing the revenues resulting from electricity production.

Two major difficulties:

- A **dynamic** problem (i.e., decisions should be taken over time). $\rightarrow$ Should consume the water now? Or store for later use?
- **Randomness** in the problem. $\rightarrow$ Unknown precipitations affect the evolution of the level of water in the dams.

Mathematical model

Consider:

- A time interval: $\{0, 1, 2, \ldots, T\}$
- A set $\mathcal{X}$: the state set
- A set $\mathcal{U}$: the control set
- A set Ω: the set of random events.
- Random variables $\xi_0, \ldots, \xi_{T-1}$, valued in Ω (the uncertainties)
  $\rightarrow$ the precipitations at each time t.

Optimization variables of the problem:

- The states $X_0, \ldots, X_T$, valued in $\mathcal{X}$ $\rightarrow$ the level of water in each dam, at every time t
- The controls $U_0, \ldots, U_{T-1}$, valued in $\mathcal{U}$ $\rightarrow$ the quantity of water that is turbined in each dam, the amount of water conveyed on the network, at each time t.

Mathematical Model

Stochastic model: the random variables are supposed to be independent and identically distributed. The set Ω is finite. We denote

$$p_\omega = \mathbb{P}\big[\xi_t = \omega\big], \quad \forall \omega \in \Omega.$$

Dynamical model: there is a function $f \colon \mathcal{X} \times \mathcal{U} \times \Omega \to X$ such that

$$X_{t+1} = f(X_t, U_t, \xi_t), \quad \forall t = 0, \ldots, T-1$$

$\rightarrow$ f describes the evolution of the level of water according to conservation principles.

Mathematical model

Cost function: we are given two functions

$$\ell\colon \mathcal{X} \times \mathcal{U} \times \Omega \to \mathbb{R} \quad \text{and} \quad \phi\colon \mathcal{X} \to \mathbb{R}.$$

We aim at minimizing

$$\sum_{t=0}^{T-1} \ell(X_t, U_t, \xi_t) + \phi(X_T).$$

Mathematical model

Decision process:

- The initial state $X_0 = x_0$ is known.
- The noise ξ_0 is discovered.
- The control U_0 is chosen.
- The sytem moves to $X_1 = f(X_0, U_0, \xi_0)$.
- The noise ξ_1 is discovered.
- The control U_1 is chosen.
- ...
- At time $t-1$, the noise ξ_{T-1} is discovered, U_{T-1} is chosen, the system moves to its final position X_T.

Measurability condition: at time t, the control U_t can depend on the revealed noises $\xi_0, \ldots, \xi_t$, but is independent of $\xi_{t+1}, \ldots, \xi_{T-1}$.

Mathematical model

We obtain the following **stochastic optimal control** problem:

$$\inf_{\substack{X_0,\dots,X_T \\ U_0,\dots,U_{T-1}}} \mathbb{E}\left[\sum_{t=0}^{T-1} \ell(X_t, U_t, \xi_t) + \phi(X_T)\right],$$

$$\text{subject to: } \begin{cases} X_{t+1} = f(X_t, U_t, \xi_t), \ \forall t = 0, \dots, T-1 \\ X_0 = x_0 \\ \text{measurability condition.} \end{cases}$$

A difficult problem! For the control U_t, there are $|\Omega|^{t+1}$ scenarios to be taken into account.

Dynamic programming

Dynamic programming is an ubiquitous principle in optimization theory.

It consists in inserting a given problem into a family of problem of increasing complexity. Then one solves them all, starting from the easiest one.

In our context, we obtain a family of problems by letting vary the initial time and the initial state.

Dynamic programming

Parameters: $s \in \{0, \dots, T\}$ and $x \in \mathcal{X}$.
Parameterized problem:

$$V_s(x) = \inf_{\substack{X_s,\dots,X_T \\ U_s,\dots,U_{T-1}}} \mathbb{E}\left[\sum_{t=s}^{T-1} \ell(X_t, U_t, \xi_t) + \phi(X_T)\right], \qquad (P_s(x))$$

$$\text{subject to:} \quad \begin{cases} X_{t+1} = f(X_t, U_t, \xi_t), \ \forall t = s, \dots, T-1 \\ X_s = x \\ \text{measurability condition.} \end{cases}$$

We refer to the function $(s, x) \mapsto V_s(x)$ as the value function.
We are interested in solving $P_0(x_0)$.

Dynamic programming

Theorem

The following relation hold true:

$$V_s(x) = \sum_{\omega\in\Omega} p_\omega \Big(\inf_{u\in U} \ell(x,u,\omega) + V_{s+1}(f(x,u,\omega)) \Big),$$
$$\forall x \in \mathcal{X}, \ \forall s = 0, \dots, T-1$$
$$V_T(x) = \phi(x), \quad \forall x \in \mathcal{X}.$$

This opens up the possibility to compute numerically the value function, in a backward fashion, from $V_T(\cdot)$, $V_{T-1}(\cdot)$, … to $V_0(\cdot)$.

This raises many difficulties since often, $V_t(x)$ cannot be computed for all values of x (curse of dimensionality).

Dynamic programming

How to use the value function?

We come back to our decision process.

At time t, we are at state X_t. The noise ξ_t is discovered. The control U_t should be taken as a solution to:

$$\inf_{u \in U} \ell(X_t, u, \xi_t) + V_{s+1}(f(X_t, u, \xi_t)).$$

Micro-grid management

Lefranc, Carpentier, Chancelier, De Lara. EMSx: a numerical benchmark for energy management systems, *Energy systems*, 2021.

The article proposes an optimal control model for microgrid, a small electricity unit involving:

- consumers
- a source of renewable energy
- an electricity producer
- batteries for storing energy.

Gas management

Pfeiffer, Apparigliato, Auchapt. Two methods of pruning Benders' cuts and their application to the management of a gas portfolio, *Inria research report*, 2012.

The article proposes an optimal control approach for gas management, involving:

- Storages
- Contracts
- A random demand in gas.

Motivating example: smart charging

- Consider a parking for electrical vehicles (in China up 1000!).
- How to charge optimally the parking?
  - the total electrical load must be spread conveniently over time
  - the electricity price may vary over the day
  - each electrical car has its own constraints.
- As the number N of vehicles increases, the problem becomes intractable.
- Aim of our work, in parnership with EDF: the development of a numerical method that scales well as N increases.

Bonnans, Liu, Oudjane, Pfeiffer, Wan. Large-scale nonconvex optimization: randomization, gap estimation, and numerical resolution, *SIAM J. Optim.*, 2023.

Frank-Wolfe algorithm

Consider the following problem:

$$\inf_{x \in \mathbb{R}^n} F(x), \quad \text{subject to: } x \in K. \qquad (\mathcal{P})$$

Assumptions:

- $F : \mathbb{R}^n \to \mathbb{R}$ is convex, continuously differentiable, with Lipschitz-continuous gradient.
- $K \subseteq \mathbb{R}^n$ is convex and compact.

The **linearized problem** at $\tilde{x}$ is defined by

$$\inf_{x \in \mathbb{R}^n} \langle \nabla F(\tilde{x}), x \rangle, \quad \text{subject to: } x \in K. \qquad (\mathcal{P}_{\text{lin}}(\tilde{x}))$$

We assume that it is easy to solve numerically, for any $\tilde{x}$.

Frank-Wolfe algorithm

Algorithm 1: Frank-Wolfe algorithm

Input: $\bar{x}_0 \in K$;
for $k = 0, 1, \ldots$ **do**
  Find a solution x_k to $\mathcal{P}_{\text{lin}}(\bar{x}_k)$;
  Set $\omega_k = 2/(k+2)$;
  Set $\bar{x}_{k+1} = (1-\omega_k)\bar{x}_k + \omega_k x_k$;
end

Lemma

There exists a constant C such that

$$F(\bar{x}_k) \leq F(\bar{x}) + \frac{C}{k}, \quad \forall k > 0,$$

where $\bar{x}$ denotes a solution of $(\mathcal{P})$.

Frank-Wolfe algorithm

Assume the case where K is the Cartesian product of m sets:

$$K = K_1 \times K_2 \times \ldots K_m.$$

The optimization variable x takes the form of a tuple $(x_1, \ldots, x_m)$.

The linearized problem at $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_m)$ writes:

$$\inf_{(x_1,\ldots,x_m)} \sum_{k=1}^{m} \langle \nabla_{x_k} F(\tilde{x}), x_k \rangle, \quad \text{subject to: } x_k \in K_k.$$

This problem is equivalent to solve K independent problems:

$$\inf_{x_k} \langle \nabla_{x_k} F(\tilde{x}), x_k \rangle, \quad \text{subject to: } x_k \in K_k, \quad \forall k = 1, \ldots, K.$$

$\rightarrow$ They can be solved in parallel! Specific methods can be utilized for each of them.

Setting

Consider the N-agent problem

$$\inf_{x \in \mathcal{X}} J(x) = f\Big(\underbrace{\frac{1}{N} \sum_{i=1}^{N} g_i(x_i)}_{\text{aggregate}} \Big) + \frac{1}{N} \sum_{i=1}^{N} h_i(x_i), \qquad (\mathcal{P})$$

where $x = (x_1, \dots, x_N) \in \mathcal{X} = \prod_{i=1}^{N} \mathcal{X}_i$.

Data:

- the feasible sets $\mathcal{X}_i$
- the individual costs $h_i \colon \mathcal{X}_i \to \mathbb{R}$
- the aggregate space $\mathcal{E}$, a Hilbert space
- the contribution functions $g_i \colon \mathcal{X}_i \to \mathcal{E}$
- the social cost $f \colon \mathcal{E} \to \mathbb{R}$.

Application

Applications in energy management problems:

- Set of agents: a (large) set of **small flexible consumptions units** (e.g. batteries, heating devices).
  Flexible: consumption can be shifted over time.
- Aggregate: the **total consumption**, at each time step of a given time interval.
- Social cost: **penalty function** for the difference between total consumption and a reference production level.

Wang. Vanishing Price of Decentralization in Large Coordinative Nonconvex Optimization, *SIAM J. Optimization*, 2017.

Séguret et al. Decomposition of convex high dimensional aggregative stochastic control problems, *Appl. Math Optim.*, 2023.

Assumptions

Assumptions:

- f is convex
- ∇f is D-Lipschitz continuous
- for all $i = 1, \ldots, N$, $\mathrm{diam}(g_i(\mathcal{X}_i)) \leq D$.

All constants appearing later on depend on D but not on N.
Another "numerical" assumption will be made later.

General difficulties:

- No convexity property of J.
- No regularity property for $\mathcal{X}_i$, g_i, h_i. In general, J is not differentiable.
- Large-scale (when N is large)... but N large actually helps!

Relaxation

General idea:

- Variable x_i replaced by a **probability distribution** $\mu_i \in \mathcal{P}(\mathcal{X}_i)$.
- The terms $g_i(x_i)$ and $h_i(x_i)$ are respectively replaced by

$$\mathbb{E}_{\mu_i}[g_i] := \int_{\mathcal{X}_i} g_i(x_i)\,\mathrm{d}\mu_i(x_i), \quad \mathbb{E}_{\mu_i}[h_i] := \int_{\mathcal{X}_i} h_i(x_i)\,\mathrm{d}\mu_i(x_i).$$

The relaxed problem:

$$\inf_{\mu} \; \tilde{J}(\mu) := f\Big(\frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\mu_i}[g_i]\Big) + \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\mu_i}[h_i], \qquad (\tilde{\mathcal{P}})$$

where $\mu = (\mu_1, ..., \mu_N) \in \prod_{i=1}^{N} \mathcal{P}(\mathcal{X}_i)$.

Remark: The cost function $\tilde{J}$ is **convex**.

Mean field relaxation

Remark: In the **homonegeous** case where $\mathcal{X} = \mathcal{X}_i$, $g = g_i$, $h = h_i$, for all $i = 1, ..., N$, the original problem is equivalent to

$$\inf_{\mu \in \mathcal{P}_N(\mathcal{X})} f\left(\mathbb{E}_\mu[g]\right) + \mathbb{E}_\mu[h],$$

where $\mathcal{P}_N(\mathcal{X}) = \left\{ \frac{1}{N} \sum_{i=1}^N \delta_{x_i} \,\middle|\, x_i \in \mathcal{X}, \ \forall i = 1, \dots, N \right\}$.

The relaxed problem is equivalent to:

$$\inf_{\mu \in \mathcal{P}(\mathcal{X})} f\left(\mathbb{E}_\mu[g]\right) + \mathbb{E}_\mu[h],$$

in which μ models the **distribution of the decisions** of a continuum of agents.

Gap estimation

Theorem

There exists $C > 0$ (depending on D only) such that

$$\mathrm{Val}(\tilde{\mathcal{P}}) \leq \mathrm{Val}(\mathcal{P}) \leq \mathrm{Val}(\tilde{\mathcal{P}}) + \frac{C}{N}.$$

Proof. **Lower bound** of $\mathrm{Val}(\mathcal{P})$.
Let $x \in \mathcal{X}$. Let $\mu = (\delta_{x_1}, ..., \delta_{x_N})$. Then,

$$\mathrm{Val}(\tilde{\mathcal{P}}) \leq \tilde{J}(\mu) = J(x).$$

Minimizing with respect to x yields the result.

Gap estimation

Upper bound of $\mathrm{Val}(\mathcal{P})$. Let $\varepsilon > 0$. Let $\mu \in \prod_{i=1}^{N} \mathcal{P}(\mathcal{X}_i)$ be ε-optimal for the relaxed problem.
Let $X_1,\ldots,X_N$ be N independent random variables such that

$$\mathrm{Law}(X_i) = \mu_i, \quad i = 1, \ldots, N.$$

Then, setting $Y = \frac{1}{N} \sum_{i=1}^{N} g_i(X_i)$,

$$\begin{aligned} \tilde{J}(\mu) &= f\Big(\frac{1}{N} \sum_{i=1}^{N} \mathbb{E}[g_i(X_i)] \Big) + \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}[h_i(X_i)], \\ &= f(\mathbb{E}[Y]) + \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}[h_i(X_i)]. \end{aligned}$$

Therefore, $\mathbb{E}[J(X)] - \tilde{J}(\mu) = \mathbb{E}[f(Y)] - f(\mathbb{E}[Y])$.

Gap estimation

Using the Lipschitz continuity of ∇f, it is easy to show that:

$$\mathbb{E}[f(Y)] - f(\mathbb{E}[Y]) \leq \frac{D}{2}\mathbb{E}\Big[\|Y - \mathbb{E}[Y]\|^2\Big]$$

Since $Y = \frac{1}{N}\sum_{i=1}^{N} g_i(X_i)$ and since the X_i are independent,

$$\mathbb{E}\Big[\|Y - \mathbb{E}[Y]\|^2\Big] = \frac{1}{N^2}\sum_{i=1}^{N}\mathbb{E}\Big[\|g_i(X_i) - \mathbb{E}[g_i(X_i)]\|^2\Big] \leq \frac{D^2}{N}.$$

It finally follows that

$$\begin{aligned} \mathrm{Val}(\mathcal{P}) - \mathrm{Val}(\tilde{\mathcal{P}}) &\leq \mathbb{E}[J(X)] - \tilde{J}(\mu) + \varepsilon \\ &\leq \frac{L}{2}\mathbb{E}\Big[\|Y - \mathbb{E}[Y]\|^2\Big] + \varepsilon \leq \frac{D^2 L}{2N} + \varepsilon. \end{aligned}$$

The subproblem

We call any map $\mathbb{S}\colon \lambda \in \mathcal{E} \mapsto \big(\mathbb{S}_1(\lambda), \ldots, \mathbb{S}_N(\lambda)\big) \in \mathcal{X}$ a **best-response** function if for any $\lambda \in \mathcal{E}$,

$$\mathbb{S}_i(\lambda) \in \underset{x_i \in \mathcal{X}_i}{\operatorname{argmin}} \ \langle \lambda, g_i(x_i) \rangle + h_i(x_i), \quad \text{for } i = 1, \ldots, N.$$

The variable λ can be here interpreted as a **price** for the contribution to the aggregate.

Numerical assumption. We assume that such a function can be easily constructed numerically. The evaluation of $\mathbb{S}$ relies on the resolution of N **independent** optimization problems.

The subproblem

Lemma

Let $\tilde{\mu} \in \prod_{i=1}^N \mathcal{P}(\mathcal{X}_i)$*. Let* $\lambda = \nabla f\left(\frac{1}{N}\sum_{i=1}^N \mathbb{E}_{\tilde{\mu}_i}[g_i]\right)$*. Define*

$$\hat{\mu} = \left(\delta_{\mathbb{S}_1(\lambda)}, \dots, \delta_{\mathbb{S}_N(\lambda)}\right).$$

Then $\hat{\mu}$ *is a solution to*

$$\inf_{\mu \in \prod_{i=1}^N \mathcal{P}(\mathcal{X}_i)} D\tilde{J}(\tilde{\mu}).\mu. \qquad (\tilde{\mathcal{P}}_{\text{lin}}(\tilde{\mu}))$$

Proof. Straightforward calculations yield:

$$D\tilde{J}(\tilde{\mu}).\mu = \frac{1}{N}\sum_{i=1}^N \mathbb{E}_{\mu_i}\Big[\langle \lambda, g_i(\cdot)\rangle + h_i(\cdot)\Big].$$

Frank-Wolfe algorithm

Algorithm 2: Frank-Wolfe algorithm

Input: $\bar{\mu}^0$;
for $k = 0, 1, \dots$ **do**
  Find a solution μ^k to $\tilde{\mathcal{P}}_{\text{lin}}(\bar{\mu}^k)$;
  Set $\omega_k = \frac{2}{k+2}$;
  Set $\bar{\mu}^{k+1} = (1 - \omega_k)\bar{\mu}^k + \omega_k \mu^k$;
end

Difficulties:

- How to deduce an **approximate solution** to $(\mathcal{P})$ from $\bar{\mu}^k$?
- The support of $\bar{\mu}_i^k$ possibly is of cardinality k.

Selection

Selection: A simple **stochastic method** for constructing $x \in \mathcal{X}$ out of $\mu \in \prod_{i=1}^{N} \mathcal{P}(\mathcal{X}_i)$.

Algorithm 3: Selection algorithm

Input: μ, $n \in \mathbb{N}$;

Construct a random variable $X = (X_1, \dots, X_N)$ such that

$$X_1,\dots,X_N \text{ are independent}, \qquad \mathrm{Law}(X_i) = \mu_i.$$

for $j = 1, \dots, n$ **do**

  Draw samples $\hat{x}^j = (x_1^j, \dots, x_N^j)$ of $(X_1, \dots, X_N)$.

end

Output: $\hat{x} \in \underset{x \in \{\hat{x}^1,\dots,\hat{x}^n\}}{\operatorname{argmin}} J(x)$.

Selection

Lemma

Let $\mu \in \prod_{i=1}^{N} \mathcal{P}(\mathcal{X}_i)$ and let $n \in \mathbb{N}$. There exists a constant $C > 0$ such that for any $\varepsilon > 0$,

$$\mathbb{P}\Big[J(\hat{x}) \geq \tilde{J}(\mu) + \frac{C}{N} + \varepsilon\Big] \leq \exp\Big(-\frac{nN\varepsilon^2}{C}\Big).$$

Proof. Let X be as in the selection algorithm. We know that

$$\tilde{J}(\mu) - \mathbb{E}[J(X)] \leq \frac{C}{N}.$$

Concentration inequality: by McDiarmid's inequality, there exists $C > 0$ such that for any $\varepsilon > 0$,

$$\mathbb{P}\Big[J(X) \geq \mathbb{E}[J(X)] + \varepsilon\Big] \leq \exp\Big(-\frac{N\varepsilon^2}{C}\Big).$$

Stochastic Frank-Wolfe (SFW) algorithm

Algorithm 4: Stochastic Frank-Wolfe algorithm

Input: $\bar{\mu}^0$, a sequence $(n_k)_{k\in\mathbb{N}}$;
for $k = 0, 1, \dots$ **do**
  Find a solution μ^k to $\tilde{\mathcal{P}}_{\text{lin}}(\bar{\mu}^k)$;
  Set $\omega_k = \frac{2}{k+2}$;
  Set $\tilde{\mu}^{k+1} = (1-\omega_k)\bar{\mu}^k + \omega_k \mu^k$;
  Set $\bar{x}^{k+1} = \text{Selection}(\tilde{\mu}^{k+1}, n_k)$;
  Set $\bar{\mu}^{k+1} = \left(\delta_{\bar{x}_1^{k+1}}, \dots, \delta_{\bar{x}_N^{k+1}}\right)$.
end

The algorithm can be re-written as an **easy-to-implement** algorithm that does not involve probability distributions.

Stochastic Frank-Wolfe algorithm

Algorithm 5: SFW algorithm: practical version

Input: $\bar{x}^{(0)}$, a sequence $(n_k)_{k\in\mathbb{N}}$;
for $k = 0, 1, \ldots$ **do**
  Set $\lambda^k = \nabla f(\frac{1}{N}\sum_{i=1}^{N} g_i(\bar{x}_i^k))$;
  Compute $x^k = \mathbb{S}(\lambda^k)$;
  Set $\omega_k = 2/(k+2)$;
  for $j = 1, \ldots, n_k$ **do**
    for $i = 1, \ldots, N$ **do**
      Draw $Z_i^{k,j} \sim (1-\omega_k)\delta_0 + \omega_k\delta_1$;
      Set $x_i^{k,j} = (1 - Z_i^{k,j})\bar{x}_i^k + Z_i^{k,j} x_i^k$;
    end
    Set $x^{k,j} = (x_i^{k,j})_{i=1,\ldots,N}$;
  end
  Find $\bar{x}^{(k+1)} \in \underset{x\in\{x^{k,1},\ldots,x^{k,n_k}\}}{\operatorname{argmin}} J(x)$
end

Convergence result

Theorem

There exists a constant $C > 0$ such that for all $K \leq 2N$, for all $\varepsilon > 0$, it holds:

$$\mathbb{P}\Big[J(\bar{x}^K) \geq \mathrm{Val}(\tilde{P}) + \frac{C}{K} + \varepsilon \Big] \leq \exp\Big(- \frac{N\varepsilon^2}{C_1(K) + \varepsilon C_2(K)} \Big),$$

where

$$C_1(K) = C \sum_{k=1}^{K-1} \frac{k(k+1)^2}{n_k K^2 (K+1)^2},$$

$$C_2(K) = C \max_{k \leq K-1} \frac{(k+1)(k+2)}{n_k K(K+1)}.$$

Remark. We can find a C/N-optimal solution with arbitrarily small probability if $n_k \geq Ak^2/N$, with A large enough.

Numerical example

Let $A \in \mathbb{R}^{M \times N}$ and let $\bar{y} \in \mathbb{R}^M$. Consider:

$$\min_{x \in \{0,1\}^N} \frac{1}{N^2} \|Ax - \bar{y}\|^2 = \left\| \frac{1}{N} \sum_{i=1}^{N} \left(A_i x_i - \frac{\bar{y}_i}{N} \right) \right\|^2. \qquad \text{(MIQP)}$$

Data: $M = N = 100$.

Remark: Problem (MIQP) is a discrete problem, over a set of cardinality 2^{100}.

Numerical example

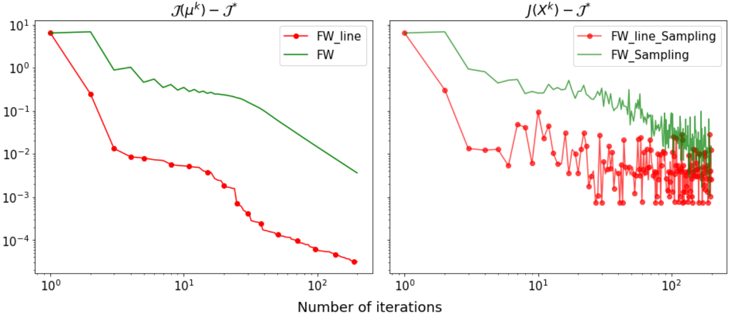

Figure: Convergence of the relaxed optimality gap.

Left: Frank-Wolfe for the relaxed problem.
Right: Selection algorithm applied to the iterates.

Numerical example

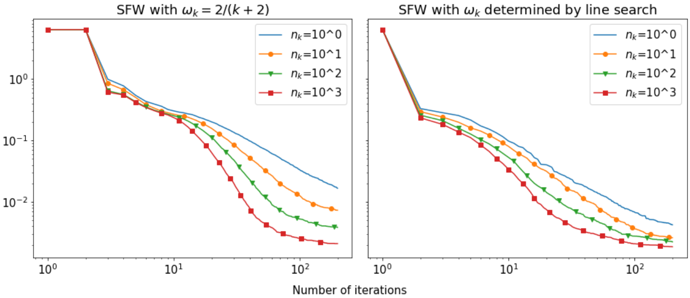

Figure: Relaxed optimality gap for Stochastic Frank-Wolfe algorithm.

Left: Stepsize $\delta_k = 2/(k+2)$.
Right: Stepsize determined by line-search.

Model

A simple model for smart charging, published in:

Liu, Oudjane, Pfeiffer. Decomposed resolution of finite-state aggregative optimal control problems, Conference on Control and applications, 2023.

A model with

- N agents, indexed by $i = 1, \dots, N$
- time steps, ranging over $t = 0, 1, \dots, T$.

For any agent i, we fix

- a finite state set S_i
- a finite control set U_i
- a transition functions $\pi_i^t \colon S_i \times U_i \to S_i$
- the feasible controls of the agent: $U_i^t \colon S_i \to 2^{U_i}$
- the feasible initial states S_i^0, a subset of S_i.

Model

Cost functions:

- social cost: $f_t : \mathbb{R} \to \mathbb{R}$
- contribution: $h_i^t : S_i^t \times U_i \to \mathbb{R}$
- individual cost: $\ell_i^t : S_i^t \times U_i \to \mathbb{R}$.

The optimal control problem of interest reads:

$$\begin{cases} \inf_{(s,u)} & J(s,u) := \sum_{t=0}^{T} f_t\Big(\frac{1}{N}\sum_{i=1}^{N} h_i^t(s_i^t, u_i^t)\Big) \\ & \qquad\qquad + \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T} \ell_i^t(s_i^t, u_i^t), \\ \text{s.t.} & s_i^{t+1} = \pi_i^t(s_i^t, u_i^t),\ u_i^t \in U_i^t(s_i^t),\ s_i^0 \in S_i^0, \\ & \forall t = 0, 1, \ldots, T-1,\ i = 1, 2, \ldots, N, \end{cases}$$

where $(s,u) = (s_i^t, u_i^t)_{i=1,\ldots,N}^{t=0,\ldots,T}$.

Model

Batteries parameters:

- an initial state of charge $s_i^{\text{in}} \in \mathbb{N}$
- a maximal state of charge $s_i^{\max} \in \mathbb{N}$
- a maximal load speed $u_i^{\max} \in \mathbb{N}$.

We define:

$$
\begin{aligned}
S_i &= \{s_i^{\text{in}}, \dots, s_i^{\max}\}, \\
S_i^0 &= \{s_i^{\text{in}}\}, \\
U_i &= \{0, \dots, u_i^{\max}\}, \\
U_i^t(s_i^t) &= \{0, \dots, \min(u_i^{\max}, s_i^{\max} - s_i^t)\}, \\
\pi_i^t(s_i^t, u_i^t) &= s_i^t + u_i^t.
\end{aligned}
$$

In words: the charging of the battery is additive, the charging speed is bounded by $u_i^{\max}$ and is such that s_i^t cannot exceed $s_i^{\max}$.

Model

Some positive coefficients $(\beta_i)_{i=1,\dots,N}$, $(\alpha_t)_{t=0,\dots,T-1}$, and $(c_t)_{t=0,\dots,T-1}$ are given. The individual costs are

$$\ell_i^t(s_i^t, u_i^t) = 0, \quad \forall t = 0, \dots, T-1,$$
$$\ell_i^T(s_i^T, u_i^T) = \beta_i(s_i^{\max} - s_i^T)^2.$$

The contributions are defined by $h_i^T(s_i^T, u_i^T) = 0$ and

$$h_i^t(s_i^t, u_i^t) = u_i^t, \quad \forall t = 0, \dots, T-1.$$

The social costs f_t are defined by $f_T(y_T) = 0$ and

$$f^t(y_t) = \alpha^t(y_t - c_t)^2, \quad \forall t = 0, \dots, T-1.$$

Therefore, the cost function J reads

$$\sum_{t=0}^{T-1} \alpha^t \left(\left(\frac{1}{N} \sum_{i=1}^{N} u_i^t \right) - c^t \right)^2 + \frac{1}{N} \sum_{i=1}^{N} \beta_i \left(s_i^T - s_i^{\max} \right)^2 .$$

Model

The parameters are chosen as follows:

- $N = 100$, $T = 24$
- s_i^{in} (resp. $s_i^{\max}$) is chosen randomly and uniformly in $\{0, 1, \ldots, 20\}$ (resp. $\{20, 21, \ldots, 40\}$), $u_i^{\max} = 4$
- α^t is chosen randomly and uniformly in $[1, 2]$, β_i is chosen randomly and uniformly in $[0, 1]$
- $c^t = 1.5\lfloor \sin(\pi t/12) + 1 \rfloor$.

Results

Average convergence results for $J(x^k) - \mathcal{J}^*$, for different choices of n_k.

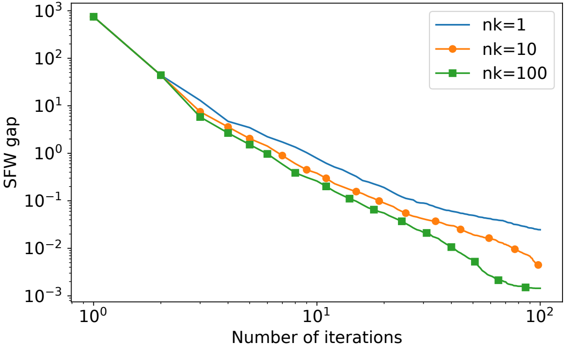

Figure: Stochastic Frank-Wolfe Algorithm with 100 iterations, expectation of the gap.

Thank you for your attention!