Distributed resolution of high-dimensional aggregative optimal control problems

Kang Liu[1]

joint work with J.F. Bonnans[2], N. Oudjane[3], L. Pfeiffer[2], and C. Wan[3]

[1]FAU DCN-AvH

[2]Université Paris-Saclay, CNRS, CentraleSupélec, Inria, L2S

[3]OSIRIS Department, EDF Lab Saclay, and FiME

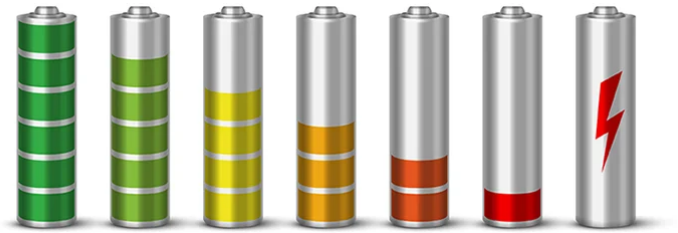
August 2024

Friedrich-Alexander-Universität
DYNAMICS, CONTROL, MACHINE LEARNING AND NUMERICS

Table of Contents

Table of Contents

Introduced example

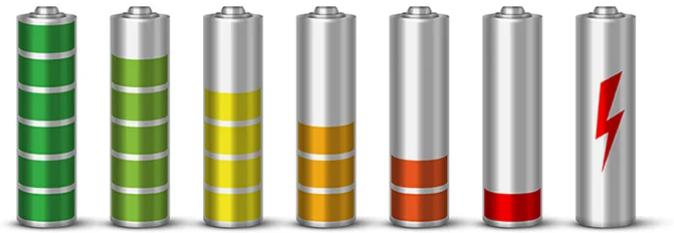


N non-symmetric batteries to be charged.

For battery i :

- State set: $S_i = \{s_i^{\text{in}}, \ldots, s_i^{\max}\}$;
- Control set: $U_i^t(s_i^t) = \{0, \ldots, \min(u_i^{\max}, s_i^{\max} - s_i^t)\}$;
- Dynamic: $s_i^{t+1} = s_i^t + u_i^t, \quad t = 0, \ldots T-1.$

Introduced example



N non-symmetric batteries to be charged.

For battery i :

- State set: $S_i = \{s_i^{\text{in}}, \ldots, s_i^{\max}\}$;
- Control set: $U_i^t(s_i^t) = \{0, \ldots, \min(u_i^{\max}, s_i^{\max} - s_i^t)\}$;
- Dynamic: $s_i^{t+1} = s_i^t + u_i^t, \quad t = 0, \ldots T-1.$

Cost function:

$$J(u) = \sum_{t=0}^{T-1} \underbrace{\alpha^t \left(\left(\frac{1}{N} \sum_{i=1}^{N} u_i^t \right) - c^t \right)^2}_{\text{Average charging levels' cost}} + \underbrace{\frac{1}{N} \sum_{i=1}^{N} \beta_i \left(s_i^T - s_i^{\max} \right)^2}_{\text{Final SoCs' preference}}.$$

Aggregative battery charging problem

Optimal control formulation:

$$\begin{cases} \inf_u J(u) = \sum_{t=0}^{T-1} \underbrace{\alpha^t \left(\left(\frac{1}{N} \sum_{i=1}^{N} u_i^t \right) - c^t \right)^2}_{\text{Average charging levels' cost}} + \underbrace{\frac{1}{N} \sum_{i=1}^{N} \beta_i \left(s_i^T - s_i^{\max} \right)^2}_{\text{Final SoCs' preference}}; \\ \text{s.t.} \quad \begin{cases} s_i^{t+1} = s_i^t + u_i^t, \\ u_i^t \in U_i^t(s_i^t), \end{cases} \quad \text{for } t = 0, 1, \ldots, T, \text{ and } i = 1, 2, \ldots, N. \end{cases}$$

Aggregative battery charging problem

Optimal control formulation:

$$\begin{cases} \inf_u J(u) = \sum_{t=0}^{T-1} \underbrace{\alpha^t \left(\left(\frac{1}{N} \sum_{i=1}^{N} u_i^t \right) - c^t \right)^2}_{\text{Average charging levels' cost}} + \underbrace{\frac{1}{N} \sum_{i=1}^{N} \beta_i \left(s_i^T - s_i^{\max} \right)^2}_{\text{Final SoCs' preference}}; \\ \text{s.t.} \quad \begin{cases} s_i^{t+1} = s_i^t + u_i^t, \\ u_i^t \in U_i^t(s_i^t), \end{cases} \quad \text{for } t = 0, 1, \ldots, T, \text{ and } i = 1, 2, \ldots, N. \end{cases}$$

Some classical approaches:

- **PMP**: Not applicable since $U_i^t(s_i^t)$ is discrete.

Aggregative battery charging problem

Optimal control formulation:

$$\begin{cases} \inf_u J(u) = \sum_{t=0}^{T-1} \underbrace{\alpha^t \left(\left(\frac{1}{N} \sum_{i=1}^{N} u_i^t \right) - c^t \right)^2}_{\text{Average charging levels' cost}} + \underbrace{\frac{1}{N} \sum_{i=1}^{N} \beta_i \left(s_i^T - s_i^{\max} \right)^2}_{\text{Final SoCs' preference}}; \\ \text{s.t.} \begin{cases} s_i^{t+1} = s_i^t + u_i^t, \\ u_i^t \in U_i^t(s_i^t), \end{cases} \quad \text{for } t = 0, 1, \ldots, T, \text{ and } i = 1, 2, \ldots, N. \end{cases}$$

Some classical approaches:

- **PMP**: Not applicable since $U_i^t(s_i^t)$ is discrete.
- **Dynamic Programming**: Suffers from the curse of dimensionality due to the N-dimensional Bellman equation.

Aggregative battery charging problem

Optimal control formulation:

$$\begin{cases} \inf_u J(u) = \sum_{t=0}^{T-1} \underbrace{\alpha^t \left(\left(\frac{1}{N} \sum_{i=1}^{N} u_i^t \right) - c^t \right)^2}_{\text{Average charging levels' cost}} + \underbrace{\frac{1}{N} \sum_{i=1}^{N} \beta_i \left(s_i^T - s_i^{\max} \right)^2}_{\text{Final SoCs' preference}}; \\ \text{s.t.} \begin{cases} s_i^{t+1} = s_i^t + u_i^t, \\ u_i^t \in U_i^t(s_i^t), \end{cases} \quad \text{for } t = 0, 1, \ldots, T, \text{ and } i = 1, 2, \ldots, N. \end{cases}$$

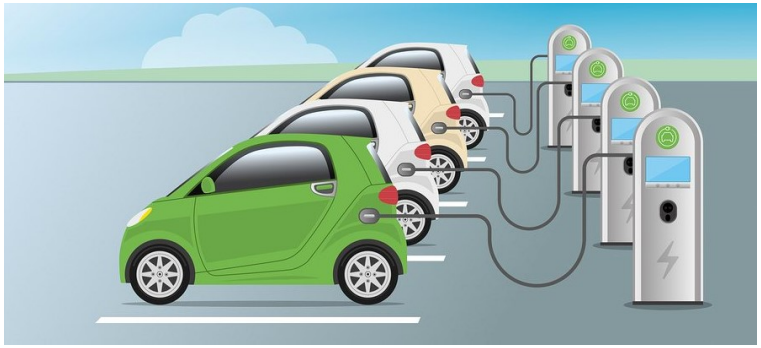
Issues:

- $U_i^t(s_i^t)$ is discrete (non-convexity);
- N is large (curse of dimensionality).

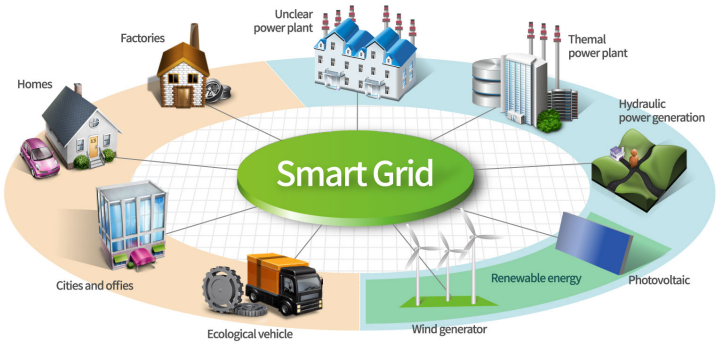
Key solutions:

- Convex relaxation;
- Distributed algorithm.

Applications

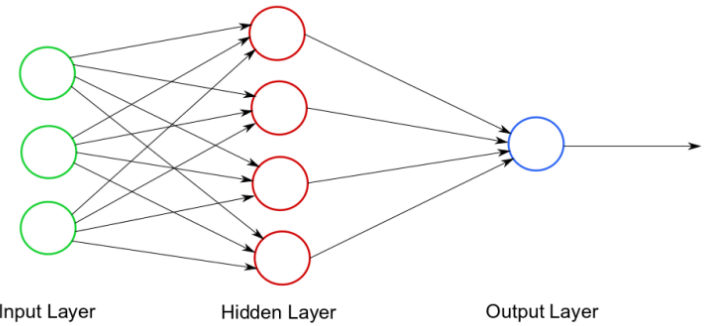


a. Electric vehicle charging



b. Smart grid

c. Social cooperation



d. Neural network

References: [Wang17], [Boyd11], [Mallat09], [Mei et al.18], [Chizat&Bach18], [Seguret et al.23], etc.

Table of Contents

A general optimization framework

$$\inf_{y\in\mathcal{Y}} J(y) = f\big(G(y)\big), \qquad \text{where:} \quad \begin{cases} G(y) = & \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} g_i(y_i) \\ \mathcal{Y} = & \prod_{i=1}^{N} \mathcal{X}_i. \end{cases} \tag{P}$$

Explanation:

- Individuals' number: $N \gg 1$;
- Individual strategy: $y_i \in \mathcal{Y}_i$;
- Individual contribution: $g_i(y_i)$;
- Social contribution (aggregate): $G(y)$;
- Social cost: $f(G(y))$.

Randomized relaxation

Primal problem:

$$\inf_{y\in\mathcal{Y}} J(y) = f\Big(\frac{1}{N}\sum_{i=1}^{N} g_i(y_i)\Big), \qquad \text{where } \mathcal{Y} = \prod_{i=1}^{N} \mathcal{Y}_i. \tag{P}$$

Randomized problem (convex):

$$\inf_{\mu\in\mathcal{P}} \tilde{J}(\mu) = f\Big(\frac{1}{N}\sum_{i=1}^{N} \int_{\mathcal{Y}_i} g_i \; d\mu_i\Big), \qquad \text{where } \mathcal{P} = \prod_{i=1}^{N} \mathcal{P}(\mathcal{Y}_i). \tag{PR}$$

Here, $\mathcal{P}(\mathcal{Y}_i)$ is the set of probability measure on $\mathcal{Y}_i$.

Randomized relaxation

Primal problem:

$$\inf_{y\in\mathcal{Y}} J(y) = f\Big(\frac{1}{N}\sum_{i=1}^{N} g_i(y_i)\Big), \qquad \text{where } \mathcal{Y} = \prod_{i=1}^{N} \mathcal{Y}_i. \tag{P}$$

Randomized problem (convex):

$$\inf_{\mu\in\mathcal{P}} \tilde{J}(\mu) = f\Big(\frac{1}{N}\sum_{i=1}^{N} \int_{\mathcal{Y}_i} g_i \; d\mu_i\Big), \qquad \text{where } \mathcal{P} = \prod_{i=1}^{N} \mathcal{P}(\mathcal{Y}_i). \tag{PR}$$

Here, $\mathcal{P}(\mathcal{Y}_i)$ is the set of probability measure on $\mathcal{Y}_i$.

Two key problems

1. Relaxation gap;
2. Reconstruction (from $\mu \in \mathcal{P}$ to $y \in \mathcal{Y}$).

Selection method

1. **Input**: $\mu = (\mu_i)_{i=1}^N \in \mathcal{P}$;
2. **Output**: $Y = (Y_i)_{i=1}^N \in \mathcal{Y}$, where $Y_i \sim \mu_i$ independently.

Notation: $Y = \mathit{Select}\ (\mu)$.

Selection method

1. **Input**: $\mu = (\mu_i)_{i=1}^N \in \mathcal{P}$;
2. **Output**: $Y = (Y_i)_{i=1}^N \in \mathcal{Y}$, where $Y_i \sim \mu_i$ independently.

Notation: $Y = \mathit{Select}\ (\mu)$.

Theorem (Bonnans-L.-Oudjane-Pfeiffer-Wan, 2023)

There exists a constant $C > 0$ independent of N such that for any $\epsilon > 0$,

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

$$\mathbb{P}\left(J(Y) - \tilde{J}(\mu) \leq \frac{C}{N} + \epsilon\right) \geq 1 - \exp\left(-\frac{N\epsilon^2}{C}\right).$$

As a consequence, $\mathrm{val(P)} - \mathrm{val(PR)} \leq C/N$.

The proof is based on McDiarmid's concentration inequality [McDiarmid, 1989].

Selection method

1. **Input**: $\mu = (\mu_i)_{i=1}^N \in \mathcal{P}$;
2. **Output**: $Y = (Y_i)_{i=1}^N \in \mathcal{Y}$, where $Y_i \sim \mu_i$ independently.

Notation: $Y = \mathit{Select}\ (\mu)$.

Theorem (Bonnans-L.-Oudjane-Pfeiffer-Wan, 2023)

There exists a constant $C > 0$ independent of N such that for any $\epsilon > 0$,

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

$$\mathbb{P}\left(J(Y) - \tilde{J}(\mu) \leq \frac{C}{N} + \epsilon\right) \geq 1 - \exp\left(-\frac{N\epsilon^2}{C}\right).$$

As a consequence, $\mathrm{val(P)} - \mathrm{val(PR)} \leq C/N$.

The proof is based on McDiarmid's concentration inequality [McDiarmid, 1989].
Remark: As $N \to \infty$,

1. the relaxation gap decreases to 0;
2. the probability of success increases to 1.

Table of Contents

Recall of the Frank-Wolfe algorithm

Consider a general convex optimization problem:

$$F^* = \inf_{x \in A} F(x),$$

where A is convex and compact, F is convex and L-smooth (∇F is L-Lipschitz).

Frank-Wolfe iteration [Dunn-Harshbarger 78]

At iteration k, given $x^k \in A$, do:

1. (Oracle) Find $\bar{x}^k \in \arg\min_{x \in A} \langle \nabla F(x^k), x \rangle$;
2. (Learning) Update $x^{k+1} = (1 - \omega_k) x^k + \omega_k \bar{x}^k$, with $\omega_k = 2/(k+2)$.

Convergence rate [1]: $F(x^k) - F^* \leq C/k$.

[1] same convergence rate for ω_k chosen by line-search [Jaggi 13]

Frank-Wolfe algorithm

- **Best-response** mapping: the resolution of the oracle

$$\bar{x}^k \in \arg\min_{x \in A} \langle \nabla F(x^k), x \rangle.$$

Frank-Wolfe algorithm

- **Best-response** mapping: the resolution of the oracle

$$\bar{x}^k \in \arg\min_{x \in A} \langle \nabla F(x^k), x \rangle.$$

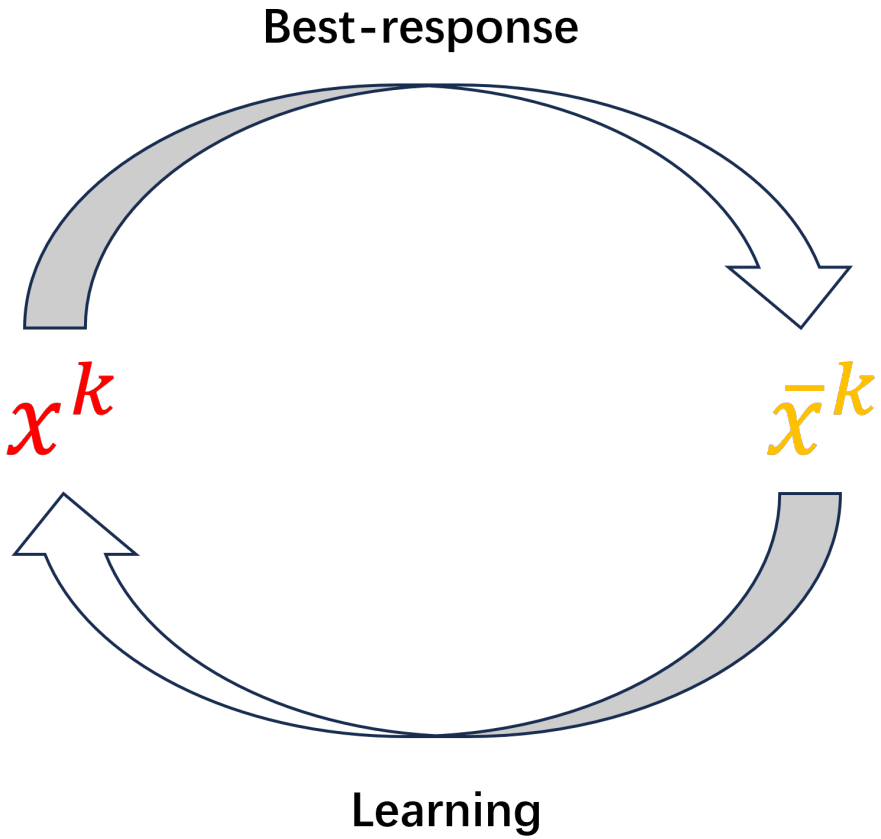


Figure: Diagram of the FW algorithm

FW algorithm for the relaxed problem

Recall the relaxed problem (convex):

$$\inf_{\mu\in\mathcal{P}} \tilde{J}(\mu) = f\Big(\frac{1}{N}\sum_{i=1}^{N}\int_{\mathcal{Y}_i} g_i \; d\mu_i\Big), \qquad \text{where } \mathcal{P} = \prod_{i=1}^{N}\mathcal{P}(\mathcal{Y}_i). \qquad \text{(PR)}$$

FW algorithm for the relaxed problem

Recall the relaxed problem (convex):

$$\inf_{\mu\in\mathcal{P}} \tilde{J}(\mu) = f\Big(\frac{1}{N}\sum_{i=1}^{N}\int_{\mathcal{Y}_i} g_i \; d\mu_i\Big), \qquad \text{where } \mathcal{P} = \prod_{i=1}^{N}\mathcal{P}(\mathcal{Y}_i). \qquad \text{(PR)}$$

First variation of $\tilde{J}$:

$$\left\langle \frac{\partial \tilde{J}(\mu)}{\partial \mu}, \bar{\mu} \right\rangle = \frac{1}{N}\sum_{i=1}^{N}\left\langle \nabla f(z), \int_{\mathcal{Y}_i} g_i d\bar{\mu}_i \right\rangle,$$

where $z = \frac{1}{N}\sum_{i=1}^{N}\int_{\mathcal{Y}_i} g_i \; d\mu_i$.

FW algorithm for the relaxed problem

Recall the relaxed problem (convex):

$$\inf_{\mu\in\mathcal{P}} \tilde{J}(\mu) = f\Big(\frac{1}{N}\sum_{i=1}^{N}\int_{\mathcal{Y}_i} g_i \; d\mu_i\Big), \qquad \text{where } \mathcal{P} = \prod_{i=1}^{N}\mathcal{P}(\mathcal{Y}_i). \qquad \text{(PR)}$$

First variation of $\tilde{J}$:

$$\left\langle \frac{\partial \tilde{J}(\mu)}{\partial \mu}, \bar{\mu} \right\rangle = \frac{1}{N}\sum_{i=1}^{N}\left\langle \nabla f(z), \int_{\mathcal{Y}_i} g_i d\bar{\mu}_i \right\rangle,$$

where $z = \frac{1}{N}\sum_{i=1}^{N}\int_{\mathcal{Y}_i} g_i \; d\mu_i$.

Individual best-response mapping (distributed)

Given any $z \in \mathcal{H}$, solve the following:

$$\mathbb{S}_i(z) = \arg\min_{y_i\in\mathcal{Y}_i} \langle \nabla f(z), g_i(y_i)\rangle.$$

Assume that $\mathbb{S}_i(z)$ is non-empty, and we can numerically find at least one element in $\mathbb{S}_i(z)$.

FW algorithm for the relaxed problem

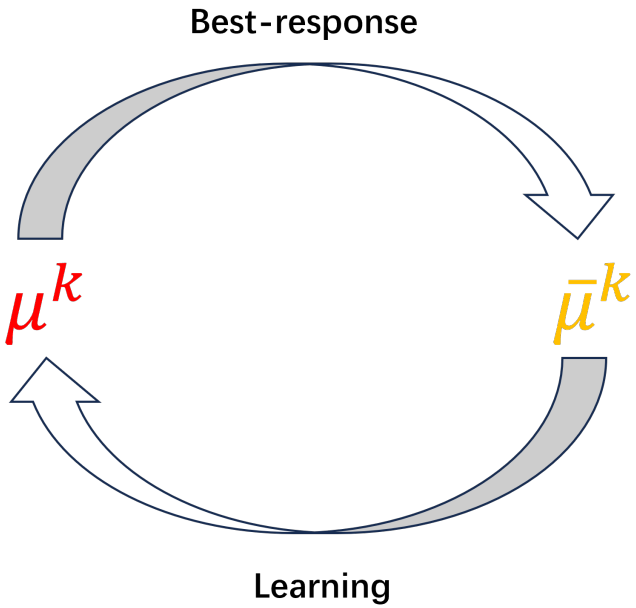


Figure: FW algorithm for (PR)

Best-response

1. Aggregate: $z^k = \frac{1}{N}\sum_{i=1}^{N}\int_{\mathcal{Y}_i} g_i \; d\mu^k$;
2. Parallel computation: $\bar{y}_i^k \in \mathbb{S}_i(z^k)$;
3. $\bar{\mu}^k = (\delta_{\bar{y}_1^k}, \ldots, \delta_{\bar{y}_N^k})$.

Learning

4. $\mu^{k+1} = \frac{k}{k+2}\mu^k + \frac{2}{k+2}\bar{\mu}^k$.

FW algorithm for the relaxed problem

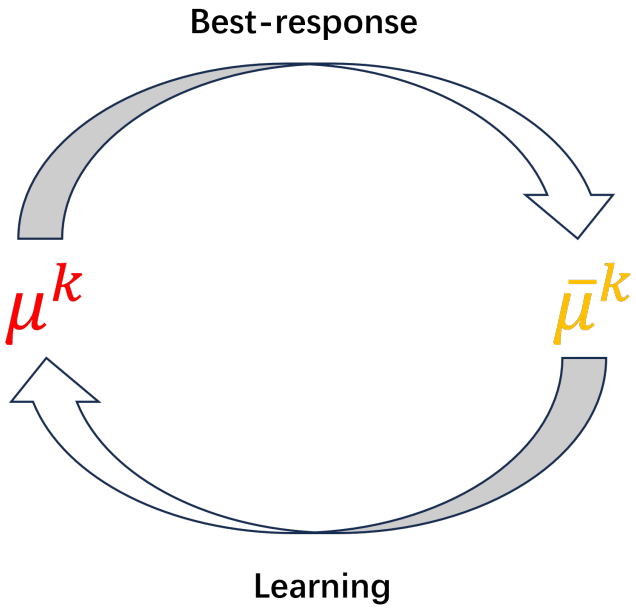


Figure: FW algorithm for (PR)

Best-response

1. Aggregate: $z^k = \frac{1}{N}\sum_{i=1}^N \int_{\mathcal{Y}_i} g_i \; d\mu^k$;
2. Parallel computation: $\bar{y}_i^k \in \mathbb{S}_i(z^k)$;
3. $\bar{\mu}^k = (\delta_{\bar{y}_1^k}, \ldots, \delta_{\bar{y}_N^k})$.

Learning

4. $\mu^{k+1} = \frac{k}{k+2}\mu^k + \frac{2}{k+2}\bar{\mu}^k$.

Convergence

$\tilde{J}(\mu^K) - \text{val(PR)} \leq C/K$.

Memory overflow issue

$\text{supp}(\mu_i^{k+1}) = \text{supp}(\mu_i^k) \cup \{\bar{y}_i^k\}$.

Stochastic FW (SFW) algorithm for the primal problem

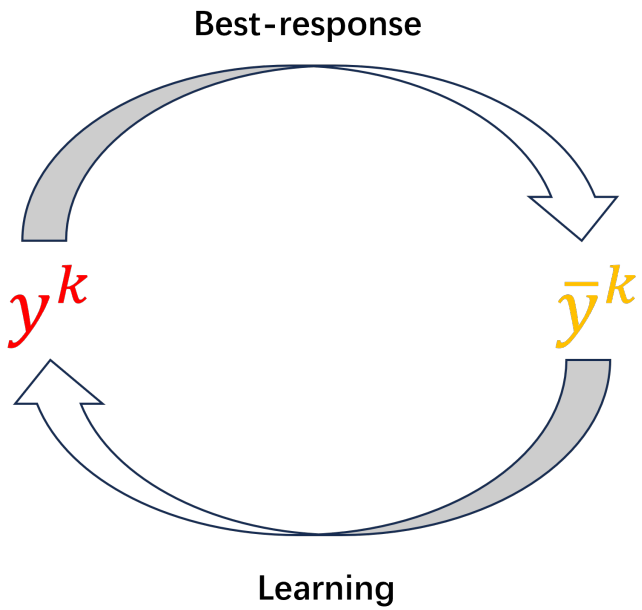


Figure: SFW algorithm for (P)

Best-response

1. Aggregate: $z^k = \frac{1}{N}\sum_{i=1}^N g_i(y_i^k)$;
2. Parallel computation: $\bar{y}_i^k \in \mathbb{S}_i(z^k)$.

Learning

3. $\mu_i = \frac{k}{k+2}\delta_{y_i^k} + \frac{2}{k+2}\delta_{\bar{y}_i^k}$.

Selection

4. Fixing $n_k \in \mathbb{N}_+$, for $j = 1, \ldots, n_k$, get $\hat{y}^{k,j} = \mathit{Select}\ (\mu)$;
5. Update $y^{k+1} \in \arg\min\left\{J(\hat{y}^{k,j}) \,\middle|\, j = 1, 2, \ldots, n_k\right\}$.

Remark: If $n_k = 1$ for all k, then (4)-(5) $\Longleftrightarrow$ $y^{k+1} = \mathit{Select}\ (\mu)$.

Theorem (Bonnans-L.-Oudjane-Pfeiffer-Wan, 2023)

There exists $C > 0$ independent of N such that for all $K = 1, \ldots, 2N$,

$$\mathbb{E}[\gamma_K] \leq \frac{C}{K}, \quad \text{where } \gamma_K = J(y^K) - \text{val(PR)}.$$

Moreover, for all $\epsilon > 0$,

$$\mathbb{P}\left[\gamma_K < \frac{C}{K} + \epsilon\right] \geq 1 - \exp\left(\frac{-\epsilon^2 N}{2(v_K + \epsilon m_K/3)}\right).$$

where v_K, m_K (independent of N) are small when $\{n_k\}_{k=1}^K$ are large.

Theorem (Bonnans-L.-Oudjane-Pfeiffer-Wan, 2023)

There exists $C > 0$ independent of N such that for all $K = 1, \ldots, 2N$,

$$\mathbb{E}[\gamma_K] \leq \frac{C}{K}, \quad \text{where } \gamma_K = J(y^K) - \text{val(PR)}.$$

Moreover, for all $\epsilon > 0$,

$$\mathbb{P}\left[\gamma_K < \frac{C}{K} + \epsilon\right] \geq 1 - \exp\left(\frac{-\epsilon^2 N}{2(v_K + \epsilon m_K/3)}\right).$$

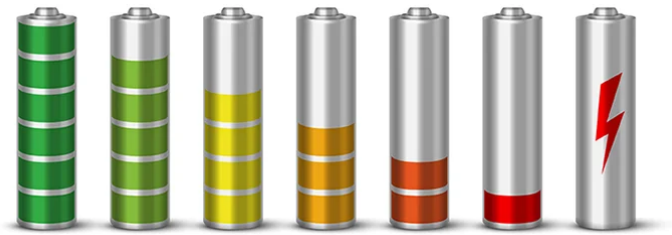
where v_K, m_K (independent of N) are small when $\{n_k\}_{k=1}^K$ are large.

A special choice

Taking $n_k \geq \max\left(\frac{Ak^2}{N}, 1\right)$, then, $\mathbb{P}\left[\gamma_K < \frac{2C}{K}\right] \xrightarrow[A\to\infty]{} 1$.

Table of Contents

Aggregative battery charging



N non-symmetric batteries to be charged.

- State set: $S_i = \{s_i^{\text{in}}, \ldots, s_i^{\max}\}$;
- Control set: $U_i^t(s_i^t) = \{0, \ldots, \min(u_i^{\max}, s_i^{\max} - s_i^t)\}$;
- Dynamic: $s_i^{t+1} = s_i^t + u_i^t, \quad t = 0, \ldots T-1.$

Cost function:

$$J(u) = \sum_{t=0}^{T-1} \underbrace{\alpha^t \left(\left(\frac{1}{N} \sum_{i=1}^{N} u_i^t \right) - c^t \right)^2}_{\text{Average charging levels' cost}} + \underbrace{\frac{1}{N} \sum_{i=1}^{N} \beta_i \left(s_i^T - s_i^{\max} \right)^2}_{\text{Final SoCs' preference}}.$$

Subproblems

Recall the subproblems: Given any z in the domain of f,

$$\mathbb{S}_i(z) = \arg\min_{y_i \in \mathcal{Y}_i} \langle \nabla f(z), g_i(y_i) \rangle.$$

In the aggregate battery charging case: for $i = 1, 2, \ldots, N$,

$$\begin{cases} \inf_{u_i} \sum_{t=0}^{T-1} 2\alpha^t (z^t - c^t) u_i^t + \beta_i \left(s_i^T - s_i^{\max} \right)^2; \\ \text{s.t.} \quad \begin{cases} s_i^{t+1} = s_i^t + u_i^t, \\ u_i^t \in U_i^t(s_i^t), \end{cases} \quad \text{for } t = 0, 1, \ldots, T. \end{cases}$$

This is a discrete-time finite-state 1-dimsnsional optimal control problem!

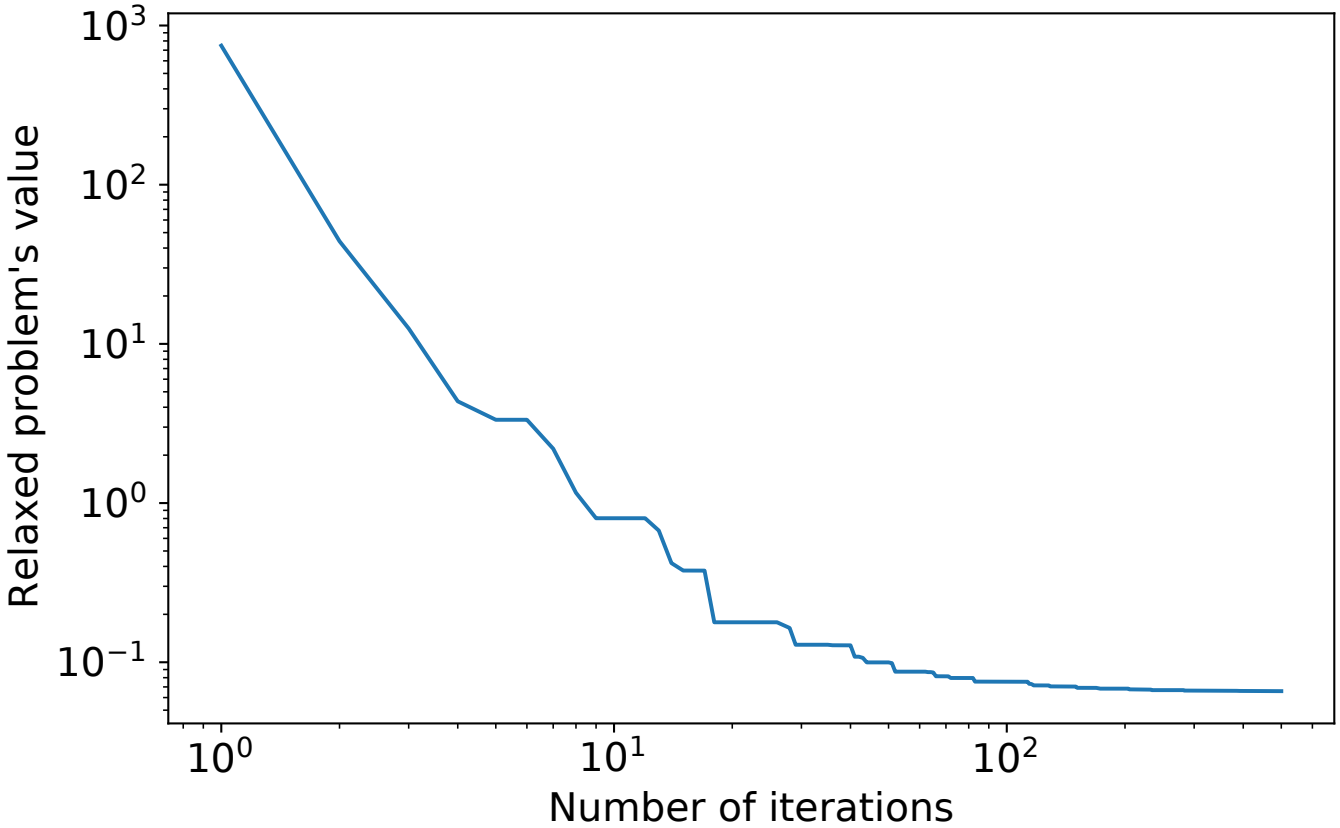
Parameters setting

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

Attention: If we solve directly by the dynamic programming principle, $N = 100$ leads to a 100-dimensional Bellman's equation!

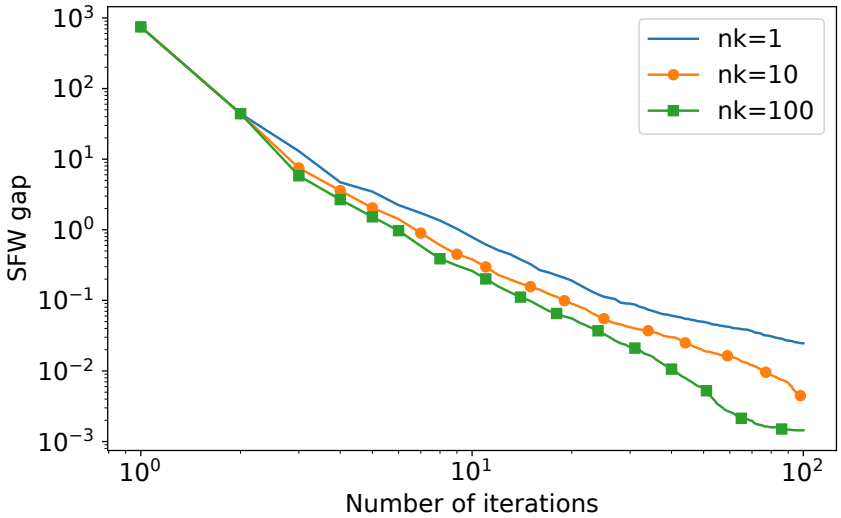
By the FW (SFW) algorithm, at each iteration, we solve at most 100 1-dimensional Bellman's equations!

FW for battery charging problem

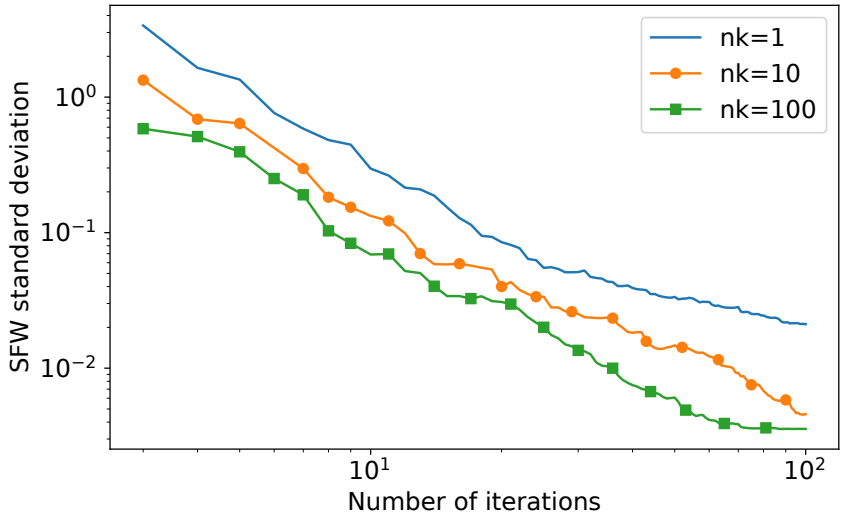


The FW algorithm for the relaxed problem with 500 iterations

SFW for battery charging problem



The SFW Algorithm with 100 iterations, **expectation** of the gap.



The SFW algorithm with 100 iterations, **standard deviation** of the gap.

Bibliography


J.F. Bonnans, K. Liu, N. Oudjane, L. Pfeiffer and C. Wan. *Large-scale nonconvex optimization: randomization, gap estimation, and numerical resolution*. SIOPT, 2023.


K. Liu, N. Oudjane and L. Pfeiffer. *Decomposed resolution of fnite-state aggregative optimal control problems*. CT23.

Thank You!

Appendix

Sketch of proof

Introduce $\mu_k = \delta_{x^k}$ and $\hat{\mu}^k = (1-\omega_k)\mu^k + \omega_k \delta_{\bar{x}^k}$.

Step 1. $\gamma_{k+1} \leq \gamma_k + a_k + b_k + c_k$, where

$$\text{selection error:} \qquad a_k = \frac{1}{n_k} \sum_{j=1}^{n_k} \Big(J(\hat{x}^{k,j}) - \mathbb{E}\big[J(\hat{x}^{k,j}) \,|\, x^k \big] \Big),$$

$$\text{variance:} \qquad b_k = \frac{1}{n_k} \sum_{j=1}^{n_k} \Big(\mathbb{E}\big[J(\hat{x}^{k,j}) \,|\, x^k \big] - \tilde{J}(\hat{\mu}^k) \Big),$$

$$\text{bias:} \qquad c_k = \tilde{J}(\hat{\mu}^k) - \tilde{J}(\mu^k) = \tilde{J}(\hat{\mu}^k) - J(x^k).$$

We then get $\gamma_K \leq \frac{4C_1}{K} + S_K$, where $S_K = \sum_{k=0}^{K-1} \frac{(k+1)(k+2)}{K(K+1)} a_k$.

Step 2. Decompose S_K to the sum of small variance martingales.

Step 3. Apply a variant of McDiarmid's inequality for martingales [**?**, B. Delyon] to S_K.