A Privacy-preserving Disaggregation Algorithm for Non-intrusive Management of Flexible Energy
Paulin Jacquot, Olivier Beaude, Pascal Benchimol, Stéphane Gaubert, Nadia Oudjane

To cite this version:
Paulin Jacquot, Olivier Beaude, Pascal Benchimol, Stéphane Gaubert, Nadia Oudjane. A Privacy-preserving Disaggregation Algorithm for Non-intrusive Management of Flexible Energy. CDC 2019 - 58th IEEE Conference on Decision and Control, Dec 2019, Nice, France. hal-02150209

HAL Id: hal-02150209
https://hal.archives-ouvertes.fr/hal-02150209
Submitted on 7 Jun 2019

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.
A Privacy-preserving Disaggregation Algorithm for Non-intrusive Management of Flexible Energy

Paulin Jacquot, Olivier Beaude, Pascal Benchimol, Stéphane Gaubert, Nadia Oudjane

Abstract—We consider a resource allocation problem involving a large number of agents with individual constraints subject to privacy, and a central operator whose objective is to optimizing a global, possibly non-convex, cost while satisfying the agents’ constraints. We focus on the practical case of the management of energy consumption flexibilities by the operator of a microgrid. This paper provides a privacy-preserving algorithm that does compute the optimal allocation of resources, avoiding each agent to reveal her private information (constraints and individual solution profile) neither to the central operator nor to a third party. Our method relies on an aggregation procedure: we maintain a global allocation of resources, and gradually disaggregate this allocation to enforce the satisfaction of private constraints, by a protocol involving the generation of polyhedral cuts and secure multiparty computations (SMC). To obtain these cuts, we use an alternate projections method à la Von Neumann, which is implemented locally by each agent, preserving her privacy needs. Our theoretical and numerical results show that the method scales well as the number of agents gets large, and thus can be used to solve the allocation problem in high dimension, while addressing privacy issues.

I. INTRODUCTION

Motivation. Consider an operator of an electricity microgrid optimizing the joint production schedules of renewable and thermal power plants in order to satisfy, at each time period, the consumption constraints of its consumers. To optimize the costs and the renewables integration, this operator relies on demand response techniques, that is, taking advantage of the flexibilities of some of the consumers electric appliances—those which can be controlled without impacting the consumer’s comfort, as electric vehicles or water heaters [1]. However, for privacy reasons, consumers are not willing to provide neither their consumption constraints nor their consumption profiles to a central operator or any third party, as this information could be used to induce private information such as their presence at home.

The global problem of the operator is to find an allocation of power (aggregate consumption) \( p = (p_t) \), at each time period (resource) \( t \in T \), such that \( p \in P \) (feasibility constraints of power allocation, induced by the power plants constraints). Besides, this aggregate allocation has to match an individual consumption profile \( x_n = (x_{n,t}) \in \mathcal{X} \) for each of the consumer (agent) \( n \in \mathcal{N} \) considered. The problem can be written as follows:

\[
\begin{align}
\min_{x \in \mathbb{R}^{N \times T}, \ p \in P} & \quad f(p) \tag{1a} \\
x_n \in \mathcal{X}_n, \ \forall n \in \mathcal{N} & \tag{1b} \\
\sum_{n \in \mathcal{N}} x_{n,t} = p_t, \ \forall t \in T & \tag{1c}
\end{align}
\]

The (aggregate) allocation \( p \) can be made public, that is, revealed to all agents. However, the individual constraint set \( \mathcal{X}_n \) and individual profiles \( x_n \) constitute private information of agent \( n \), and should not be revealed to the operator or any third party.

It will be helpful to think of Problem (1) as the combination of two interdependent subproblems:

i) given an aggregate allocation \( p \), the disaggregation problem consists in finding, for each agent \( n \), an individual profile \( x_n \) satisfying her individual constraint (1b), so that constraint (1c) is satisfied; when this is possible, we say that a disaggregation exists for \( p \);

ii) Each subset \( Q \subset P \) determines an optimal resource allocation problem, or master problem, \( \min_{p \in Q} f(p) \).

When \( Q \) is precisely the set of aggregate allocations for which a disaggregation exists, the optimal solutions of the master problem correspond to the optimal solutions of (1).

Aside from the example above, resource allocation problems (optimizing common resources shared by multiple agents) find many applications in energy [1, 2], logistics [3], distributed computing [4], health care [5] and telecommunications [6]. In these applications, several entities or agents (e.g. consumers, stores, tasks) share a common resource (energy, products, CPU time, broadband) which has a global cost for the system. For large systems composed of multiple agents, the dimension of the overall problem can be prohibitive and one can rely on decomposition and distributed approaches [7–9] to answer to this issue. Besides, agents’ individual constraints are often subject to privacy issues [10]. These considerations have paved the way to the development of privacy-preserving, or non-intrusive methods and algorithms, e.g. [11, 12].

In this work, we consider that each agent has a global demand constraint (e.g. energy demand or product quantity), which confers to the disaggregation problem the particular structure of a transportation polytope [13]: the sum over the agents is fixed by the aggregate solution \( p \), while the sum over the \( T \) resources are fixed by the agent global demand constraint. Besides, individual constraints can also include minimal and maximal levels on each resource, as for instance...
electricity consumers require, through their appliances, a minimal and maximal power at each time period.

**Main Results.** The main contribution of the paper is to provide a non-intrusive and distributed algorithm (Algorithm 4) that computes an aggregated resource allocation \( p_* \), optimal solution of the—possibly nonconvex—optimization problem (1), along with feasible individual profiles \( x \) for agents, without revealing the individual constraints of each agent to a third party, either another agent or a central operator. The algorithm solves iteratively instances of master problems \( \min_{p \in \mathcal{P}(s)} f(p) \) by constructing successive approximations \( \mathcal{P}(s) \subset \mathcal{P} \) of the aggregate feasible set of (1) for which a disaggregation exists, by adding a new constraint on \( p \) to \( \mathcal{P}(s) \), before solving the next master problem.

To identify whether or not disaggregation is feasible and to add a new constraint in the latter case, our algorithm relies on the alternating projections method (APM) [14, 15] for finding a point in the intersection of convex sets. Here, we consider the two following sets: on the one hand, the affine space defined by the aggregation to a given resource profile, and on the other hand, the set defined by all agents individual constraints (demands and bounds). As the latter is defined as a Cartesian product of each agent’s feasibility set, APM can operate in a distributed fashion. The sequence constructed by the APM converges to a single point if the intersection of the convex sets is nonempty, and it converges to a periodic orbit of length 2 otherwise. Our key result is the following: if the APM converges to a periodic orbit, meaning that the disaggregation is not feasible, we construct from this orbit a polyhedral cut, i.e. a linear inequality satisfied by all feasible solutions \( p \) of the global problem (1), but violated from the current resource allocation (Thm. 4). Adding this cut to the master problem, we can recompute a new resource allocation and repeat this procedure until disaggregation is possible. Another main result stated in this paper is the explicit upper bound on the convergence speed of APM in our framework (Thm. 2), which is obtained by spectral graph theory methods, exploiting also geometric properties of transportation polytopes. This explicit speed shows a linear impact of the number of agents, which is a strong argument for the applicability of the method in large distributed systems.

**Related Work.** A standard approach to solve resource allocation problems in a distributed way is to use a Lagrangian (dual) decomposition technique [8, 16, 17]. Those techniques are generally used to decompose a large problem into several subproblems of small dimension. However, Lagrangian decomposition methods are based on strong duality property, requiring global convexity hypothesis which are not satisfied in many practical problems (e.g. MILP, see Sec. V). On the contrary, our method can be used when the master allocation problem is not convex. In [2], the authors study a disaggregation problem similar to the one considered in this paper. Their results concern zonotopic sets, which is different from the structure we described in Sec. II. The APM has been the subject of several works in itself [15, 18, 19]. The authors of [20] provide general results on the convergence rate of APM for semi-algebraic sets. They show that the convergence is geometric for polyhedra. However, it is generally hard to compute explicitly the geometric convergence rate of APM, as this requires to bound the singular values of certain matrices arising from the polyhedral constraints. In [21], the authors provide an explicit convergence rate for APM on a class of polyhedra arising in submodular optimization. The sets they consider differ from the present transportation polytopes.

**Structure.** In Sec. II, we describe the master resource allocation problem and formulate the associated disaggregation problem. In Sec. III, we focus on the APM and state our main results. In Sec. IV, we apply these results to describe a non-intrusive version of APM (NI-APM) that is used to describe our non-intrusive algorithm for computing an optimal resource allocation. Finally, in Sec. V, we provide a concrete numerical example based on a MILP to model the management of a local electricity system (microgrid), and study numerically the influence of the number of agents on the time needed for convergence of our algorithm.

**Notation.** Vectors and matrices are denoted by bold fonts, \( v^\top \) denotes the transpose of \( v \), \( I_K \) denotes the vector \((1 \ldots 1)^\top \) of size \( K \), \( U([a, b]) \) stands for the uniform distribution on \([a, b] \). We use \( \| x \|_F \) to denote the Frobenius norm \( \| x \|^2 = \sum_{n, t} x_{n, t}^2 \), and \( P_C(\cdot) \) to denote the Euclidean projection on a convex set \( C \).

II. MASTER PROBLEM AND DISAGGREGATION STRUCTURE

In this work, we suppose an operator wishes to determine an allocation of resources, represented by a \( T \)-dimensional vector \( p \), in order to minimize a global cost function \( f \), for instance, an electricity power economic dispatch (or the allocation of different types of merchandise in warehouses in logistics applications) subject to a set of constraints described by a feasibility set \( \mathcal{P} \). This problem can be nonconvex either because of nonconvex costs \( f \) or because of a nonconvex feasible set \( \mathcal{P} \) (see Sec. V). In the proposed method, the operator will consider master problems of the form:

\[
\min_{p \in \mathcal{P}} f(p) \quad \text{ s.t. } p \in \mathcal{P}(s),
\]

where the set \( \mathcal{P}(s) \subset \mathcal{P} \) is an aggregate approximation of disaggregation constraints. Indeed, the resource allocation \( p \) has to be shared between \( N \) agents (e.g. consumers). Each agent has a global demand (total energy needed) \( E_n \) and some lower and upper bounds on each of the resource \( t \in T \). The admissible set of profiles of agent \( n \) is therefore:

\[
\mathcal{X}_n := \{ x_n \in \mathbb{R}^T | \exists x_t \in E_n \text{ and } \forall t, x_{n, t} \leq x_{n, t} \leq \pi_{n, t} \}.
\]

The disaggregation problem consists in finding individual profiles \( x = (x_{n, t})_{n, t} \in \mathbb{R}^{NT} \) of a given aggregated allocation \( p \) such that \( x_n \) is feasible for each agent \( n \):

\[
\text{FIND } x \in \mathcal{X}_p \cap \mathcal{X}
\]

where \( \mathcal{X}_p := \{ y \in \mathbb{R}^{NT} | y^\top I_N = p \} \) and \( \mathcal{X} := \prod_{n \in N} \mathcal{X}_n \).
Theorem 1. The disaggregation problem (4) is feasible (i.e. $\mathcal{X} \cap \mathcal{Y}_p \neq \emptyset$) iff for any $\mathcal{T}_in \subset \mathcal{T}, \mathcal{N}_in \subset \mathcal{N}$:

$$\sum_{t \in \mathcal{T}_in} p_t \leq \sum_{t \in \mathcal{T}_in, n \in \mathcal{N}_in} p_{n,t} - \sum_{t \in \mathcal{T}_in, n \notin \mathcal{N}_in} p_{n,t} + \sum_{n \in \mathcal{N}_in} e_n. \quad (6)$$

The inequality (6) has a simple interpretation: the residual demand (the left hand side composed of demand and exports minus production) in $\mathcal{T}_in \cup \mathcal{N}_in$ cannot exceed the import capacity (right hand side of the inequality). One can see that, in the example of Fig. 1, inequality (6) does not hold when using the cut composed of the dashed nodes $p_1$ and $E_1$.

![Fig. 1. Example of disaggregation structure ($T = 2, N = 3$), with $\mathcal{X} = 0$ and $\mathcal{Y} := 1$. Although the aggregate constraints (5) are satisfied, the disaggregation (4) of $\mathcal{P}$ is not feasible in this example (see Thm. 1).](image1)

Algorithm 1 Alternate Projections Method (APM)

Require: Start with $y^{(0)}$, $k = 0$, $\varepsilon_{cvg}$, a norm $\| \cdot \|$ on $\mathbb{R}^{NT}$

1: repeat
2: $x^{(k+1)} \leftarrow P_X(y^{(k)})$
3: $y^{(k+1)} \leftarrow P_Y(x^{(k+1)})$
4: $k \leftarrow k + 1$
5: until $\| y^{(k)} - y^{(k-1)} \| < \varepsilon_{cvg}$

The convergence of Algorithm 1 is proved by Thm. 2:

Theorem 2 ([15]). Let $\mathcal{X}$ and $\mathcal{Y}$ be two convex sets with $\mathcal{X}$ bounded, and let $(x^{(k)})_k$ and $(y^{(k)})_k$ be two infinite sequences generated by Algorithm 1 with $\varepsilon_{cvg} = 0$. Then there exists $x^\infty \in \mathcal{X}$ and $y^\infty \in \mathcal{Y}$ such that:

$$x^{(k)} \rightarrow x^\infty, \quad y^{(k)} \rightarrow y^\infty; \quad (7a)$$

$$\| x^\infty - y^\infty \|_2^2 = \min_{x \in \mathcal{X}, y \in \mathcal{Y}} \| x - y \|_2. \quad (7b)$$

In particular, if $\mathcal{X} \cap \mathcal{Y} \neq \emptyset$, then $(x^{(k)})_k$ and $(y^{(k)})_k$ converge to a same point $x^\infty \in \mathcal{X} \cap \mathcal{Y}$.

If disaggregation is not feasible, Thm. 2 states that APM will “converge” to an orbit $(x^\infty, y^\infty)$ of period 2.

The convergence rate of APM has been the subject of several works [18, 20], and it strongly depends on the structure of the sets on which the projections are done: for instance, if the sets are polyhedral, [20, Prop. 4.2] shows that the convergence is geometric. However, there are very few cases in which an explicit upper bound on the convergence rate has been proved. In our case, we are able to obtain such a bound, as shown in the following theorem:

Theorem 3. For the sets $\mathcal{X}$ and $\mathcal{Y}$ defined in (3-4), the two subsequences of alternate projections converge at a geometric rate to $x^\infty \in \mathcal{X}$, $y^\infty \in \mathcal{Y}$, with:

$$\| x^{(k)} - x^\infty \|_2 \leq 2\| x^{(0)} - x^\infty \|_2 \times \rho_{NT}^k \quad (8)$$

where $\rho_{NT} \stackrel{\text{def}}{=} 1 - \frac{1}{4} (N(T + 1)^2(T - 1)) < 1$.

Same inequalities hold for the convergence of $y^{(k)}$ to $y^\infty$.

Proof. Appendix II provides a sketch of the proof.

Thm. 3 shows that the APM is efficient in our case of bounded transport polytopes. It shows that the number of iterations for a given accuracy grows linearly in the number of agents $N$.

As stated in (4), the set $\mathcal{X}$ is a Cartesian product $\prod_n \mathcal{X}_n$, so that the projection (13) can be computed by $N$ projections...
on \((X_n)_{n}\), which can be executed in parallel. Now, instead of solving the quadratic program by standard interior point methods and due to its particular structure, we can use the algorithm of Brucker [24], which has a complexity in \(O(T)\).

On the other hand, \(P_{\nu}(\cdot)\) is a projection on an affine space, and the solution can be obtained explicitly as:

\[
\forall n, t, y_{n,t} = x_{n,t} + \nu_t \quad \text{and} \quad \nu = \frac{1}{N}(p - x^\top \mathbf{1}_N) . \tag{8}
\]

### B. Generation of a cut from APM iterates

Our key result is the following: in the case where APM converges to a periodic orbit \((x^\infty, y^\infty)\) with \(x^\infty \neq y^\infty\) (see Thm. 2), we obtain from \((x^\infty, y^\infty)\) an inequality (6) that is violated by \(p\):

**Theorem 4.** For the sets \(\mathcal{X} \cap \mathcal{Y}\) defined in (3-4) and if \(\mathcal{X} \cap \mathcal{Y} = \emptyset\), the following sets given by the limit orbit \((x^\infty, y^\infty)\) defined in Thm. 2:

\[
T_0 \text{ def } \{t | p_t > \sum_{n \in N} x^\infty_{n,t}\}
\]
and

\[
N_0 \text{ def } \{n | E_n - \sum_{t \in T_0} \overrightarrow{x}_{n,t} - \sum_{t \in T_0} \overrightarrow{x}_{n,t} < 0\}
\]

define a Hoffman cut of form (6) violated by \(p\), that is:

\[
\sum_{n \in N_0} E_n - \sum_{t \in T_0} p_t + \sum_{t \in T_0, n \notin N_0} \overrightarrow{x}_{n,t} - \sum_{t \in T_0, n \in N_0} \overrightarrow{x}_{n,t} < 0 . \tag{10}
\]

This cut can be reformulated in terms of \(\mathbf{1}_N^\top x^\infty\) as:

\[
A_{T_0} \sum_{t \in T_0} p_t \text{ with } A_{T_0} \text{ def } \sum_{t \in T_0 \in N} x^\infty_{n,t} . \tag{11}
\]

**Proof.** Appendix I gives the sketch of the proof of Thm. 4. The complete proofs will be given elsewhere.

One can see that, intuitively, \(N_0\) is the subset associated to \(T_0\) that minimizes the right hand side of (6). Note that Thm. 4 gives an alternative constructive proof of Hoffman circulation’s theorem (Thm. 1) in the case of a bipartite graph of the form of Fig. 1. Moreover, in the case where the disaggregation problem (4) is not feasible, the negation of equation (11) provides a new valid constraint as a condition for the existence of a disaggregated profile of \(p\). This constraint can be used in the master problem (2) to update the vector of resources \(p\) for the next iteration. This constraint only involves the aggregate information \(\mathbf{1}_N^\top x^\infty\) on the users profile. To make the process fully non-intrusive, we explain in Sec. IV-A how the operator can compute this constraint without making the agents reveal their profiles \((x^\infty_{n})_{n \in N}\).

### IV. Non-intrusive Projections and Computation of Disaggregated Optimal Resources

**A. Non-Intrusive Alternate Projections Method (NI-APM)**

Because of the particular structure of the problem, the projections in APM can be computed separately by the operator and the agents. The projection \(P_{\bar{D}}\) is made by the operator, which only requires to know \(p\) and the aggregate profile \(x^\top \mathbf{1}_N\) according to (8). The projection \(P_{\bar{X}}\) on \(X = \prod_n \mathcal{X}_n\) is executed in parallel by each agent: \(n\) computes \(P_{\bar{X}}\), which only needs her private information \(E_n\) and \(\overrightarrow{x}_{n}, \overleftarrow{x}_{n}\). However, in the way APM is described in Algorithm 1, the operator and the agents still need to exchange the iterates \(x^{(k)}, y^{(k)}\) at each step. To avoid the transmission of agents’ profiles to the operator, we use a secure multiparty computation (SMC) technique (see [25]) which enables the operator to obtain the aggregate profile \(S^{(k)} := \mathbf{1}_N x^{(k)}\) in a non-intrusive manner, as described in Algorithm 2.

The main idea of SMC is that, instead of sending her profile \(x_n\), agent \(n\) splits \(x_{n,t}\) for each \(t\) into \(N\) random parts \((s_{n,t,m})_{m=1}^{N}\), according to an uniform distribution and summing to \(x_{n,t}\) (Lines 2-3). Thus, each part \(s_{n,t,m}\) taken individually does not reveal any information on \(x_n\) nor on \(X_n\), and can be sent to agent \(m\). Once all exchanges of parts are completed (Line 5), and \(n\) has herself received the parts from other agents, agent \(n\) computes a new aggregate quantity \(\sigma_n\) (Line 7), which does not contain either any information about any of the agents, and sends it to the operator (Line 8). The operator can finally compute the quantity \(S = x^\top \mathbf{1}_N = \sigma^\top \mathbf{1}_N\).

**Algorithm 2 SMC of Aggregate (SMCA) \(\sum_{n \in N} x_n\)**

**Require:** Each agent has a profile \((x_n)_{n \in N}\)

1: for each agent \(n \in N\) do
2: \(\text{Draw } \forall t, (s_{n,t,m})_{m=1}^{N} \in U([0, A])^{N-1}\)
3: and set \(\forall t, s_{n,t,N} = x_{n,t} - \sum_{m=1}^{N-1} s_{n,t,m}\)
4: Send \((s_{n,t,m})_{t} \in T\) to agent \(m \in N\)
5: done
6: for each agent \(n \in N\) do
7: Compute \(\forall t, \sigma_{n,t} = \sum_{m \in N} s_{m,t,n}\)
8: Send \((\sigma_{n,t})_{t} \in T\) to operator
9: done
10: Operator computes \(S = \sum_{n \in N} \sigma_{n}\)

**Remark 1.** As \(\sigma_n\) and \(s_{n,t}\) are random by construction, an eavesdropper aiming to learn the profile \(x_n\) of \(n\) has no choice but to intercept all the communications of \(n\) to all other agents (to learn \((s_{n,t,m})_{m \neq n}\) and \((\sigma_{n,t})_{t} \neq n\)) and to the operator (to learn \(\sigma_n\)). To increase the confidentiality of the procedure, one could use any encryption scheme (such as RSA [26]) for all communications involved in Algorithm 2.

We can use this non-intrusive computation of aggregate \(S\) in APM to obtain a non-intrusive algorithm NI-APM (Algorithm 3) in which agents do not reveal neither their profiles nor their constraints to the operator.

One can see that \(x\) and \(y\) computed in Lines 3 and 8 in Algorithm 3 correspond to the projections computed in the original APM Algorithm 1. In Algorithm 3, the operator obtains the aggregate profile \(S^{(k)}\) (Line 5), computes and sends the corrections \(\nu^{(k)}\) to all agents (Line 6). Then, each agent can compute locally the projection \(y^{(k)} = P_{\bar{D}}(x^{(k)})\) by applying the correction \(\nu^{(k)}\) (Line 8).

Using (8), we get \(\nu^{(k)} \rightarrow \nu^\infty = \frac{1}{N} (p - \mathbf{1}_N^\top x^\infty)\). Thm. 4 uses this limit value through \(T_0^\infty \text{ def } \{t | 0 < \nu^\infty_t\} \subseteq T\), which only requires to know \(\varepsilon_{cvg}\) small.
enough and a well-chosen value of $B$, we obtain $T_0 = T_0^\infty$, so that we get the termination result:

**Proposition 1.** For $B > (1 - \rho_{NT})^{-1}$, Algorithm 3 terminates in finite time.

The termination of the loop Lines 1-11 is ensured by Thm. 3. In the case where $\|x^{(k)} - y^{(k)}\| \leq \varepsilon_{\text{dis}}$, Algorithm 3 terminates. Otherwise, if $\|x^{\infty} - y^{\infty}\| > \varepsilon_{\text{dis}}$, then Algorithm 3 terminates (i.e. Line 18 is True and a new cut is found) as soon as $B\varepsilon_{\text{cvg}} < \min\left\{\frac{\|x^{\infty} - y^{\infty}\|}{2\sqrt{N}}, \frac{2}{3L}\right\}$, where $L \equiv \min\{|p_n| > 0\}$ and with $\|\cdot\| = \|\cdot\|_2$. The complete proof is omitted here.

In practice, we can start with a large $\varepsilon_{\text{cvg}}$ to obtain the first constraints while avoiding useless computation, and then half $\varepsilon_{\text{cvg}}$ if needed (Line 21) until the termination condition holds.

**Remark 2.** Lagrangian decomposition is another promising technique to develop privacy-preserving algorithms. However, Lagrangian decomposition requires convexity assumptions, whereas in the present approach, combining polyhedral cuts and alternate projection methods, the optimization problem can be nonconvex (we shall actually solve such a nonconvex example in Sec. V).

**B. Non-intrusive Disaggregation of Optimal Allocation**

In this section, we describe a method to compute a solution of the global problem (1), that is, an optimal resource allocation $p$ for which a disaggregation exists, along with an associated disaggregated profile $x_n$ for each agent $n$. This computation is done in a non-intrusive manner: the operator in charge of $p$ does not have access neither to the bounding constraints $\pi$ and $\pi$ of the agents nor to the agents disaggregated profile $x$, as detailed in Algorithm 4 below.

**Algorithm 4 Non-intrusive Optimal Disaggregation**

**Require:** $s = 0$, $\mathcal{P}(0) = \mathcal{P}$; $\text{DISAG} = \text{FALSE}$

1: while Not $\text{DISAG}$ do
2: Compute $p^{(s)} = \arg \min_{p \in \mathcal{P}(s)} p$
3: $\text{DISAG} \leftarrow \text{NI-APM}(p^{(s)})$
4: if $\text{DISAG}$ then
5: Operator adopts $p^{(s)}$
6: else
7: Obtain $\mathcal{T}_{0}^{(s)}$, $\Lambda_{T_{0}}^{(s)}$ from NI-APM($p^{(s)}$)
8: $\mathcal{P}(s+1) \leftarrow \mathcal{P}(s) \cap \{p: \sum_{t \in \mathcal{T}_{0}^{(s)}} p_t \leq \Lambda_{T_{0}}^{(s)}\}$
9: end
10: $s \leftarrow s + 1$
11: done

Algorithm 4 iteratively calls NI-APM (Algorithm 3) and in case disaggregation is not possible (Line 6), a new constraint is added (Line 8), obtained from the quantity $\Lambda_{T_{0}}$ defined in (11), to the resource problem (2). This constraint is an inequality on $p$ and thus does not reveal significant individual information to the operator. The algorithm stops when disaggregation is possible (Line 4). The termination of Algorithm 4 is ensured by the following property and the form of the constraints added (10):

**Proposition 2.** Algorithm 4 stops after a finite number of iterations, as at most $2^T$ constraints (Line 8) can be added to the master problem (Line 2).

Although there exist some instances with an exponential number of independent constraints, this does not jeopardize the proposed method: in practice, the algorithm stops after a very small number of constraints added (see the example of Sec. V). Intuitively, we will only add constraints “supporting” the optimal allocation $p$.

Thus, Algorithm 4 is a method which enables the operator to compute a resource allocation $p$ and the $N$ agents to adopt profiles $(x_n)_n$, such that $(x,p)$ solves the global problem (1), and the method ensures that both:

1) the information relative to each agent constraints (upper bounds $\pi_n$, lower bounds $\pi_n$, demand $E_n$);
2) the final disaggregated profile $x_n$ (as well as the iterates $(x_{(k)})_k$ and $(y_{(k)})_k$ in NI-APM)

are kept confidential by agent $n$ and can not be induced by a third party (either the operator or any other agent $m \neq n$).

**V. Application to Management of a Microgrid**

We apply the proposed method to solve a nonconvex distributed problem in the energy field. We consider a microgrid [27] composed of $N$ electricity consumers with flexible appliances (such as electric vehicles or water heaters), a photovoltaic (PV) power plant and a conventional generator.
A. Mixed Integer Problem Formulation

The operator responsible of the microgrid aims at satisfying the demand constraints of consumers over a set of time periods $T = \{1, \ldots, T\}$, while minimizing the energy cost for the community. We have the following characteristics:

- the PV plant generates a nondispachable power profile $(p^p_t)_{t \in T}$ at marginal cost zero;
- the conventional generator has a starting cost $C^s$, minimal and maximal power production $p^g, \bar{p}^g$, and piecewise-linear and continuous generation cost function $p^g \mapsto f(p^g)$:

$$f(p^g) = \alpha_t + c_t p^g,$$

where $\alpha_0 \equiv 0$ and $\theta_K \equiv \bar{p}^g$;
- each agent $n \in \mathcal{N}$ has some flexible appliances which require a global energy demand $E_n$, on $T$, and has consumption constraints on the total household consumption, on each time period $t \in T$, that are formulated with $\underline{x}_n, \bar{x}_n$. These parameters are confidential because they could for instance contain some information on agent $n$ habits.

The master problem (2) can be written as a MILP (12):

$$\min_{p, p^p, (p^g_t), (b^c_{k,t}), \mathbf{b}^{\text{ON}}, \mathbf{b}^{\text{ST}}} \sum_{t \in T} \left( \alpha_1 b^c_{t} + \sum_{k} c_k p^g_{k,t} + C^s b^s_{t} \right)$$

subject to:

$$p^g_{t} = \sum_{k=1}^{K} p^g_{k,t}, \quad \forall t \in T$$

(12b)

$$b^c_{t,1} \leq \bar{b}^c_{t,1}, \quad \forall t \in T$$

(12c)

$$b^c_{t,2} - \theta_1 (2 - \theta_2) \leq \bar{b}^c_{t,2}, \quad \forall t \in T$$

(12d)

$$0 \leq \bar{b}^c_{k,t} \leq b^c_{K-1,t}, \quad \forall t \in T$$

(12f)

$$b^s_{t} \geq b^{c^\text{ON}}_{t} - \bar{b}^{c^\text{ON}}_{t-1}, \quad \forall t \in \{2, \ldots, T\}$$

(12g)

$$\sum_{t} p^g_{k,t} \leq p^g_{t} \leq \bar{p}^g b^c_{t}, \quad \forall t \in T$$

(12h)

$$b^{c^\text{ON}}_{t}, b^{s^\text{ON}}, b^c_{t,1}, \ldots, b^c_{K-1,t} \in \{0, 1\}, \quad \forall t \in T$$

(12i)

$$\mathbf{p} \leq \mathbf{p}^\text{PV} + \mathbf{p}^g$$

(12j)

$$\mathbf{p}^T \mathbf{1}_N = \mathbf{E}^T \mathbf{1}_N$$

(12k)

$$\mathbf{x} \leq \mathbf{1}_N \leq \mathbf{L} \mathbf{1}_N.$$

In this formulation (12b-12f) are a mixed integer formulation of the generation cost function $f$: one can show that the Boolean variable $b^c_{k,t}$ is equal to one if $p^g_{k,t} \geq \theta_k$ for each $k \in \{1, \ldots, K-1\}$. Note that only $\alpha_1$ appears in (12a) because of the continuity assumption on $f$.

Constraints (12g-12h) ensure the on/off and starting constraints of the power plant. (12j) ensures that the power allocated to consumption is not above the total production, and (12k-12l) are the aggregated feasibility conditions already referred to in (5). Note that more complex and realistic MILP models exist for power plants (e.g. [28]), but with the same structure than (12). The nonconvexity of (12) comes from the existence of starting costs and on minimal the power constraint, which makes necessary to use Boolean state variables $b^s, b^{c^\text{ON}}$.

B. Parameters

We simulate the problem described above for different values of $N \in \{2^4, 2^5, 2^6, 2^7, 2^8\}$ and one hundred instances with random parameters for each value of $N$. A scaling factor $\kappa_N \equiv N/20$ is applied on parameters to ensure that production capacity is large enough to meet consumers demand. The parameters are chosen as follows:

- $T = 24$ (hours of a day);
- production costs: $K = 3$, $\theta = [0, 70, 100, 300]$, $C^s = [0.2, 0.4, 0.5]$, $p^g = 50 \kappa_N$, $\bar{p}^g = 300 \kappa_N$, $\alpha_1 = 4$ and $C^s = 15$;
- photovoltaic: $p^p_{t} = \left[50(1-\cos\left((t-6)/(2\pi)\right)) + \mathcal{U}(0, 10)\right] \kappa_N$

for $t \in \{6, \ldots, 20\}$, $p^p_{t} = 0$ otherwise (see Fig. 3);
- for consumption parameters, we used $\underline{x}_n, \bar{x}_n \sim \mathcal{U}(0, 10)$, $\mathcal{U}(0, 5) + \mathcal{U}(0, 5)$ and $E_n \sim \mathcal{U}(\mathbf{x}^T_1, \mathbf{x}^T_2)$, so that individual feasibility ($\mathcal{X}_n \neq \emptyset$) is ensured.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$2^4$</th>
<th>$2^5$</th>
<th>$2^6$</th>
<th>$2^7$</th>
<th>$2^8$</th>
</tr>
</thead>
<tbody>
<tr>
<td># master pb.</td>
<td>193.6</td>
<td>194.1</td>
<td>225.5</td>
<td>210.9</td>
<td>194.0</td>
</tr>
<tr>
<td># proj.</td>
<td>9506.9</td>
<td>15366.7</td>
<td>24319.3</td>
<td>26537.5</td>
<td>26646.4</td>
</tr>
</tbody>
</table>

TABLE I

NUMBER OF SUBPROBLEMS SOLVED (AVERAGE ON 100 INSTANCES)

![Fig. 2](image-url)

Total number of computed projections for different values of $N$. We observe that the number of agents $N$ has a sublinear impact on the total number of projections needed.

C. A limited impact of the number of agents

We implement Algorithm 4 using Python 3.5. The MILP (12) is solved using Cplex Studio 12.6 and Pyomo interface. Simulations are run on a single core of a cluster at 3GHz. For the convergence criteria (see Lines 11 and 12 of Algorithm 3), we use $\varepsilon_{\text{dis}} = 0.01$ with the operator norm defined by $\|\mathbf{x}\| = \max_{n \in \mathcal{N}} \sum_{t} |x_{n,t}|$ (to avoid the $\sqrt{N}$ factor in the convergence criteria appearing with $\|\cdot\|_2$), and starts with $\varepsilon_{\text{avg}} = 0.1$. The largest instances took around 10 minutes to be solved in this configuration and without parallel implementation. As the CPU time needed depend on the cluster load, it is not a reliable indicator of the influence on $N$ on the complexity of the problems. Moreover, one advantage of the proposed method is that the projections in APM can be computed locally by each agent in parallel, which could not be implemented here for practical reasons.

Tab. I gives two robust indicators of the influence of $N$ on the problem complexity: the number of master problems solved and the total number of projections computed, on average over the hundred instance for each value of $N$:

- one observes that the number of master problems solved (MILP (12)), which corresponds to the number of constraints
or “cuts” added to the master problem, remains almost constant when \( N \) increases;
- in all instances, this number is below the upper bound of \( 2^{24} > 1,6 \times 10^7 \) possible constraints (see proof of Prop. 2), which suggests that only a polynomial number of constraints are added in practice;
- the average total number of projections computed for each instance (total number of iterations of the while loop of Algorithm 3, Line 1 over all calls of APM in the instance) increases in a sublinear way, as illustrated in Fig. 2, which is even better that one could expect from the upper bound given in Thm. 3.

VI. CONCLUSION

We provided a non-intrusive algorithm that enables to compute an optimal resource allocation, solution of a possibly nonconvex–optimization problem, and affect to each agent an individual profile satisfying a global demand and lower and upper bounds constraints. Our method uses local projections and works in a distributed fashion. Hence, it ensures that the problem is not affected by the high dimension relative to the large number of agents, and that it is privacy-preserving, as agents do not need to reveal any information on their constraints or their individual profile to a third party. Several extensions and generalizations can be considered. First, we could generalize the abstract circulation problem on their constraints or their individual profile to a third party. Second, our method relies on the particular structure obtained from the form of constraints. Although these kind of constraints are widely used to model many practical situations, it would also be useful to obtain similar results for arbitrary (or polyhedral) agents constraints. Last, a deeper complexity analysis, with a thinner upper bound on the maximal number of constraints (cuts) added in the algorithm (see Prop. 2 and Tab. 1) would constitute interesting results.

REFERENCES

APPENDIX I
PROOF OF PROP. 3
To show Thm. 4, we formulate the projections \( P_N \) and \( P_Y \) as the solutions of the constrained quadratic programs:

\[
\min_{x \in \mathbb{R}^{NT}} \frac{1}{2} \| x - y \|_2^2 \quad (13a)
\]
\[x_{EF} = E \quad (\lambda) \quad (13b)
\]
\[x \leq x \leq x \quad (\mu, \overline{\mu}) \quad (13c)
\]
and:

\[
\min_{y \in \mathbb{R}^{NT}} \frac{1}{2} \| y - x \|_2^2 \quad (14a)
\]
\[y^\top 1_N = p \quad (\nu) \quad (14b)
\]
where \( \lambda, \mu, \overline{\mu}, \nu \) are the Lagrangian multipliers associated to the constraints. Although there is no such explicit characterization of the solution of (13) as the one (8) given for (14), we can obtain the following properties:

**Proposition 3.** Suppose that \( \mathcal{X} \cap \mathcal{Y} = \emptyset \) and consider the sets \( T_0 \) and \( N_0 \) given by (9). Then we have the following:

(i) \( \forall t \in T_0, \forall n \in N_0, \ y_n^{\infty} \geq \pi_{n,t} \) and \( x_{n,t}^{\infty} = \pi_{n,t} \);

(ii) \( \forall n \in N_0, \lambda_n < 0 \);

(iii) \( \forall t \notin T_0, \forall n \in N_0, \ x_{n,t}^{\infty} = \emptyset \);

(iv) the sets \( T_0, T_0^c, N_0 \) and \( N_0^c \) are nonempty.

The proof of Prop. 3 relies on the KKT optimality conditions associated to (13) and (14). Then we use Prop. 3 and the convergence condition to prove Thm. 4:

\[
\sum_{n \in N_0} E_n + \sum_{t \in T_0, n \notin N_0} x_{n,t} - \sum_{t \notin T_0} x_{n,t} - \sum_{t \in T_0} P_t \leq 0
\]

\[
= \sum_{t \in T_0} \left( \sum_{n \in N^c} x_{n,t} + \sum_{n \in N} y_{n,t} \right) = \sum_{t \in T_0} \left( \sum_{n \in N} -x_{n,t} \right)
\]

\[
= \sum_{t \in T_0} \left( - \sum_{n \in N} \| x_{n,t} - y_{n,t} \|_1 \right) = -\frac{\| x - y \|_1^2}{2} < 0,
\]

where the last equality comes from \( \sum_{t \in T_0} \sum_{n \in N} (x_{n,t} - x_{n,t}) = 0 \). The compact form (11) also follow from Prop. 3:

\[
A_{T_0}(x) = \sum_{n \in N_0} E_n + \sum_{t \in T_0, n \notin N_0} x_{n,t} - \sum_{t \notin T_0} x_{n,t} = \sum_{n \in N_0, t \in T_0} x_{n,t} + \sum_{t \in T_0, n \notin N_0} x_{n,t} = \sum_{t \in T_0} \sum_{n \in N} x_{n,t}.
\]

APPENDIX II
PROOF OF THM. 2
For this analysis, we use the space \( \mathbb{R}^{NT} = \mathbb{R}^T \times \cdots \times \mathbb{R}^T \), where the \((n-1)T+1\) to \(nT\) coordinates correspond to agent \( n \), for \( 1 \leq n \leq N \). We make use of the following results:

**Lemma 1** ([21]). For APM on polyhedra \( \mathcal{X} \) and \( \mathcal{Y} \), the sequences \( (x^{(k)}_k \) and \( (y^{(k)}_k \) converge at a geometric rate, where the rate is bounded by the maximal value of the square of the cosine of the Friedrichs angle \( c_F(U, V) \) between a face \( U \) of \( \mathcal{X} \) and a face \( V \) of \( \mathcal{Y} \), where \( c_F(U, V) \) is given by:

\[
c_F(U, V) = \sup \{ u^\top v \mid \| u \| \leq 1, \| v \| \leq 1 \}
\]

\[
u \in U \cap (U \cap V)^{\perp}, v \in V \cap (U \cap V)^{\perp} \}.
\]

**Lemma 2** ([21]). Let \( A \) and \( B \) be matrices with orthonormal rows and with equal numbers of columns and \( \Lambda_{sv}(AB^T) \) the set of singular values of \( AB^T \). Then if \( \Lambda_{sv}(AB^T) = \{1\} \), then \( c_F(\ker(A), \ker(B)) = 0 \). Otherwise, \( c_F(\ker(A), \ker(B)) = \max_{\lambda \in \Lambda_{sv}(AB^T)} \{ \lambda \} \).

In our case, the polyhedra \( \mathcal{Y} \) is an affine subspace \( \mathcal{Y} = \{ x \in \mathbb{R}^{NT} \mid Ax = \sqrt{N-1}1_T \} \) with \( A \defeq \sqrt{N-1}J_{1,N} \otimes I_T \), where \( \otimes \) denotes the Kronecker product. The matrix \( A \) has orthonormal rows and the direction of \( \mathcal{Y} \) is \( \ker(A) \).

Describing the faces \( \mathcal{X} \) is more complex. We have a polyhedral description of \( \mathcal{X} \), and the faces of \( \mathcal{X} \) are subsets of the collection of affine subspaces indexed by \( (T_n, T_0) \subset \mathcal{T}^N \) (with \( T \cap T = \emptyset \)):

\[
A_{T_n, T_0}(x) = \left\{ (x_{n,t} \mid \forall n, x_{n,t} \in x_{n,t} \right\}.
\]

The associated linear subspace is given by \( \ker(B) \), where the \( N \) first rows of \( B \) are given by \( B[n] \defeq \square(T_{1,n}) \otimes I_T \), and the matrix \( B \) has \( b = \sum_n |\operatorname{card}(T_n)\) more rows, where \( T_n \defeq \sum_{t \in T} 1_{t \in T_n} \), corresponding to the saturated inequalities \( x_{n,t} = \pi_{n,t} \) or \( \emptyset \).

We denote by \( K_n \defeq |\operatorname{card}(T_n)\). Then, renormalizing \( B \), we can show that the double product \( S := (AB^T)(A^T)B \), of size \( T \times T \) is given by:

\[
S = \frac{1}{N} \left( \sum_{n} \sum_{(k, \ell) \in \mathcal{T}_n} \frac{1}{T-K_n} \right) + \frac{1}{N} \sum_{1 \leq t \leq T} \sum_{t \in \mathcal{T}_n} 1_{t \in \mathcal{T}_n} E_{t,t}.
\]

Denote \( \mathcal{T} \defeq \cup_{n} \mathcal{T}_n^c \) and \( P \defeq I_T - S \). As \( P \) can be written as a block diagonal matrix \( P = \operatorname{diag}(P_{T}, 0_{T-r}) \), we can restrict ourselves to the subspace \( \operatorname{Vect}(e_t)_{t \in \mathcal{T}} \) to find the least positive eigenvalue of \( P \), that we denote by \( \lambda_1 \).

Consider the weighted graph \( \mathcal{G} = (\mathcal{T}, \mathcal{E}) \) whose vertices are the time periods \( \mathcal{T} \) and edge \((k, \ell)\) has weight \( S_{k, \ell} = \sum_{l \in E} 1_{l \in \mathcal{E}(k, \ell)} \) (if this quantity is zero, then there is no edge between \( k \) and \( \ell \)). One can show that \( \sum_{(k, \ell) \in \mathcal{T}_n} P_{k, \ell} \), which shows that \( P \) is the Laplacian matrix of \( \mathcal{G} \).

Using the Laplacian property and Cauchy-Schwarz, one shows that for any \( u \perp 1 \):

\[
u^\top Pu \geq \min_{k, \ell \in \mathcal{T}_n} (P_{k, \ell}) (u_{s, \ast} - u_{s, \ast}^\top d_{s, \ast})^2 \geq \frac{4T \| u \|^2}{(T+1)^2}.
\]

where \( u_{s, \ast} := \max\{u_1, u_2, \ldots, u_s\} \) and \( d_{s, \ast} := d_1 + d_2 + \ldots + d_s \).