



HAL
open science

Decentralized resolution of finite-state, non-convex, and aggregative optimal control problems

Kang Liu, Nadia Oudjane, Laurent Pfeiffer

► **To cite this version:**

Kang Liu, Nadia Oudjane, Laurent Pfeiffer. Decentralized resolution of finite-state, non-convex, and aggregative optimal control problems. 2022. hal-03642127v1

HAL Id: hal-03642127

<https://hal.science/hal-03642127v1>

Preprint submitted on 14 Apr 2022 (v1), last revised 7 Jul 2023 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Decentralized resolution of finite-state, non-convex, and aggregative optimal control problems

Kang Liu Nadia Oudjane Laurent Pfeiffer

Abstract—A general class of large-scale, nonconvex, and non-smooth optimization problems is introduced. It has the form of a multi-agent problem, where the agents interact through an aggregative term. A convex relaxation of the problem is provided together with an estimate of the relaxation gap. A numerical method, called stochastic Frank-Wolfe algorithm, is presented. The method allows to find approximate solutions of the original problem in a decomposed fashion. The convergence of the method is guaranteed from a theoretical point of view. An aggregative deterministic optimal control problem is formulated, with discrete state-space and discrete time. It is shown that the stochastic Frank-Wolfe algorithm can be applied to the optimal control problem; in particular, it amounts to solve at each iteration a series of small-scale optimal control problems, corresponding to each agent. These sub-problems are solved by dynamic programming. Numerical results are presented, for a toy model of the charging management of a battery fleet.

I. INTRODUCTION

This article is dedicated to a class of aggregative optimal control problems. They will be formulated as particular instances of optimization problems of the following form:

$$\inf_{x \in \mathcal{X}} J(x) := f(G(x)), \quad (\text{P})$$

where: $\mathcal{X} = \prod_{i=1}^N \mathcal{X}_i$ and $G(x) = \frac{1}{N} \sum_{i=1}^N g_i(x_i)$.

We will first focus on this general form; later on, the variables x_i will be state-control trajectories. In the general problem (P), the sets \mathcal{X}_i are given and the maps g_i are defined from \mathcal{X}_i to some Hilbert space \mathcal{E} . The function f is defined from \mathcal{E} to \mathbb{R} . An interpretation of problem (P) is as follows: N is the number of agents; the agents are indexed by i and each variable $x_i \in \mathcal{X}_i$ corresponds to the decision attributed to agent i . The mapping g_i is the *contribution* of agent i to some common good, defined by $\frac{1}{N} \sum_{i=1}^N g_i(x_i)$. We will refer to it as the *aggregate*. The function f is the cost associated with the aggregate. We refer to it as the *social cost*. We will only assume that f is convex, differentiable, with a Lipschitz-continuous gradient. Concerning the sets \mathcal{X}_i and

the mappings g_i , only the boundedness of $g_i(\mathcal{X}_i)$ will be required.

Various application problems take this general form, see [10]. In energy applications, the agents i are small production or consumption units. The contribution $g_i(x_i)$ is typically a vector containing the operational cost (incurred by agent i) and its energy production. The aggregate is then the average cost of the agents and their average production (over time).

Problem (P) was investigated by Mengdi Wang in [10]. She introduced a convex relaxation of the problem and obtained a relaxation gap based on Shapley-Folkman's lemma [9]. Other works in the literature have used this lemma to find estimates of duality gaps, let us mention the pioneering work by Aubin and Ekeland [2]. Wang proposed in [10] a numerical method based on a resolution of the dual of the problem. We recently obtained an improvement of the gap estimate of Wang in [4] and proposed a different algorithm, based on the relaxed optimization problem (rather than on the dual). We demonstrated the convergence of our algorithm. A thorough comparison of the two numerical approaches can be found in [4, Section 4.3]. A common property of the numerical methods of [4] and [10] is the following: they are iterative; at each iteration, N subproblems of small scale are solved independently from each other. For this reason, we describe our method as *decentralized* (the terminology *distributed* is also employed in the literature). There is an important literature on decentralized algorithms for aggregative problems, in various application settings, let us mention for example [6], for a flight formation problem and [1], for a smart-grid management problem. Let us underline that most contributions do not provide convergence results.

The main novelty of this article, in comparison with our work [4], is the introduction of a general model for aggregative and deterministic optimal control problems with finite state-space and discrete time. They are of the form (P). These optimal control problems cannot be addressed with the standard dynamic programming approach, since the complexity increases exponentially with N ; this phenomenon is the well-known curse of dimensionality. Instead, we show that our stochastic Frank-Wolfe algorithm can be conveniently applied to the aggregative problem: at each iteration, N optimal control sub-problems of small size need to be solved, which can be done by dynamic programming. We focus here on finite state-space and discrete-time optimal control problems for their simplicity. Many applications involve continuous dynamical systems (possibly nonlinear) which can be naturally discretized in the general discrete form investigated in this article. Let us emphasize that only

N. Oudjane was partially supported by the FiME Lab Research Initiative (Institut Europlace de Finance). This article benefited from the support of the FMJH Program PGMO and from the support to this program from EDF.

K. Liu is with CMAP, Ecole Polytechnique, CNRS, Institut Polytechnique de Paris, Palaiseau, France and Inria, Laboratoire des Signaux et des Systèmes, CentraleSupélec, CNRS, Université Paris-Saclay, Gif-sur-Yvette, France, kang.liu@polytechnique.edu

N. Oudjane is with OSIRIS department, EDF Lab, Paris-Saclay, France, nadia.oudjane@edf.fr

L. Pfeiffer is with Inria, Laboratoire des Signaux et des Systèmes, CentraleSupélec, CNRS, Université Paris-Saclay, Gif-sur-Yvette, France, laurent.pfeiffer@inria.fr

the social cost f needs to be convex and smooth, thus our methodology can be applied to aggregative optimal control problem with agents with nonconvex individual preferences.

The idea of decomposing a large-scale optimal control problem into small-size optimal control problems which are tractable by dynamic programming can be found in [3] and in [7] in the context of convex stochastic optimal control problems. Let us mention that several works propose to approximate large-scale optimal control problems by a mean-field optimal control problem, which can be understood as a limit model as the number of agents go to infinity. The mean-field model involves the distribution of the agents with respect to their state variable, at each time. See for example [8]. We do not follow this approach from a numerical point of view, but a connection between our relaxed problem and a mean-field approximation could be established (see the discussion in Remark 2).

II. ABSTRACT OPTIMIZATION PROBLEM AND ASSUMPTIONS

We first make some non-restrictive structural assumptions on (P). The aggregate space \mathcal{E} is assumed to be the Cartesian product of M Hilbert spaces $\mathcal{E}_1, \dots, \mathcal{E}_M$. Moreover, we suppose that f is of the form

$$f(y) = \sum_{j=1}^M f_j(y_j), \quad \forall y = (y_1, \dots, y_M) \in \mathcal{E}.$$

The functions f_j are defined from \mathcal{E}_j to \mathbb{R} , for $j = 1, \dots, M$. Finally, the contribution mappings g_i are of the form $g_i(x_i) = (g_{ij}(x_i))_{j=1, \dots, M}$. Therefore, the cost functional J writes:

$$J(x) = \sum_{j=1}^M f_j \left(\frac{1}{N} \sum_{i=1}^N g_{ij}(x_i) \right).$$

Some general notations will be used all along the article: given two subsets A and $B \subseteq \mathcal{E}$, we denote by $A + B$ their Minkowski sum. Given $\lambda \in \mathbb{R}$, we denote λA the set defined by $\{\lambda a \mid a \in A\}$. We denote by $\text{conv}(A)$ the convex hull of A . Next we introduce the main assumptions of the work. For any $i = 1, \dots, N$ and for any $j = 1, \dots, M$, we denote

$$Y_{ij} = \{g_{ij}(x_i) \mid x_i \in \mathcal{X}_i\} \quad \text{and} \quad Y_j = \frac{1}{N} \sum_{i=1}^N Y_{ij}.$$

Assumption A. For $i = 1, 2, \dots, N$ and $j = 1, 2, \dots, M$:

- The range set Y_{ij} has finite diameter d_{ij} .
- The function f_j is L_j -Lipschitz on $\text{conv}(Y_j)$.
- The function f_j is continuously differentiable on a neighborhood of $\text{conv}(Y_j)$, and ∇f_j is \tilde{L}_j -Lipschitz on $\text{conv}(Y_j)$.

We next define two constants $C_0 > 0$ and $C_1 > 0$ by

$$C_0 = \sum_{j=1}^M \left(L_j \max_{1 \leq i \leq N} \{d_{ij}\} \right),$$

$$C_1 = \frac{1}{N} \sum_{j=1}^M \left(\tilde{L}_j \sum_{i=1}^N d_{ij}^2 \right).$$

Assumption B. For all $j = 1, \dots, M$, the function $f_j: \mathcal{E}_j \rightarrow \mathbb{R}$ is convex.

Assumption C. For all $i = 1, \dots, N$ and for all $y \in \text{conv}(G(\mathcal{X}))$, the problem

$$\inf_{x_i \in \mathcal{X}_i} \langle \nabla f(y), g_i(x_i) \rangle \quad (1)$$

has at least a solution. For all $i = 1, \dots, N$, we fix a map $\mathbb{S}_i: \text{conv}(G(\mathcal{X})) \mapsto \mathcal{X}_i$ such that, for any $y \in \text{conv}(G(\mathcal{X}))$, $\mathbb{S}_i(y)$ is a solution to (1).

III. CONVEX RELAXATION

We introduce in this section a convex relaxation of problem (P). It will motivate the stochastic Frank-Wolfe algorithm presented in the following section. The reader only interested in a practical implementation of the algorithm can move to the next section. We first need to reformulate problem (P). Let us define

$$\mathcal{Y}_i = g_i(\mathcal{X}_i), \quad \forall i = 1, \dots, N, \quad \text{and} \quad \mathcal{Y} = \frac{1}{N} \sum_{i=1}^N \mathcal{Y}_i.$$

Problem (P) is equivalent to

$$\inf_{y \in \mathcal{E}} f(y), \quad \text{subject to: } y \in \mathcal{Y}.$$

Indeed, by definition of \mathcal{Y} , any $y \in \mathcal{E}$ lies in \mathcal{Y} if and only if there exists $x \in \mathcal{X}$ such that $y = \frac{1}{N} \sum_{i=1}^N g_i(x_i)$. For such an x , we have $f(y) = J(x)$. It is natural to consider the following relaxation:

$$\inf_{y \in \mathcal{E}} f(y), \quad \text{subject to: } y \in \text{conv}(\mathcal{Y}). \quad (\text{PR})$$

Under Assumption B, the relaxed problem is a convex optimization problem. Let J^* denote the value of problem (P) and let \mathcal{J}^* denote the value of problem (PR). We have the following result.

Proposition 1. Let Assumption A hold true. Then

$$\mathcal{J}^* \leq J^* \leq \mathcal{J}^* + \frac{C_1}{2N}.$$

Proof. The first inequality is straightforward, and the second one is proved in [4, Proposition 2.6]. \square

Let us mention that a more precise upper bound is given in [4, Theorem 4.4]. The result of Proposition 1 is to be related to Shapley-Folkman's lemma and more precisely to Starr's corollary, which gives a bound of the distance to \mathcal{Y} of a point in $\text{conv}(\mathcal{Y})$. Assuming that the coefficients d_{ij} (appearing in Assumption A) are uniformly bounded, we see that C_1 is also bounded, and thus the gap estimate $\frac{C_1}{2N}$ goes to zero as N goes to infinity. In words, there is a convexification of the problem as the number of agents increases.

Remark 2. Consider the particular case where the sets \mathcal{X}_i and the contribution functions g_i do not depend on i . Then, $\mathcal{Y}_1 = \dots = \mathcal{Y}_N$. It follows that

$$\begin{aligned} \text{conv}(\mathcal{Y}) &= \text{conv} \left(\frac{1}{N} \sum_{i=1}^N \mathcal{Y}_1 \right) = \frac{1}{N} \sum_{i=1}^N \text{conv}(\mathcal{Y}_1) \\ &= \text{conv}(\mathcal{Y}_1). \end{aligned}$$

In this case, the relaxed problem does not depend on N , it can be interpreted as a mean-field relaxation, i.e., it can be interpreted as the limit problem as the number of agents $N \rightarrow \infty$.

From Proposition 1, we note that any ϵ -solution of problem (PR) is an ϵ -solution of problem (P) as soon as it is feasible for problem (P). Since problem (PR) is convex, it is easier to handle numerically. Algorithm 1 generates a minimizing sequence $(y^k)_{y \in \mathcal{N}}$ in $\text{conv}(\mathcal{Y})$ for the relaxed problem, by a direct application of the Frank-Wolfe algorithm [5]. The general idea of Algorithm 2 is to introduce an approximation step at each iteration, to recover points in \mathcal{Y} .

Algorithm 1: Frank-Wolfe Algorithm for the relaxed problem

Initialization: $y^0 \in \text{conv}(\mathcal{Y})$;
for $k = 0, 1, 2, \dots$ **do**
 Find a solution \bar{y}^k to the subproblem

$$\inf_{y \in \text{conv}(\mathcal{Y})} \langle \nabla f(y^k), y \rangle \quad (2)$$

 Choose $\omega_k \in [0, 1]$;
 Set $y^{k+1} = (1 - \omega_k)y^k + \omega_k \bar{y}^k$;
end

The algorithm is known to converge for various choices of the parameter ω_k . In particular, for $\omega_k = 2/(k+2)$, one can show the existence of a constant $C > 0$ such that for any k , $f(y^k) \leq \mathcal{J}^* + \frac{C}{k}$. Besides the guaranty of convergence of the algorithm, its interest lies in the decomposability of the sub-problems to be solved at each iteration. Problem (2) is indeed equivalent to

$$\inf_{x \in \mathcal{X}} \left\langle \nabla f(y^k), \sum_{i=1}^N g_i(x_i) \right\rangle. \quad (3)$$

Obviously, x is a solution if and only if x_i is a solution to (1) (with $y = y^k$). Therefore a solution to (2) is given by

$$\bar{y}^k = \frac{1}{N} \sum_{i=1}^N g_i(\mathbb{S}_i(y^k)). \quad (4)$$

Note that $\bar{y}^k \in \mathcal{Y}$. However, even if y^k also belonged to \mathcal{Y} , there would be no reason to have $y^{k+1} \in \mathcal{Y}$.

The stochastic Frank-Wolfe algorithm (Algorithm 2) introduced in the next section allows us to overcome this difficulty. At the iteration k , a point x^k has been constructed, with aggregate $y^k = \frac{1}{N} \sum_{i=1}^N g_i(x_i^k)$. The same sub-problems are solved, yielding a point $\bar{x}^k = (\mathbb{S}_1(y^k), \dots, \mathbb{S}_N(y^k))$ with aggregate \bar{y}^k . Next, the algorithm generates a sample of n_k points independently and identically distributed (i.i.d.) in \mathcal{X} , denoted $\hat{x}^{k,j} = (\hat{x}_i^{k,j})_{i=1, \dots, N}$, with $j = 1, \dots, n_k$. The point $\hat{x}_i^{k,j}$ is equal to x_i^k with probability $1 - \omega_k$ and to $\bar{x}_i^{k,j}$ with probability ω_k . In practice, we simulate Nn_k i.i.d. random variables $\lambda_i^{k,j} \sim \text{Bern}(\omega_k)$, where $\text{Bern}(\omega_k)$ denotes the Bernoulli distribution of parameter $\omega_k \in [0, 1]$ and we set

$$\hat{x}_i^{k,j} = (1 - \lambda_i^{k,j})x_i^k + \lambda_i^{k,j}\bar{x}_i^k.$$

Then x^{k+1} is taken as a minimizer of J over the union of the set of points randomly generated and $\{x^k\}$.

IV. STOCHASTIC FRANK-WOLFE ALGORITHM

We provide in Algorithm 2 an explicit implementation of our stochastic Frank-Wolfe algorithm.

Algorithm 2: Stochastic Frank-Wolfe Algorithm

Initialization: $x^0 \in \mathcal{X}$;
for $k = 0, 1, 2, \dots$ **do**
 Step 1: Resolution of the subproblems.
 Compute $y^k = \frac{1}{N} \sum_{i=1}^N g_i(x_i^k)$;
 for $i = 1, 2, \dots, N$ **do**
 Compute $\bar{x}_i^k = \mathbb{S}_i(y^k)$;
 end
 Step 2: Update.
 Choose $n_k \in \mathbb{N}^*$ and $\omega_k \in [0, 1]$;
 for $j = 1, 2, \dots, n_k$ **do**
 for $i = 1, 2, \dots, N$ **do**
 Simulate $\lambda_i^{k,j} \sim \text{Bern}(\omega_k)$, independently
 of all previously defined random
 variables;
 Set $\hat{x}_i^{k,j} = (1 - \lambda_i^{k,j})x_i^k + \lambda_i^{k,j}\bar{x}_i^k$;
 end
 Set $\hat{x}^{k,j} = (\hat{x}_i^{k,j})_{i=1, \dots, N}$;
 end
 Find $x^{k+1} \in \text{argmin}\{J(x) \mid x \in X^k\}$, where
 $X^k = \{\hat{x}^{k,j}, j = 1, 2, \dots, n_k\} \cup \{x^k\}$;
end

We have the following result, proved in [4, Theorem 3.7].

Theorem 3. *Let Assumptions A, B, and C hold true. Assume that $\omega_k = \frac{2}{k+2}$, for all $k \in \mathbb{N}$ in Algorithm 2. Then, for all $K = 1, \dots, 2N$,*

$$\mathbb{E}[\gamma_K] \leq \frac{4C_1}{K}, \quad \text{where } \gamma_K = J(x^K) - \mathcal{J}^*.$$

Moreover, for all $\epsilon > 0$,

$$\mathbb{P}\left[\gamma_K < \frac{4C_1}{K} + \epsilon\right] \geq 1 - \exp\left(\frac{-\epsilon^2 N}{2(v_K + \epsilon m_K/3)}\right),$$

where the constants m_K and v_K are given by

$$v_K = \frac{2C_0^2}{K^2(K+1)^2} \left(\sum_{k=1}^{K-1}, \frac{k(k+1)^2}{n_k} \right)$$

$$m_K = \frac{C_0}{K(K+1)} \left(\max_{k=1, \dots, K-1} \frac{(k+1)(k+2)}{n_k} \right).$$

Note that the constants v_K and m_K can be made arbitrarily small by choosing sufficiently large values of $(n_k)_{k=0, \dots, K}$. Thus for arbitrarily small values of $\epsilon > 0$ and $\epsilon' > 0$, one can choose appropriate numbers of random simulations so that $\mathbb{P}[\gamma_{2N} < \frac{2C_1}{N} + \epsilon] \geq 1 - \epsilon'$.

Remark 4. Theorem 3 focuses on the choice of stepsize $\omega_k = 2/(k+2)$, which we have utilized in the numerical

simulations. It is also possible to determine ω_k by line search, see [4, Remark 3.10].

V. AGGREGATIVE OPTIMAL CONTROL

In this section we describe a general aggregative optimal control problem with finite state-space and discrete time. The time steps of the problem are denoted $\{0, 1, \dots, T\}$.

A. State-control sets

We fix in this subsection an agent i and describe its state-control feasible set \mathcal{X}_i . For the description of \mathcal{X}_i , we need the following:

- a finite set, called *state set*, denoted S_i ,
- a finite set, called *control set*, denoted U_i
- a sequence of $(T+1)$ subsets of S_i , called *feasible state sets*, denoted $(S_i^t)_{t=0,1,\dots,T}$
- for each $t = 0, 1, \dots, T-1$, a mapping $U_i^t: S_i^t \rightarrow 2^{U_i}$ (i.e. for $s_i^t \in S_i^t$, $U_i^t(s_i^t)$ is a subset of U_i) called *feasible controls* at state s_i^t .
- for each $t = 0, 1, \dots, T-1$, a function $\pi_i^t: S_i \times U_i \rightarrow S_i$, called *transition function*.

We call feasible *state-control trajectory* an element $x_i = (s_i, u_i)$, where $s_i = (s_i^0, \dots, s_i^T) \in (S_i)^{T+1}$ and $u_i = (u_i^0, \dots, u_i^{T-1}) \in (U_i)^T$, such that

- $s_i^t \in S_i^t$, for $t = 0, \dots, T$
- $u_i^t \in U_i^t(s_i^t)$, for $t = 0, \dots, T-1$
- $s_i^{t+1} = \pi_i^t(s_i^t, u_i^t)$, for $t = 0, \dots, T-1$.

We denote by \mathcal{X}_i the set of feasible state-control trajectories. We assume that it is non-empty. We set $\mathcal{X} = \prod_{i=1}^N \mathcal{X}_i$.

The non-emptiness of \mathcal{X}_i can be verified by constructing a sequence of sets $(\tilde{S}_i^t)_{t=0,\dots,T}$ with the following backward procedure. Set $\tilde{S}_i^T = S_i^T$. For $t = T-1, \dots, 0$, define \tilde{S}_i^t as the set of points $s_i^t \in S_i^t$ for which there exists $u_i^t \in U_i^t(s_i^t)$ such that $\pi_i^t(s_i^t, u_i^t) \in \tilde{S}_i^{t+1}$. Then, \mathcal{X}_i is non-empty if and only if \tilde{S}_i^0 is non-empty. Obviously, one can replace the sets S_i^t by the sets \tilde{S}_i^t in the definition of \mathcal{X}_i , without changing the resulting set of feasible state-control trajectories. Therefore, without loss of generality, we can consider the following assumption.

Assumption 1. *The sets $(S_i^t)_{t=0,\dots,T}$ are non-empty. Moreover, for all $t = 0, \dots, T-1$, for all $s_i^t \in S_i^t$, there exists $u_i^t \in U_i^t(s_i^t)$ such that $\pi_i^t(s_i^t, u_i^t) \in S_i^{t+1}$.*

B. Cost and contribution functions

Let us consider again an agent i . Its *individual cost* is described by $T+1$ functions:

- $\ell_i^t: S_i \times U_i \rightarrow \mathbb{R}$, for $t = 0, \dots, T-1$
- $\ell_i^T: S_i \rightarrow \mathbb{R}$

A sequence of Hilbert spaces $(\mathcal{E}_t)_{t=0,\dots,T}$ is supposed to be given. The $T+1$ *contribution* functions of the agent i are as follows:

- $h_i^t: S_i \times U_i \rightarrow \mathcal{E}_t$, for $t = 0, \dots, T-1$
- $h_i^T: S_i \rightarrow \mathcal{E}_T$.

C. The aggregative problem

Finally, $T+1$ coupling functions $f_t: \mathcal{E}_t \rightarrow \mathbb{R}$ are supposed to be given. The optimal control problem of interest is:

$$\begin{aligned} \inf_{x \in \mathcal{X}} J(x) := & \frac{1}{N} \sum_{i=1}^N \left(\sum_{t=0}^{T-1} \ell_i^t(s_i^t, u_i^t) + \ell_i^T(s_i^T) \right) \\ & + \sum_{t=0}^{T-1} f_t \left(\frac{1}{N} \sum_{i=1}^N h_i^t(s_i^t, u_i^t) \right) + f_T \left(\frac{1}{N} \sum_{i=1}^N h_i^T(s_i^T) \right). \end{aligned} \quad (5)$$

This problem is a particular instance of problem (P). To see this, define f_{T+1} as the identity function on $\mathcal{E}_{T+1} := \mathbb{R}$. For $t = 0, \dots, T+1$, define $g_{it}(x_i)$ as follows:

- if $t < T$, $g_{it}(x_i) = h_i^t(s_i^t, u_i^t)$
- if $t = T$, $g_{it}(x_i) = h_i^T(s_i^T)$
- if $t = T+1$, $g_{it}(x_i) = \left(\sum_{\tau=0}^{T-1} \ell_i^\tau(s_i^\tau, u_i^\tau) \right) + \ell_i^T(s_i^T)$.

Then we have

$$J(x) = \sum_{t=0}^{T+1} f_t \left(\frac{1}{N} \sum_{i=1}^N g_{it}(s_i) \right).$$

As before, we denote $g_i(x_i) = (g_{it}(x_i))_{t=0,\dots,T+1}$, $\mathcal{E} = \prod_{t=0}^{T+1} \mathcal{E}_t$ and for $y \in \mathcal{E}$, $f(y) = \sum_{t=0}^{T+1} f_t(y_t)$. For any $i = 1, \dots, N$ and for any $t = 0, \dots, T+1$, we denote

$$Y_{it} = \{g_{it}(x_i) \mid x_i \in \mathcal{X}_i\} \quad \text{and} \quad Y_t = \frac{1}{N} \sum_{i=1}^N Y_{it}.$$

Assumption 2. *For $i = 1, 2, \dots, N$ and for $t = 0, 1, \dots, T+1$,*

- f_t is L_t -Lipschitz on $\text{conv}(Y_t)$,
- f_t is continuously differentiable on a neighborhood of $\text{conv}(Y_t)$, ∇f_t is \tilde{L}_t -Lipschitz on $\text{conv}(Y_t)$
- f_t is convex on $\text{conv}(Y_t)$.

Assumptions 1 and 2 imply Assumptions A and B for problem (5). Assumption C is trivially satisfied since \mathcal{X}_i is a finite set.

Remark 5. Let us note that from a theoretic point of view, problem (5) could be addressed by dynamic programming. This would allow the computation of an exact solution. However, this would require to compute a value function of the form $V^t(s^t)$, where $s^t = (s_1^t, \dots, s_N^t) \in \prod_{i=1}^N S_i^t$. The resulting complexity, of order $\sum_{t=0}^T \prod_{i=1}^N \text{card}(S_i^t)$, is prohibitive even for moderate values of N . In contrast, the complexity of each iteration of the stochastic Frank-Wolfe algorithm is linear with respect to N , while the accuracy of the algorithm improves as N increases.

D. Resolution of the sub-problems

We explain now how to solve the sub-problems (1) associated with the aggregative optimal control problem (5). Let $y \in \mathcal{E}$. Let $\mu \in \mathcal{E}$ be defined by $\mu^t = \nabla f_t(y^t)$. By definition

of f_{T+1} , $\mu^{T+1} = 1$. The sub-problem (1) reads:

$$\inf_{x_i \in \mathcal{X}_i} \left(\sum_{t=0}^{T-1} \ell_i^t[\mu^t](s_i^t, u_i^t) \right) + \ell_i^T[\mu^T](s_i^T), \quad (6)$$

$$\text{where: } \begin{cases} \ell_i^t[\mu^t](s_i^t, u_i^t) = \ell_i^t(s_i^t, u_i^t) + \langle \mu^t, h_i^t(s_i^t, u_i^t) \rangle, \\ \ell_i^T[\mu^T](s_i^T) = \ell_i^T(s_i^T, u_i^T) + \langle \mu^T, h_i^T(s_i^T) \rangle. \end{cases}$$

The sub-problem (6) can be solved by dynamic programming. Algorithm 3 yields a solution to (6). It consists of two steps: first in a backward pass, a sequence of value functions $(V_i^t)_{t=0, \dots, T+1}$ is computed, where $V_i^t: S_i^t \rightarrow \mathbb{R}$. A globally optimal solution is obtained in a forward pass. Note that the value of the optimization problem (7) is finite as a consequence of Assumption 1.

Algorithm 3: Dynamic programming algorithm

Step 1: Backward pass.

```

for  $s_i^T \in S_i^T$  do
  | Set  $V_i^T(s_i^T) = \ell_i^T[\mu^T](s_i^T)$ ;
end
for  $t = T - 1, T - 2, \dots, 0$  do
  | for  $s_i^t \in S_i^t$  do
    | Define  $V_i^t(s_i^t)$  as the value of the problem
      |
      | 
$$\min_{u_i^t} \ell_i^t[\mu^t](s_i^t, u_i^t) + V_i^{t+1}(\pi_i^t(s_i^t, u_i^t)),$$

      | 
$$\text{s.t.: } \begin{cases} \pi_i^t(s_i^t, u_i^t) \in S_i^{t+1}, \\ u_i^t \in U_i^t(s_i^t). \end{cases} \quad (7)$$

      |
    | end
  | end
end

```

Step 2: Forward pass.

```

Find  $\bar{s}_i^0 \in \operatorname{argmin}_{s_i^0 \in S_i^0} V_i^0(s_i^0)$ ;
for  $t = 0, \dots, T - 1$  do
  | Find a solution  $\bar{u}_i^t$  to the problem
    |
    | 
$$\min_{u_i^t} \ell_i^t[\mu^t](\bar{s}_i^t, u_i^t) + V_i^{t+1}(\pi_i^t(\bar{s}_i^t, u_i^t)),$$

    | 
$$\text{s.t.: } \begin{cases} \pi_i^t(\bar{s}_i^t, u_i^t) \in S_i^{t+1}, \\ u_i^t \in U_i^t(\bar{s}_i^t). \end{cases}$$

    |
  | Set  $\bar{s}_i^{t+1} = \pi_i^t(\bar{s}_i^t, \bar{u}_i^t)$ .
end

```

Remark 6. Algorithm 2 can be applied to a more general class of aggregative optimal control problems. For $i = 1, \dots, N$ and $x_i \in \mathcal{X}_i$, denote

$$h_i(x_i) = \left(h_i^0(s_i^0, u_i^0), \dots, h_i^{T-1}(s_i^{T-1}, u_i^{T-1}), h_i^T(s_i^T) \right).$$

Next consider the following generalization of (5):

$$\inf_{x \in \mathcal{X}} \frac{1}{N} \sum_{i=1}^N \left(\sum_{t=0}^{T-1} \ell_i^t(s_i^t, u_i^t) + \ell_i^T(s_i^T) \right) + f \left(\frac{1}{N} \sum_{i=1}^N h_i(x_i) \right).$$

Note that it is not even possible to formulate a dynamic programming principle in that case, however, the associated

sub-problems are still of the form (6) and can be solved with Algorithm 3.

VI. APPLICATION EXAMPLE

Let us now turn to the problem of the charging of a fleet of batteries. We propose a very simple model which is essentially illustrative, rather than realistic. However, it is emphasised that the proposed approach can easily incorporate more realistic constraints on battery operation (e.g. taking into account limits on cycles numbers). Indeed, these refinements remain localized at the sub-problem level (impacting only the dynamic programming Algorithm 3). They consist either in adding a state variable or in modifying the local costs in order to penalise undesired behaviour. Suppose that there are N batteries to be charged. Let s_i^t be the state of charge (SoC) for the battery i at the time t .

A. Dynamics of the batteries

The dynamics of each battery is characterized by three parameters: an initial state of charge $s_i^{\text{in}} \in \mathbb{N}$, a maximal state of charge $s_i^{\text{max}} \in \mathbb{N}$, a maximal load speed $u_i^{\text{max}} \in \mathbb{N}$. We define:

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

In words: the initial condition s_i^{in} is given, the charging of the battery is additive, the charging speed is bounded by u_i^{max} and is such that s_i^t can never exceed s_i^{max} .

B. Cost and contribution functions

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

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

The contribution functions are defined by $h_i^T(s_i^T) = 0$ 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 $J(\theta)$ writes

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

The cost function has two contributions, one depends on the average of charging levels of all the batteries, the other one depends on the individual final SoC of each battery. To be more precise, for $t \leq T - 1$, the average charging level needs to approach some target power c_t . For $t = T$, the batteries expect to approach their maximum SoCs.

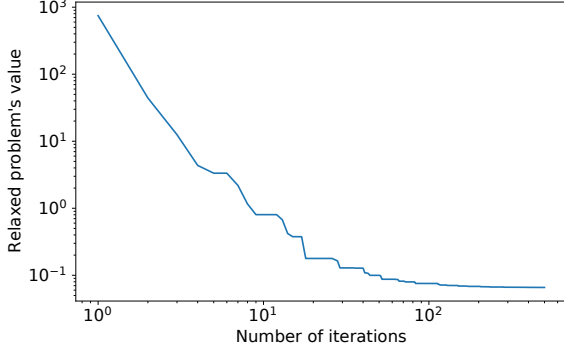


Fig. 1. Frank-Wolfe Algorithm with 500 iterations for the relaxed problem.

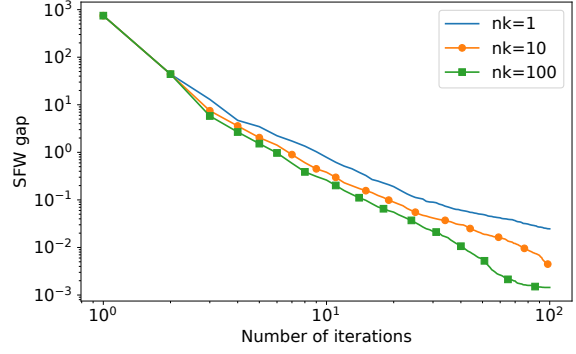


Fig. 2. Stochastic Frank-Wolfe Algorithm with 100 iterations, expectation of the gap.

C. Numerical simulations

The parameters are chosen as follows:

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

Thus, for $t = 0, 1, \dots, 23$, the diameter of the range set Y_{it} is less than $u_i^{\text{max}} = 4$, and the Lipschitz constant \bar{L}_t is $2\alpha^t$, which is less than 4. Then, we have the following upper bound for the relaxation gap $C_1/2N$:

$$\frac{C_1}{2N} \leq \frac{1}{200} * \frac{1}{100} * \sum_{t=0}^{23} \left(4 * \sum_{i=1}^{100} 4^2 \right) = 7.68.$$

Fig. 1 shows the outcome of Algorithm 1 with 500 iterations to get an approximation of the minimum \mathcal{J}^* of the relaxed problem. The curve represents the relaxed cost. Fig. 2 shows the outcome of Algorithm 2, for different choices of n_k with 100 iterations. Since the algorithm is stochastic, we ran it 50 times independently to evaluate its efficiency; the curves represent the average value of $\gamma_k = J(x^k) - \mathcal{J}^*$. The standard deviation is displayed on Fig. 3. In all cases, an average value of the gap significantly smaller than 7.68 can be reached; the standard deviation is also significantly smaller than 7.68 at the last iterations. We have initialized the algorithm with values of x_i^0 such that $u_i^t = 0$, for any $t = 0, \dots, T - 1$.

REFERENCES

- [1] I. Atzeni, L.G. Ordóñez, G. Scutari, D.P. Palomar, and J.R. Fonollosa. Noncooperative and cooperative optimization of distributed energy generation and storage in the demand-side of the smart grid. *IEEE transactions on signal processing*, 61(10):2454–2472, 2013.
- [2] J.P. Aubin and I. Ekeland. Estimates of the duality gap in nonconvex optimization. *Maths. of Operations Research*, 1(3):225–245, 1976.
- [3] K. Barty, P. Carpentier, and P. Girardeau. Decomposition of large-scale stochastic optimal control problems. *RAIRO-Operations Research*, 44(3):167–183, 2010.

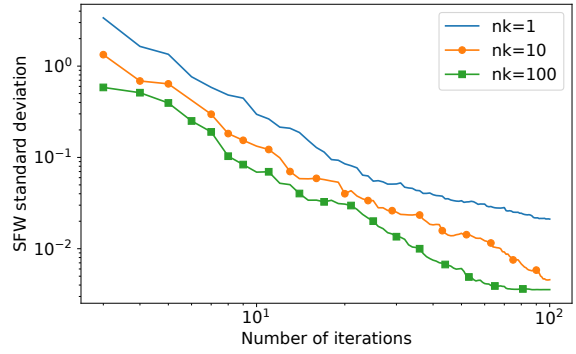


Fig. 3. Stochastic Frank-Wolfe Algorithm with 100 iterations, standard deviation of the gap.

- [4] J.F. Bonnans, K. Liu, N. Oudjane, L. Pfeiffer, and C. Wan. Large-scale nonconvex optimization: randomization, gap estimation, and numerical resolution. *ArXiv preprint*, 2022.
- [5] J.C. Dunn and S. Harshbarger. Conditional gradient algorithms with open loop step size rules. *Journal of Mathematical Analysis and Applications*, 62(2):432–444, 1978.
- [6] R.L. Raffard, C.J. Tomlin, and S.P. Boyd. Distributed optimization for cooperative agents: Application to formation flight. In *2004 43rd IEEE Conference on Decision and Control*, volume 3, pages 2453–2459. IEEE, 2004.
- [7] A. Seguret, Clémence Alasseur, J Frédéric Bonnans, Antonio De Paola, Nadia Oudjane, and Vincenzo Trovato. Decomposition of high dimensional aggregative stochastic control problems. *arXiv preprint arXiv:2008.09827*, 2020.
- [8] A. Séguret, C. Wan, and C. Alasseur. A mean field control approach for smart charging with aggregate power demand constraints. In *2021 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, pages 01–05. IEEE, 2021.
- [9] R.M. Starr. Quasi-equilibria in markets with non-convex preferences. *Econometrica: journal of the Econometric Society*, pages 25–38, 1969.
- [10] M. Wang. Vanishing price of decentralization in large coordinative nonconvex optimization. *SIAM J. Optim.*, 27(3):1977–2009, 2017.