SIAM J. Optim.
Vol. 30, No. 3, pp. 2303–2336

# A PRIVACY-PRESERVING METHOD TO OPTIMIZE DISTRIBUTED RESOURCE ALLOCATION[*]

OLIVIER BEAUDE[†], PASCAL BENCHIMOL[‡], STÉPHANE GAUBERT[§], PAULIN JACQUOT[¶], AND 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 optimize a global, possibly nonconvex, cost while satisfying the agents' constraints, for instance, an energy operator in charge of the management of energy consumption flexibilities of many individual consumers. We provide a privacy-preserving algorithm that computes the optimal allocation of resources, and in which each agent's private information (constraints and individual solution profile) is never revealed either to the central operator or to a third party. Our method relies on an aggregation procedure: we compute iteratively a global allocation of resources, and gradually ensure existence of a disaggregation, that is, individual profiles satisfying agents' private constraints, by a protocol involving the generation of polyhedral cuts and secure multiparty computations. To obtain these cuts, we use an alternate projection method, which is implemented locally by each agent, preserving her privacy needs. We address especially the case in which the local and global constraints define a transportation polytope. Then, we provide theoretical convergence estimates together with numerical results, showing that the algorithm can be effectively used to solve the allocation problem in high dimension, while addressing privacy issues.

**Key words.** resource allocation, privacy, alternate projections, aggregation

**AMS subject classification.** 90

**DOI.** 10.1137/19M127879X

## 1. Introduction.

### 1.1. 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 power generation or market costs and the integration of renewable energies, this operator relies on demand response techniques, taking advantage of the flexibilities of some of the consumers' electric appliances—those which can be controlled without impacting the consumers' comfort, such as electric vehicles or water heaters [26]. However, for confidentiality reasons, consumers may not be willing to provide their consumption constraints or their consumption profiles to a central operator or to any third party, as this information could be used to infer private information such as their presence at home.

The *global problem* of the operator is to find an allocation of power (aggregate flexible consumption) $\boldsymbol{p} = (p_t)_{t\in[T]}$, where $[T] := \{1, \dots, T\}$ is the set of time periods, such that $\boldsymbol{p} \in \mathcal{P}$ (feasibility constraints of power allocation, induced by the power

[*] Received by the editors August 2, 2019; accepted for publication (in revised form) June 22, 2020; published electronically August 24, 2020. Some of the present results have been announced in *Proceedings of the IEEE* 58*th Conference on Decision and Control*, 2019.
https://doi.org/10.1137/19M127879X

[†] EDF R&D, OSIRIS, 91120 Palaiseau, France (olivier.beaude@edf.fr, nadia.oudjane@edf.fr).

[‡] CMAP, École polytechnique, CNRS, and Triscale Innov, 91120 Palaiseau, France (pascal.benchimol@polytechnique.edu).

[§] Inria and CMAP, École polytechnique, CNRS, 91120 Palaiseau, France (stephane.gaubert@inria.fr).

[¶] EDF R&D, OSIRIS, Inria, and CMAP, École polytechnique, CNRS, 91120 Palaiseau, France (paulin.jacquot@polytechnique.edu).

plants constraints), while minimizing the cost $f(\boldsymbol{p})$ (representing the production and operations costs of the centrally controlled power plants). Besides, this aggregate allocation has to match an individual consumption profile $\boldsymbol{x}_n = (x_{n,t})_{t\in[T]}$ for each of the consumer (agent) $n \in [N]$ considered. The problem can be written as follows:

$$\min_{\boldsymbol{x}\in\mathbb{R}^{N\times T},\ \boldsymbol{p}\in\mathcal{P}} f(\boldsymbol{p}), \tag{1.1a}$$

$$\boldsymbol{x}_n \in \mathcal{X}_n \ \forall n \in [N], \tag{1.1b}$$

$$\sum_{n\in[N]} x_{n,t} = p_t \ \forall t \in [T]. \tag{1.1c}$$

The (aggregate) allocation $\boldsymbol{p}$ can be made *public*, that is, revealed to all agents. However, the individual constraint set $\mathcal{X}_n$ and individual profiles $\boldsymbol{x}_n$ 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 (1.1) as the combination of two interdependent subproblems:

(i) Given an aggregate allocation $\boldsymbol{p}$, the *disaggregation problem* consists in finding, for each agent $n$, an individual profile $\boldsymbol{x}_n$ satisfying her individual constraint (1.1b), so that constraint (1.1c) is satisfied. This is equivalent to

$$\text{Find } \boldsymbol{x} \in \mathcal{Y}_{\boldsymbol{p}} \cap \mathcal{X}, \tag{1.2a}$$

$$\text{where } \mathcal{Y}_{\boldsymbol{p}} := \{\boldsymbol{y} \in \mathbb{R}^{NT} | \boldsymbol{y}^\top \mathbb{1}_N = \boldsymbol{p}\} \text{ and } \mathcal{X} := \prod_{n\in[N]} \mathcal{X}_n\ . \tag{1.2b}$$

When (1.2) has a solution, we say that a *disaggregation* exists for $\boldsymbol{p}$.

(ii) For a given subset $\mathcal{Q} \subset \mathcal{P}$, we define the *master problem*,

$$\min_{\boldsymbol{p}\in\mathcal{Q}} f(\boldsymbol{p})\ . \tag{1.3}$$

When $\mathcal{Q}$ 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 (1.1).

Aside from the example above, *resource allocation problems* (optimizing common resources shared by multiple agents) with the same structure as (1.1) find many applications in energy [32, 26], logistics [29], distributed computing [31], health care [36], and telecommunications [46]. 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. Hence, a solution is to rely on decomposition and distributed approaches [10, 35, 41]. Besides, agents' individual constraints are often subject to privacy issues [24]. These considerations have paved the way to the development of privacy-preserving, or nonintrusive, methods and algorithms, e.g., [45, 27].

In this work, except in section 4, we consider that each agent $n \in [N]$ 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 [11]: the sum over the agents is fixed by the aggregate solution $\boldsymbol{p}$, while the sum over the $T$ resources is fixed by the agent global demand constraint. Besides, individual constraints can also include minimal and maximal levels on each resource. For instance, electricity consumers require, through their appliances, a minimal and a maximal power at each time period.

**1.2. Main results.** The main contribution of the paper is to provide a nonintrusive and distributed algorithm (Algorithm 3.4) that computes an aggregate resource allocation $\boldsymbol{p}$, optimal solution of the—possibly nonconvex—optimization problem (1.1), along with feasible individual profiles $\boldsymbol{x}$ 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* $\min_{\boldsymbol{p}\in\mathcal{P}^{(s)}} f(\boldsymbol{p})$, obtained by constructing a decreasing sequence of successive approximations $\mathcal{P}^{(s)}$ of the set of aggregate consumptions $\boldsymbol{p}$ for which a disaggregation exists (see (1.1)), starting from $\mathcal{P}^{(0)} = \mathcal{P}$. At each step, we reduce the set $\mathcal{P}^{(s)}$ by incorporating a new constraint on $\boldsymbol{p}$ (a *cutting plane*), before solving the next master problem. We shall see that this cutting plane can be computed and added to the master problem without revealing any individual information on the agents.

More precisely, to identify whether or not disaggregation (1.2) is feasible and to add a new constraint in the latter case, our algorithm relies on the alternate projections method (APM) [40, 21] for finding a point in the intersection of convex sets. Here, we consider the two following sets: on the one hand, the affine space of profiles $\boldsymbol{x} \in \mathbb{R}^{NT}$ aggregating to a given resource allocation $\boldsymbol{p}$, 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 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. If 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 $\boldsymbol{p}$ of the global problem (1.1), but violated by the current resource allocation (Theorem 3.3). Adding this cut to the *master problem* (1.3) by updating $\mathcal{Q}$ to a specific subset, we can recompute a new resource allocation and repeat this procedure until disaggregation is possible. At this stage, the use of a cryptographic protocol, *secure multiparty computation* (SMC), allows us to preserve the privacy of agents. Another main result stated in this paper is the explicit upper bound on the convergence speed of APM in our framework (Theorem 3.2), 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.

**1.3. Related work.** A standard approach (e.g., [35, 42, 38]) to solve resource allocation problems in a distributed way is to rely on a Lagrangian based decomposition technique, for instance, dual subgradient methods [9, Chapter 6] or ADMM [19]. Such techniques are generally used to decompose large problems into several subproblems of small dimension. However, those methods often require global convexity hypotheses, which are not satisfied in many practical problems (e.g., MILP). We refer the reader to [9, Chapter 6] for more background. On the contrary, our method can be used when the allocation problem (1.1) is not convex.

As explained in section 4, the method proposed here can be related to Benders' decomposition [8]. It differs from Benders' approach in the way cuts are generated: instead of solving linear programs, we use APM and our theoretical results, which provides a decentralized, privacy-preserving, and scalable procedure. In contrast, at each stage, Benders' algorithm requires solving a linear program requiring knowledge of the private constraints of each individual agent (see Remark 4.6 for more details).

The problem of the aggregation of constraints has been studied in the field of energy, in the framework of smart grids [32, 3]. In [32], the authors study the management of energy flexibilities and propose to approximate individual constraints by zonotopic sets to obtain an aggregate feasible set. A centralized aggregated problem is solved via a subgradient method, and a disaggregation procedure of a solution computes individual profiles. In [3], the authors propose to solve the economic power dispatch of a microgrid, subject to several agents' private constraints, by using a Dantzig–Wolfe decomposition method.

APM has been the subject of several works in itself [21, 5, 7]. The authors of [12] provide general results on the convergence rate of APM for semialgebraic 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 bounding the singular values of certain matrices arising from the polyhedral constraints. A remarkable example where an explicit convergence rate for APM has been obtained is [34]. The authors consider there a different class of polyhedra arising in submodular optimization. A common point with our results is the use of spectral graph theory arguments to estimate singular values.

### 1.4. Structure.

In section 2, we describe the class of resource allocation problems that we address. We formulate the idea of decomposition via *disaggregation* subproblems. In section 3, we focus on APM, the subroutine used to solve the disaggregation subproblems. In subsection 3.1, after recalling basic properties of APM, we establish the key result on which relies the proposed decomposition: how to generate a new cut to add in the master problem, from the output of APM. In subsection 3.2, we show how to guarantee the privacy of the proposed procedure by using SMC techniques. In subsection 3.4 , we establish an explicit upper bound on the rate of convergence of APM in our case. In section 4, we generalize part of our results and propose a modified algorithm which applies to polyhedral agents constraints. Finally, in section 5, we present numerical examples: subsection 5.1 gives an illustrative toy example in dimension $T = 4$, while in subsection 5.2, we consider a larger-scale, nonconvex example, coming from the microgrid application presented at the beginning of the introduction.

**Notation.** Bold fonts, like "$\boldsymbol{x}$," are used to denote vectors, while normal fonts, like "$x$," refer to a scalar quantities. We denote by $\boldsymbol{v}^\top$ the transpose of a vector $\boldsymbol{v}$. Recall that we denote by $[N]$ the set $\{1, \dots, N\}$. Calligraphic letters such as $\mathcal{T}, \mathcal{N}, \mathcal{X}$ are used to denote sets, and if $\mathcal{T} \subset [T]$, $\mathcal{T}^c := \{t \in [T] \setminus \mathcal{T}\}$ denotes the complementary set of $\mathcal{T}$. The notation $\mathcal{U}([a, b])$ stands for the uniform distribution on $[a, b]$. The notation $P_{\mathcal{C}}(.)$ refers to the Euclidean projection onto a convex set $\mathcal{C}$. For $d \in \mathbb{N}$, $\mathbb{1}_d$ denotes the vector of ones $(1 \dots 1)^\top \in \mathbb{R}^d$.

## 2. Resource allocation and transportation structure.

### 2.1. A decomposition method based on disaggregation.

As stated in the introduction, we consider a centralized entity (e.g., an energy operator) interested in minimizing a possibly nonconvex cost function $\boldsymbol{p} \mapsto f(\boldsymbol{p})$, where $\boldsymbol{p} \in \mathbb{R}^T$ is the *aggregate allocation* of $T$ dimensional resources (for example, power production over $T$ time periods). This resource allocation $\boldsymbol{p}$ is to be shared between $N$ individual agents, each agent obtaining a part $\boldsymbol{x}_n \in \mathcal{X}_n$, where $\mathcal{X}_n$ denotes the individual feasibility set of agent $n$.

The global problem the operator wants to solve is described in (1.1). We assume that in problem (1.1), the constraints set $\mathcal{X}_n$ and individual profile $\boldsymbol{x}_n$ are *confidential* to agent $n$ and should not be disclosed to the central operator or to another agent.

Let us define the set $\mathcal{P}_{\mathrm{D}}$ of feasible aggregate allocations that are disaggregable as

$$\mathcal{P}_{\mathrm{D}} := \left\{\boldsymbol{p} \in \mathcal{P} \mid \exists \boldsymbol{x} \in \mathcal{X} \ ; \ \boldsymbol{p} = \textstyle\sum_n \boldsymbol{x}_n\right\} . \tag{2.1}$$

Feasibility of problem (1.1) is equivalent to having $\mathcal{P}_{\mathrm{D}}$ not empty.

Constraints for each agent are composed of a global demand over the resources and lower and upper bounds over each resource, as given below.

*Assumption* 1. For each $n \in [N]$, there exist fixed parameters $E_n > 0$, $\overline{\boldsymbol{x}}_n \in \mathbb{R}^T$, $\underline{\boldsymbol{x}}_n \in \mathbb{R}^T$ such that

$$\mathcal{X}_n = \{\boldsymbol{x}_n \in \mathbb{R}^T : \textstyle\sum_{t\in[T]} x_{n,t} = E_n \text{ and } \underline{x}_{n,t} \leqslant x_{n,t} \leqslant \overline{x}_{n,t}\} \ \neq \emptyset. \tag{2.2}$$

In particular, $\mathcal{X}_n$ is convex and compact. Given an allocation $\boldsymbol{p}$, the structure obtained on the matrix $(\boldsymbol{x}_{n,t})_{n,t}$, where sums of coefficients along columns and along rows are fixed, is often referred to as a *transportation problem*. The latter has many applications (see, e.g., [2, 33]). We focus on transportation problems in sections 2 and 3, while in section 4, we give a generalization of some of our results in the general case where $\mathcal{X}_n$ is a polyhedron.

Given a particular allocation $\boldsymbol{p} \in \mathcal{P}$, the operator will be interested to know if this allocation is *disaggregable*, that is, if there exist individual profiles $(\boldsymbol{x}_n)_{n\in[N]} \in \prod_n \mathcal{X}_n$ summing to $\boldsymbol{p}$, or equivalently if the *disaggregation problem* (1.2) has a solution.

Following (1.2), the *disaggregate* profile refers to $\boldsymbol{x}$, while the *aggregate* profile refers to the allocation $\boldsymbol{p}$. Problem (1.2) may not always be feasible. Some necessary conditions for a disaggregation to exist, obtained by summing the individual constraints on $[N]$, are the following *aggregate* constraints:

$$\boldsymbol{p}^\top \mathbb{1}_T = \boldsymbol{E}^\top \mathbb{1}_N, \tag{2.3a}$$

$$\text{and } \ \underline{\boldsymbol{x}}^\top \mathbb{1}_N \leqslant \boldsymbol{p} \leqslant \overline{\boldsymbol{x}}^\top \mathbb{1}_N \ . \tag{2.3b}$$

Because they are necessary, we assume that those aggregate conditions (2.3) hold for vectors of $\mathcal{P}$, as follows.

*Assumption* 2. All vectors $\boldsymbol{p} \in \mathcal{P}$ satisfy (2.3), that is, $\mathcal{P} \subset \{\boldsymbol{p} \in \mathbb{R}^T \mid (2.3) \text{ hold } \}$.

However, conditions (2.3) are not sufficient in general, as explained in the following section and illustrated in Figure 2.1.

**2.2. Equivalent flow problem and Hoffman conditions.** Owing to its special structure, the problem under study can be rewritten as a flow problem in a graph, as stated in Proposition 2.2 and illustrated in Figure 2.1. We refer the reader to the book [15, Chapter 3] for terminology and background.

DEFINITION 2.1. *Consider a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$, edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, demands $d : \mathcal{V} \to \mathbb{R}$ (where $d_v < 0$ means that $v$ is a production node), edge lower capacities $\ell : \mathcal{E} \to \mathbb{R}_+$, and upper capacities $u : \mathcal{E} \to \mathbb{R}_+$. A flow on $\mathcal{G}$ is a function $\boldsymbol{x} : \mathcal{E} \to \mathbb{R}_+$ such that $\boldsymbol{x}$ satisfies the capacity constraints, that is, $\forall e \in \mathcal{E}, \ell_e \leqslant x_e \leqslant u_e$, and Kirchoff's law, that is, $\forall v \in \mathcal{V}, \sum_{e\in\delta_v^+} x_e = d_v + \sum_{e\in\delta_v^-} x_e$, where $\delta_v^+$ (resp., $\delta_v^-$) is the set of edges ending at (resp., departing from) vertex $v$.*

The following proposition is immediate.

PROPOSITION 2.2. *Consider the bipartite graph $\mathcal{G}$ whose set of vertices is the disjoint union $\mathcal{V} = [T] \sqcup [N]$ and whose set of edges is $\mathcal{E} = \{(t, n)\}_{t\in[T],n\in[N]}$. Define*

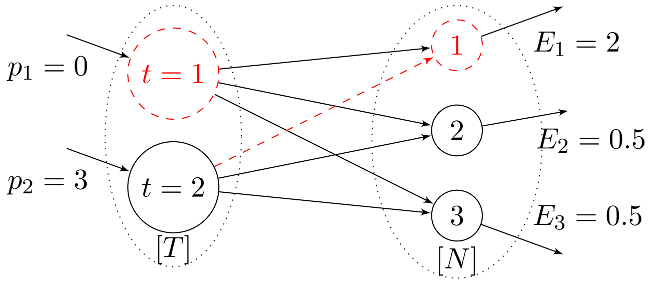


FIG. 2.1. *Example of a flow representation of the disaggregation problem ($T = 2$ and $N = 3$, $\underline{\boldsymbol{x}} = 0$, $\overline{\boldsymbol{x}} = 1$). Here, the aggregate constraints (2.3) are verified, but condition (2.4) written with $\mathcal{A} = \{t_1, n_1\}$ (dashed nodes) does not hold.*

*demands on nodes $[T]$ by $d_t = -p_t$ and demands on nodes $[N]$ by $d_n = E_n$. Assign to each edge $(t, n)$ an upper capacity $u_{n,t} = \overline{x}_{n,t}$ and lower capacity $\ell_{n,t} = \underline{x}_{n,t}$. Then, finding a solution $\boldsymbol{x}$ to (1.2) is equivalent to finding a feasible flow in $\mathcal{G}$.*

Hoffman [23] gave a necessary and sufficient condition for the flow problem to be feasible in a graph with *balanced* demands, that is, $d(\mathcal{V}) := \sum_{v\in\mathcal{V}} d_v = 0$ (total production matches total positive demand, in our case (2.3a)). This generalizes a result of Gale [18]. The stated condition is intuitive: there cannot be a subset of nodes whose demand exceeds its "import capacity."

THEOREM 2.3 (see [23]). *Given a digraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with demand $d \in \mathbb{R}^{\mathcal{V}}$ such that $d(\mathcal{V}) = 0$ and capacities $\ell \in (\mathbb{R} \cup \{-\infty\})^{\mathcal{E}}$ and $u \in (\mathbb{R} \cup \infty)^{\mathcal{E}}$ with $\ell \leqslant u$, there exists a feasible flow $\boldsymbol{x} \in \mathcal{E} \to \mathbb{R}_+$ on $\mathcal{G}$ iff*

$$\forall \mathcal{A} \subset \mathcal{V}, \quad \sum_{e\in\delta^+(\mathcal{A})} u_e \geqslant \sum_{v\in\mathcal{A}} d_v + \sum_{e\in\delta^+(\mathcal{A}^c)} \ell_e \ , \tag{2.4}$$

*where $\delta_+(\mathcal{A}) := \{(u, v) \in \mathcal{E} | u \in \mathcal{A}^c, v \in \mathcal{A}\}$ is the set of edges coming to set $\mathcal{A}$ and $\mathcal{A}^c := \mathcal{V} \setminus \mathcal{A}$.*

Proposition 2.4 translates Theorem 2.3 in our framework:

PROPOSITION 2.4. *Given an aggregate resource allocation $\boldsymbol{p} = (p_t)_{t\in[T]} \in \mathcal{P}$, the disaggregation problem is feasible, meaning that $\boldsymbol{p} \in \mathcal{P}_{\mathrm{D}}$, iff*

$$\forall \mathcal{T} \subset [T], \forall \mathcal{N} \subset [N], \quad \sum_{t\in\mathcal{T}} p_t - \sum_{n\in\mathcal{N}} E_n + \sum_{t\notin\mathcal{T}, n\in\mathcal{N}} \underline{x}_{n,t} \leqslant \sum_{t\in\mathcal{T}, n\notin\mathcal{N}} \overline{x}_{n,t} \, . \tag{2.5}$$

*Proof.* We apply (2.4) with $\mathcal{A} := \mathcal{T}^c \cup \mathcal{N}^c$ and use the equality $d(\mathcal{V}) = 0 = \sum_{v\in\mathcal{A}} d_v + \sum_{v\in\mathcal{A}^c} d_v$. ▯

From Theorem 2.3 or Proposition 2.4 above, one can see that the aggregate constraints (2.3) are in general not sufficient to ensure that the disaggregation problem has a solution.

For a given set $\mathcal{T}$, there is a choice of $\mathcal{N}$ which leads to the strongest inequality (2.5), namely

$$\sum_{t\in\mathcal{T}} p_t \leqslant \min_{\mathcal{N}\subset[N]} \left\{ \sum_{n\in\mathcal{N}} E_n - \sum_{t\notin\mathcal{T}, n\in\mathcal{N}} \underline{x}_{n,t} + \sum_{t\in\mathcal{T}, n\notin\mathcal{N}} \overline{x}_{n,t} \right\}, \tag{2.6}$$

In this way, we get $2^T - 2$ inequalities corresponding to the proper subsets $\mathcal{T} \subset [T]$. Moreover, in general, these $2^T - 2$ inequalities are not redundant. Although this is not stated in [23], this is a classical result whose proof is elementary.

## 3. Disaggregation based on APM.

**3.1. Generation of Hoffman's cuts by APM.** In this section, we propose an algorithm that solves (1.1) while preserving the privacy of the agent's constraints $\mathcal{X}_n$ and individual profile $\boldsymbol{x}_n \in \mathbb{R}^T$. To do this, the proposed algorithm is implemented in a decentralized manner and relies on APM to solve the disaggregation problem (1.2).

Let us consider the polyhedron enforcing the agent's constraints:

$$\mathcal{X} := \mathcal{X}_1 \times \cdots \times \mathcal{X}_N \quad \text{where } \mathcal{X}_n := \Big\{ \boldsymbol{x}_n \in \mathbb{R}^T_+ \mid \textstyle\sum_{t\in[T]} x_{n,t} = E_n \text{ and} \\ \forall t,\ \underline{x}_{n,t} \leqslant x_{n,t} \leqslant \overline{x}_{n,t} \Big\}.$$

Besides, given an allocation $\boldsymbol{p} \in \mathcal{P}$, we consider the set of profiles aggregating to $\boldsymbol{p}$:

$$\mathcal{Y}_{\boldsymbol{p}} := \Big\{ \boldsymbol{x} \in \mathbb{R}^{NT} \mid \forall t \in [T], \textstyle\sum_{n\in[N]} x_{n,t} = p_t \Big\} \ .$$

Note that $\mathcal{Y}_{\boldsymbol{p}}$ is an affine subspace of $\mathbb{R}^{NT}$ (to be distinguished from $\mathcal{P}$ which is a subset of $\mathbb{R}^T$) and that $\mathcal{Y}_{\boldsymbol{p}} \cap \mathcal{X}$ is empty iff $\boldsymbol{p} \notin \mathcal{P}_{\mathrm{D}}$, according to the definition of $\mathcal{P}_{\mathrm{D}}$ in (2.1). The idea of the proposed algorithm is to build a finite sequence of decreasing subsets $(\mathcal{P}^{(s)})_{0\leqslant s\leqslant S}$ such that

$$\mathcal{P} = \mathcal{P}^{(0)} \supset \mathcal{P}^{(1)} \supset \cdots \supset \mathcal{P}^{(S)} \supset \mathcal{P}_{\mathrm{D}} \ .$$

At each iteration, a new aggregate resource allocation $\boldsymbol{p}^{(s)}$ is obtained by solving an instance of the master problem introduced in (1.3) with $\mathcal{Q} = \mathcal{P}^{(s)}$:

$$\min_{\boldsymbol{p}\in\mathbb{R}^T} f(\boldsymbol{p}) \tag{3.1a}$$

$$\text{s.t. } \boldsymbol{p} \in \mathcal{P}^{(s)} \ . \tag{3.1b}$$

In what follows, we will refer to (3.1) as the *master problem.* Our procedure relies on the following immediate observation.

PROPOSITION 3.1. *If $\boldsymbol{p}^{(s)}$ is a solution of* (3.1) *and $\boldsymbol{x} \in \mathcal{Y}_{\boldsymbol{p}^{(s)}} \cap \mathcal{X}$, then $(\boldsymbol{p}^{(s)}, \boldsymbol{x})$ is an optimal solution of the initial problem* (1.1).

Having in hand a solution $\boldsymbol{p}^{(s)}$, we can check whether $\mathcal{Y}_{\boldsymbol{p}^{(s)}} \cap \mathcal{X} \neq \emptyset$, and then find a vector $\boldsymbol{x}$ in this set, using APM on $\mathcal{X}$ and $\mathcal{Y}_{\boldsymbol{p}^{(s)}}$, as described in Algorithm 3.1 below (where $\mathcal{Y} = \mathcal{Y}_{\boldsymbol{p}}$).

The idea of using cyclic projections to compute a point in the intersection of two sets comes from von Neumann [40], who applied it to the case of affine subspaces. We next recall the basic convergence result concerning the APM.

THEOREM 3.2 (see [21]). *Let $\mathcal{X}$ and $\mathcal{Y}$ be two closed convex sets with $\mathcal{X}$ bounded, and let $(\boldsymbol{x}^{(k)})_k$ and $(\boldsymbol{y}^{(k)})_k$ be the two infinite sequences generated by the APM on $\mathcal{X}$ and $\mathcal{Y}$ (Algorithm* 3.1*) with $\varepsilon_{\mathrm{cvg}} = 0$. Then there exists $\boldsymbol{x}^\infty \in \mathcal{X}$ and $\boldsymbol{y}^\infty \in \mathcal{Y}$ such that*

**Algorithm 3.1.** Alternate projections method (APM)

```
Require: Start with $\boldsymbol{y}^{(0)}$, $k = 0$ , $\varepsilon_{\text{cvg}}$, a norm $\|\cdot\|$ on $\mathbb{R}^{NT}$
repeat
    $\boldsymbol{x}^{(k+1)} \leftarrow P_{\mathcal{X}}(\boldsymbol{y}^{(k)})$
    $\boldsymbol{y}^{(k+1)} \leftarrow P_{\mathcal{Y}}(\boldsymbol{x}^{(k+1)})$
    $k \leftarrow k + 1$
until $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\| < \varepsilon_{\text{cvg}}$
```

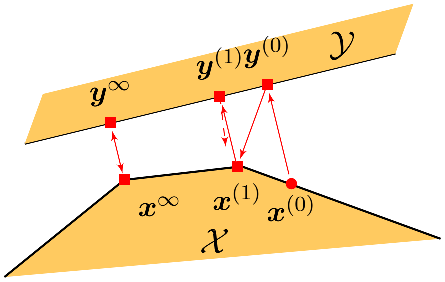

$\boldsymbol{y}^{\infty}$
$\boldsymbol{y}^{(1)}\boldsymbol{y}^{(0)}$
$\mathcal{Y}$
$\boldsymbol{x}^{\infty}$
$\boldsymbol{x}^{(1)}$
$\boldsymbol{x}^{(0)}$
$\mathcal{X}$

FIG. 3.1. *APM on two sets $\mathcal{X}$ and $\mathcal{Y}$. When $\mathcal{X} \cap \mathcal{Y} = \emptyset$, APM cycles over two points $\boldsymbol{x}^{\infty}$ and $\boldsymbol{y}^{\infty}$.*

$$\boldsymbol{x}^{(k)} \underset{k\to\infty}{\longrightarrow} \boldsymbol{x}^{\infty} \ , \quad \boldsymbol{y}^{(k)} \underset{k\to\infty}{\longrightarrow} \boldsymbol{y}^{\infty}; \tag{3.2a}$$

$$\|\boldsymbol{x}^{\infty} - \boldsymbol{y}^{\infty}\|_2 = \min_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{y}\in\mathcal{Y}} \|\boldsymbol{x} - \boldsymbol{y}\|_2 \ . \tag{3.2b}$$

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

The convergence theorem is illustrated in Figure 3.1 in the case where $\mathcal{X} \cap \mathcal{Y} = \emptyset$, that is, when the disaggregation problem (1.2) is not feasible. The idea of the algorithm proposed in this paper is that when $\mathcal{Y}_{\boldsymbol{p}^{(s)}} \cap \mathcal{X} = \emptyset$, use the resulting vectors $\boldsymbol{x}^{\infty}$ and $\boldsymbol{y}^{\infty}$ to construct a new subset $\mathcal{P}^{(s+1)}$ by adding a constraint of type (2.5) to $\mathcal{P}^{(s)}$: indeed, from Proposition 2.4, we know that if $\mathcal{Y}_{\boldsymbol{p}^{(s)}} \cap \mathcal{X} = \emptyset$, there exists at least one violated inequality (2.5).

The difficulty is to guess such a violated inequality from the $2^T$ possible inequalities. It turns out that using the output of APM, we can build such an inequality.

Suppose that we obtain $\boldsymbol{x}^{\infty} \neq \boldsymbol{y}^{\infty}$ as defined in Theorem 3.2. We get a periodic cycle of APM, that is, we have $\boldsymbol{x}^{\infty} = P_{\mathcal{X}}(\boldsymbol{y}^{\infty})$ and $\boldsymbol{y}^{\infty} = P_{\mathcal{Y}}(\boldsymbol{x}^{\infty})$, and the couple $(\boldsymbol{x}^{\infty}, \boldsymbol{y}^{\infty})$ is the solution of the following optimization problem, with given parameters $(E_n)_n$ and $(p_t)_t$:

$$\min_{\boldsymbol{x},\boldsymbol{y}} \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 \tag{3.3a}$$

$$\forall n \in [N], \ \sum_{t\in[T]} x_{n,t} = E_n \qquad (\lambda_n), \tag{3.3b}$$

$$\forall n \in [N], \forall t \in [T], \ \underline{x}_{n,t} \leqslant x_{n,t} \leqslant \overline{x}_{n,t} \qquad (\underline{\mu}_{n,t}, \overline{\mu}_{n,t}), \tag{3.3c}$$

$$\forall t \in [T], \ \sum_{n\in[N]} y_{n,t} = p_t \qquad (\nu_t) \ , \tag{3.3d}$$

where $\lambda_n \in \mathbb{R}$, $\underline{\mu}_{n,t}, \overline{\mu}_{n,t} \in \mathbb{R}_+$, and $\nu_t \in \mathbb{R}$ are the Lagrangian multipliers associated to the constraints (3.3b), (3.3c), (3.3d), with the associated Lagrangian function:

$$\begin{aligned}\mathcal{L}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\lambda},\boldsymbol{\mu},\boldsymbol{\nu}) = \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 - \boldsymbol{\lambda}^\top \left(\textstyle\sum_t x_{n,t} - E_n\right)_n - \underline{\boldsymbol{\mu}}^\top(\boldsymbol{x}-\underline{\boldsymbol{x}}) \\ - \overline{\boldsymbol{\mu}}^\top(\overline{\boldsymbol{x}}-\boldsymbol{x}) - \boldsymbol{\nu}^\top \left(\textstyle\sum_n \boldsymbol{y}_n - \boldsymbol{p}\right).\end{aligned}$$

The stationarity condition of the Lagrangian with respect to the variable $y_{n,t}$ yields

$$\forall n \in [N], \forall t \in [T],\ y_{n,t} = x_{n,t} + \nu_t\ . \tag{3.4}$$

Let us consider the sets $\mathcal{T} \subset [T]$ and $\mathcal{N} \subset [N]$ defined from the output $(\boldsymbol{x}^\infty, \boldsymbol{y}^\infty)$ of APM on $\mathcal{X}$ and $\mathcal{Y}_{\boldsymbol{p}}$ as follows:

$$\mathcal{T} := \left\{t \in [T] \mid \exists n \in [N],\ y^\infty_{n,t} > \overline{x}_{n,t}\right\}\ , \tag{3.5a}$$

$$\mathcal{N} := \left\{n \in [N] \mid E_n - \textstyle\sum_{t\notin\mathcal{T}} \underline{x}_{n,t} - \sum_{t\in\mathcal{T}} \overline{x}_{n,t} < 0\right\}\ . \tag{3.5b}$$

In Theorem 3.3 below, we show that applying the inequality (2.5) with the sets $\mathcal{T}$ and $\mathcal{N}$ defined in (3.5) defines a valid inequality for the disaggregation problem violated by the current allocation $\boldsymbol{p}$.

The intuition behind the definition of $\mathcal{T}$ and $\mathcal{N}$ in (3.5) is the following: $\mathcal{T}$ is the subset of resources for which there is an oversupply (which overcomes the upper bound for at least one agent). Once $\mathcal{T}$ is defined, $\mathcal{N}$ is the associated subset of $[N]$ minimizing the right-hand side of (2.6). Indeed, (2.6) can be rewritten as

$$\sum_{t\in\mathcal{T}} p_t \leqslant \min_{\mathcal{N}\subset[N]} \left\{\sum_{n\in\mathcal{N}} \left(E_n - \sum_{t\notin\mathcal{T}} \underline{x}_{n,t} - \sum_{t\in\mathcal{T}} \overline{x}_{n,t}\right)\right\} + \sum_{t\in\mathcal{T}, n\in[N]} \overline{x}_{n,t}.$$

Theorem 3.3 below is the key result on which relies the algorithm proposed in this paper.

Theorem 3.3. *Consider the sequence of iterates* $(\boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)})_{k\in\mathbb{N}}$ *generated by APM on* $\mathcal{X}$ *and* $\mathcal{Y}_{\boldsymbol{p}}$ *(see Algorithm* 3.1*). Then one of the following holds:*

(i) *if* $\mathcal{X} \cap \mathcal{Y}_{\boldsymbol{p}} \neq \emptyset$, *then* $\boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)} \underset{k\to\infty}{\longrightarrow} \boldsymbol{x}^\infty \in \mathcal{X} \cap \mathcal{Y}_{\boldsymbol{p}}$;

(ii) *otherwise, if* $\mathcal{X} \cap \mathcal{Y}_{\boldsymbol{p}} = \emptyset$, *then* $\boldsymbol{x}^{(k)} \underset{k\to\infty}{\longrightarrow} \boldsymbol{x}^\infty \in \mathcal{X}$ *and* $\boldsymbol{y}^{(k)} \underset{k\to\infty}{\longrightarrow} \boldsymbol{y}^\infty \in \mathcal{Y}_{\boldsymbol{p}}$. *Then, considering the sets* $\mathcal{T}$ *and* $\mathcal{N}$ *defined in* (3.5) *gives an inequality of Hoffman* (2.5) *violated by* $\boldsymbol{p}$, *that is,*

$$\sum_{n\in\mathcal{N}} E_n - \sum_{t\in\mathcal{T}} p_t + \sum_{t\in\mathcal{T}, n\notin\mathcal{N}} \overline{x}_{n,t} - \sum_{t\notin\mathcal{T}, n\in\mathcal{N}} \underline{x}_{n,t} < 0\ . \tag{3.6}$$

*Moreover, this Hoffman inequality can be written as a function of* $\boldsymbol{x}^\infty$, *as follows:*

$$A_{\mathcal{T}}(\boldsymbol{x}^\infty) < \sum_{t\in\mathcal{T}} p_t \text{ with } A_{\mathcal{T}}(\boldsymbol{x}^\infty) := \sum_{t\in\mathcal{T}} \sum_{n\in[N]} x^\infty_{n,t}\ . \tag{3.7}$$

Before giving the proof of Theorem 3.3, we need to show some technical properties on the sets $\mathcal{T}$, $\mathcal{N}$. For simplicity of notation, we use $\boldsymbol{x}$ and $\boldsymbol{y}$ to denote $\boldsymbol{x}^\infty$ and $\boldsymbol{y}^\infty$ in Proposition 3.4 and the proof of Theorem 3.3.

PROPOSITION 3.4. *With $\boldsymbol{x} \neq \boldsymbol{y}$ solutions of problem* (3.3) *(outputs of APM on $\mathcal{X}$ and $\mathcal{Y}_{\boldsymbol{p}}$ with $\varepsilon_{\text{cvg}} = 0$),*

(i) $\forall t \in \mathcal{T}, \forall n \notin \mathcal{N},\ x_{n,t} = \overline{x}_{n,t}$ *and* $y_{n,t} > \overline{x}_{n,t}$*;*

(ii) $\mathcal{T} = \{t \mid \nu_t > 0\} = \{t \mid p_t > \sum_n x_{n,t}\}$*, where $\nu_t$ is the optimal Lagrangian multiplier associated to* (3.3d)*;*

(iii) $\forall n \in \mathcal{N}, \lambda_n < 0$*;*

(iv) $\forall t \notin \mathcal{T}, \forall n \in \mathcal{N},\ x_{n,t} = \underline{x}_{n,t}$*;*

(v) *the sets $\mathcal{T}, \mathcal{T}^c$, $\mathcal{N}$, and $\mathcal{N}^c$ are nonempty.*

The proof of Proposition 3.4 is technical and given in Appendix A. With Proposition 3.4, we are now ready to prove Theorem 3.3.

*Proof of Theorem* 3.3. We have

$$\begin{aligned}
&\sum_{n\in\mathcal{N}} E_n + \sum_{t\in\mathcal{T},n\notin\mathcal{N}} \overline{x}_{n,t} - \sum_{t\notin\mathcal{T},n\in\mathcal{N}} \underline{x}_{n,t} - \sum_{t\in\mathcal{T}} p_t \\
&\quad = \sum_{n\in\mathcal{N}} \sum_{t} x_{n,t} + \sum_{t\in\mathcal{T},n\notin\mathcal{N}} \overline{x}_{n,t} - \sum_{t\notin\mathcal{T},n\in\mathcal{N}} \underline{x}_{n,t} - \sum_{t\in\mathcal{T}} p_t \qquad \text{(from (3.3b))} \\
&\quad = \sum_{n\in\mathcal{N}} \left( \sum_{t\notin\mathcal{T}} \underline{x}_{n,t} + \sum_{t\in\mathcal{T}} x_{n,t} \right) + \sum_{t\in\mathcal{T},n\notin\mathcal{N}} x_{n,t} \\
&\qquad - \sum_{t\notin\mathcal{T},n\in\mathcal{N}} \underline{x}_{n,t} - \sum_{t\in\mathcal{T}} p_t \quad \text{(from Proposition 3.4(i) and (iv))} \\
&\quad = \sum_{t\in\mathcal{T}} \left( \sum_{n\in\mathcal{N}} x_{n,t} + \sum_{n\notin\mathcal{N}} x_{n,t} \right) - \sum_{t\in\mathcal{T}} p_t = \sum_{t\in\mathcal{T}} \left( \sum_{n\in[N]} x_{n,t} - \sum_{n\in[N]} y_{n,t} \right) \\
&\quad = \sum_{t\in\mathcal{T}} \left( \sum_{n\in[N]} -\nu_t \right) = \sum_{t\in\mathcal{T}} \left( -\sum_{n\in[N]} |x_{n,t} - y_{n,t}| \right)
\end{aligned}$$

using the stationarity conditions (3.4) and $\forall t \in \mathcal{T}, \nu_t > 0$ by Proposition 3.4(ii). Moreover, using

$$\sum_{t\in[T]} \sum_{n\in[N]} (x_{n,t} - y_{n,t}) = \sum_{n\in[N]} E_n - \sum_{t\in[T]} p_t = 0 \ , \tag{3.8}$$

obtained from Assumption 2, we see that

$$\sum_{t\in\mathcal{T}} \left( -\sum_{n\in[N]} |x_{n,t} - y_{n,t}| \right) = -(\|\boldsymbol{x} - \boldsymbol{y}\|_1)/2 < 0 \ , \tag{3.9}$$

which shows (3.6). We now show that inequality (3.7) is obtained by a rewriting of (3.6); indeed,

$$\begin{aligned}
&\sum_{n\in\mathcal{N}} E_n + \sum_{t\in\mathcal{T},n\notin\mathcal{N}} \overline{x}_{n,t} - \sum_{t\notin\mathcal{T},n\in\mathcal{N}} \underline{x}_{n,t} \\
&\quad = \sum_{n\in\mathcal{N}} \sum_{t\in[T]} x_{n,t} + \sum_{t\in\mathcal{T},n\notin\mathcal{N}} x_{n,t} - \sum_{t\notin\mathcal{T},n\in\mathcal{N}} x_{n,t} \qquad \text{(from Proposition 3.4(i) and (iv))} \\
&\quad = \sum_{t\in\mathcal{T},n\in\mathcal{N}} x_{n,t} + \sum_{t\in\mathcal{T},n\notin\mathcal{N}} x_{n,t} = \sum_{t\in\mathcal{T}} \sum_{n\in[N]} x_{n,t} \ = A_{\mathcal{T}}(\boldsymbol{x}).
\end{aligned}$$

□

*Remark* 3.5. An alternative to APM is to use Dykstra's projections algorithm [17]. It is shown in [6] that the outputs of Dykstra's algorithm also satisfy the conditions given in Theorem 3.2, thus Theorem 3.3 will also hold for this algorithm. However, in this paper we focus on APM as it is simpler and, to our knowledge, there is no guarantee that Dykstra's algorithm would be faster in this framework.

Suppose, as before, that the two sequences generated by APM on $\mathcal{X}$ and $\mathcal{Y}$ converge to two distinct points $\boldsymbol{x}^\infty$ and $\boldsymbol{y}^\infty$. Then, at each round $k$ and considering an arbitrary $n \in [N]$, let us define the multiplier $\boldsymbol{\nu}^{(k)} = \boldsymbol{y}_n^{(k)} - \boldsymbol{x}_n^{(k)}$. The sequence $(\boldsymbol{\nu}^{(k)})_k$ converges to $\boldsymbol{\nu}^\infty = \boldsymbol{y}_n^\infty - \boldsymbol{x}_n^\infty$. The set $\mathcal{T}$ of Theorem 3.3 is

$$\mathcal{T}^\infty := \{t \in [T] \mid 0 < \nu_t^\infty\}\,, \tag{3.10}$$

which raises an issue for practical computation, as $\boldsymbol{\nu}^\infty$ is only obtained *ultimately* by APM, possibly in infinite time. To have access to $\mathcal{T}^\infty$ *in finite time*, that is, from one of the iterates $(\boldsymbol{\nu}^{(k)})_k$, we consider the set

$$\mathcal{T}^{(K)} := \left\{t \in [T] \mid B\varepsilon_{\text{cvg}} < \nu_t^{(K)}\right\}, \tag{3.11}$$

where $\varepsilon_{\text{cvg}}$ is the tolerance for convergence of APM as defined in Algorithm 3.1, $B > 0$ is a constant, and $K$ (depending on $\varepsilon_{\text{cvg}}$) is the first integer such that $\|\boldsymbol{x}^{(K)} - \boldsymbol{x}^{(K-1)}\| < \varepsilon_{\text{cvg}}$.

We next show that we can choose $B$ to ensure that $\mathcal{T}^{(K)} = \mathcal{T}^\infty$ for $\varepsilon_{\text{cvg}}$ small enough. We rely on the geometric convergence rate of APM on polyhedra [12, 34].

Proposition 3.6 (see [34]). *If $\mathcal{X}$ and $\mathcal{Y}$ are polyhedra, there exists $\rho \in (0, 1)$ such that the sequences $(\boldsymbol{x}^{(k)})_k$ and $(\boldsymbol{y}^{(k)})_k$ generated by APM verify $\forall k \geqslant 1$*

$$\left\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\right\|_2 \leqslant \rho \left\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\right\|_2 \quad \textit{and} \quad \left\|\boldsymbol{y}^{(k+1)} - \boldsymbol{y}^{(k)}\right\|_2 \leqslant \rho \left\|\boldsymbol{y}^{(k)} - \boldsymbol{y}^{(k-1)}\right\|_2 \ .$$

Proposition 3.6 applies to any polyhedra $\mathcal{X}$ and $\mathcal{Y}$. In subsection 3.4 we shall give an explicit upper bound on the constant $\rho$ in the specific transportation case given by (1.1c) and (2.2).

From the previous proposition, we can quantify the distance to the limits in terms of $\rho$.

Lemma 3.7. *Consider an integer $K$ such that the sequence $(\boldsymbol{x}^{(k)})_{k \geqslant 0}$ generated by APM satisfies $\|\boldsymbol{x}^{(K)} - \boldsymbol{x}^{(K-1)}\| \leqslant \varepsilon_{\text{cvg}}$; then we have for any $K' \geqslant K - 1$*

$$\left\|\boldsymbol{x}^\infty - \boldsymbol{x}^{(K')}\right\| \leqslant \frac{\varepsilon_{\text{cvg}}}{1-\rho} \ .$$

*Proof.* From Proposition 3.6, we have for any $k \geqslant K$

$$\left\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(K')}\right\| \leqslant \sum_{s=0}^{k-K'} \left\|\boldsymbol{x}^{(K+s+1)} - \boldsymbol{x}^{(K+s)}\right\| \leqslant \sum_{s=0}^{k-K'} \rho^s \left\|\boldsymbol{x}^{(K'+1)} - \boldsymbol{x}^{(K')}\right\| \leqslant \frac{1}{1-\rho}\varepsilon_{\text{cvg}} \ ,$$

so that, by taking the limit $k \to \infty$, one obtains $\|\boldsymbol{x}^\infty - \boldsymbol{x}^{(K')}\| \leqslant \frac{\varepsilon_{\text{cvg}}}{1-\rho}$. □

With this lemma, we can state the condition on $B$ ensuring the desired property.

Proposition 3.8. *Define $\underline{\nu} := \min\{|\nu_t^\infty| > 0\}$ (least nonzero element of $\boldsymbol{\nu}^\infty$). If the constants $B$ and $\varepsilon_{\text{cvg}} > 0$ are chosen such that $B > \frac{1}{1-\rho}$ and $\varepsilon_{\text{cvg}} \times 2B < \underline{\nu}$, and Algorithm 3.1 stops at iteration $K$, then we have*

$$\mathcal{T}^{(K)} = \mathcal{T}^\infty \ .$$

*Proof.* Let $t \in \mathcal{T}^\infty$, that is, $\nu_t^\infty > 0$, which is equivalent to $\nu_t^\infty \geqslant \underline{\nu}$ by definition of $\underline{\nu}$. We have

$$\nu_t^{(K)} = \frac{1}{N}\left(p_t - \sum_n x_{n,t}^{(K)}\right) = \frac{1}{N}\left(p_t - \sum_n x_{n,t}^{\infty}\right) + \frac{1}{N}\left(\sum_n x_{n,t}^{\infty} - \sum_n x_{n,t}^{(K)}\right)$$
$$> \nu_t^\infty - \tfrac{\varepsilon_{\text{cvg}}}{1-\rho} \geqslant \underline{\nu} - \tfrac{\varepsilon_{\text{cvg}}}{1-\rho} > \varepsilon_{\text{cvg}}\left(2B - \tfrac{1}{1-\rho}\right)\ ,$$

and this last quantity is greater than $B\varepsilon_{\text{cvg}}$ as soon as $B \geqslant \frac{1}{1-\rho}$, thus $t \in \mathcal{T}^{(K)}$.

Conversely, if $t \in \mathcal{T}^{(K)}$, then

$$\nu_t^{\infty} = \frac{1}{N}\left(p_t - \sum_n x_{n,t}^{\infty}\right) = \frac{1}{N}\left(p_t - \sum_n x_{n,t}^{(K)}\right) - \frac{1}{N}\left(\sum_n x_{n,t}^{\infty} - \sum_n x_{n,t}^{(K)}\right)$$
$$\geqslant \nu_t^{(K)} - \tfrac{B}{1-\rho}\ >\ \nu_t^{(K)} - B\varepsilon_{\text{cvg}} \geqslant (B-B)\,\varepsilon_{\text{cvg}} \geqslant 0\ ,$$

so that $t \in \mathcal{T}^\infty$. Furthermore, the "approximated" cut $\sum_{t\in\mathcal{T}}(\sum_{n\in[N]} x_{n,t}^{(K)} - p_t) \geqslant 0$ is violated by the current value of $\boldsymbol{p}$ (or $\boldsymbol{p}^{(s)}$ at iteration $s$) in the algorithm as

$$\sum_{t\in\mathcal{T}}\left(\sum_{n\in[N]} x_{n,t}^{(K)} - p_t\right) \leqslant \sum_{t\in\mathcal{T}}\left(\sum_{n\in[N]} x_{n,t}^{(K)} - x_{n,t}^{\infty}\right) + \sum_{t\in\mathcal{T}}\left(\sum_{n\in[N]} x_{n,t}^{\infty} - p_t\right)$$
$$\leqslant \left\|\boldsymbol{x}^{(K)} - \boldsymbol{x}^\infty\right\|_1 - \frac{1}{2}\|\boldsymbol{x}^\infty - \boldsymbol{y}^\infty\|_1$$

using (3.8) and (3.9). This last quantity is negative as soon as $\|\boldsymbol{x}^{(K)} - \boldsymbol{x}^\infty\|_1 < \frac{1}{2}\|\boldsymbol{x}^\infty - \boldsymbol{y}^\infty\|_1$, which holds in particular if $B\varepsilon_{\text{cvg}} < \frac{1}{2}\|\boldsymbol{x}^\infty - \boldsymbol{y}^\infty\|_1$. □

This second proposition shows a surprising result: even if we do not have access to the limit $\boldsymbol{x}^\infty$, we can compute *in finite time* the *exact* left-hand-side term $A_{\mathcal{T}}(\boldsymbol{x}^\infty)$ of the cut (3.7).

PROPOSITION 3.9. *Under the hypotheses of Proposition* 3.8*, we have*

$$A_{\mathcal{T}}(\boldsymbol{x}^{(K)}) = \sum_{t\in\mathcal{T}}\sum_{n\in[N]} x_{n,t}^{(K)} = A_{\mathcal{T}}(\boldsymbol{x}^\infty)\ .$$

*Proof.* We start by showing some technical properties similar to Proposition 3.4.

LEMMA 3.10. *The iterate $\boldsymbol{x}^{(K)}$ satisfies the following properties:*

(i) $\forall t \in \mathcal{T}, \forall n \notin \mathcal{N}, x_{n,t}^{(K)} = x_{n,t}^\infty = \overline{x}_{n,t}$;

(ii) $\forall t \notin \mathcal{T}, \forall n \in \mathcal{N},\ x_{n,t}^{(K)} = x_{n,t}^\infty = \underline{x}_{n,t}$.

The proof of Lemma 3.10 is similar to Proposition 3.4 and is given in Appendix B. Then, having in mind that $\mathcal{T}^{(K)} = \mathcal{T}^\infty$ from Proposition 3.8, and $\mathcal{N}$ is obtained from $\mathcal{T}^\infty$ by (3.5), we obtain

$$A_{\mathcal{T}}(\boldsymbol{x}^{(K)}) = \sum_{n\in\mathcal{N}}\left(\sum_{t\notin\mathcal{T}} \underline{x}_{n,t} + \sum_{t\in\mathcal{T}} x_{n,t}^{(K)}\right) - \sum_{t\notin\mathcal{T}, n\in\mathcal{N}} \underline{x}_{n,t} + \sum_{t\in\mathcal{T}, n\notin\mathcal{N}} x_{n,t}^{(K)}$$
$$\leqslant \sum_{n\in\mathcal{N}}\sum_{t\in[T]} x_{n,t}^{(K)} - \sum_{t\notin\mathcal{T}, n\in\mathcal{N}} \underline{x}_{n,t} + \sum_{t\in\mathcal{T}, n\notin\mathcal{N}} \overline{x}_{n,t} \qquad \text{(from Lemma 3.10)},$$

which equals to $A_{\mathcal{T}}(\boldsymbol{x}^\infty)$ as we have $\sum_{t\in[T]} x_{n,t}^{(K)} = E_n$ for each $n \in [N]$. □

Before presenting our algorithm using this last result, we focus on the technique of SMC, which will be used here to ensure the privacy of an agent's constraints and profiles while running APM.

**3.2. Privacy-preserving projections through SMC.** APM, as described in Algorithm 3.1, enables a distributed implementation in our context, by the structure of the algorithm itself: the operator computes the projection on $\mathcal{Y}_{\boldsymbol{p}}$ while each agent $n$ can compute, possibly in parallel, the projection on $\mathcal{X}_n$ of the new profile transmitted by the operator. This enables each agent (as well as the operator) to keep her individual constraint and not reveal it to the operator or other agents. However, if this agent had to transmit back her newly computed individual profile to the operator for the next iteration, privacy would not be preserved. We next show that using an SMC summation protocol [43] we can avoid this communication of individual profiles and implement APM without revealing the sequence of agents' profiles $\boldsymbol{x}$ to the operator.

For this, we use the fact that $\mathcal{Y}_{\boldsymbol{p}}$ is an affine subspace and thus the projection on $\mathcal{Y}_{\boldsymbol{p}}$ can be obtained explicitly componentwise. Indeed, as $\mathcal{P}_{\mathcal{Y}_{\boldsymbol{p}}}(\boldsymbol{x})$ is the solution $\boldsymbol{y}$ of the quadratic program $\min_{\boldsymbol{y}\in\mathbb{R}^{NT} \mid \sum_n \boldsymbol{y}_n = \boldsymbol{p}} \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$, we obtain optimality conditions similar to (3.4), and summing these equalities on $[N]$, we get $N\nu_t = \boldsymbol{p}_t - \sum_{n\in[N]} x_{n,t}$ for each $t \in [T]$, and thus

$$\forall n \in [N],\ [\mathcal{P}_{\mathcal{Y}_{\boldsymbol{p}}}(\boldsymbol{x})]_n = \boldsymbol{x}_n + \frac{1}{N}\left(\boldsymbol{p} - \textstyle\sum_{m\in[N]} \boldsymbol{x}_m\right) . \tag{3.12}$$

Thus, having access to the *aggregate* profile $\boldsymbol{S} := \sum_{n\in[N]} \boldsymbol{x}_n$, each agent can compute locally the component of the projection on $\mathcal{Y}_{\boldsymbol{p}}$ of her profile, instead of transmitting the profile to the operator for computing the projection in a centralized way.

Using the SMC principle, introduced by [43] and [20], the sum $\boldsymbol{S}$ can be computed in a nonintrusive manner and by several communications between agents and the operator, as described in Algorithm 3.2. The main idea of the SMC summation protocol [4, 39] below is that, instead of sending her profile $\boldsymbol{x}_n$, agent $n$ *splits* $x_{n,t}$ for each $t$ into $N$ random parts $(s_{n,t,m})_m$, according to a uniform distribution and summing to $\boldsymbol{x}_{n,t}$ (lines 2–3). Thus, each part $s_{n,t,m}$ taken individually does not reveal any information on $\boldsymbol{x}_n$ nor on $\mathcal{X}_n$ 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 $\boldsymbol{\sigma}_n$ (line 7), which does not contain any

**Algorithm 3.2.** SMC of aggregate (SMCA) $\sum_{n\in[N]} \boldsymbol{x}_n$

```
Require: profile $\boldsymbol{x}_n$ for each $n \in [N]$, parameter $R > \max_t \max\{\sum_n x_{n,t}\}$ given to
    each agent
 1: for each agent $n \in [N]$ do
 2:     Draw $\forall t, (s_{n,t,m})_{m=1}^{N-1} \in \mathcal{U}([0,R]^{N-1})$
 3:     and set $\forall t, s_{n,t,N} := x_{n,t} - \sum_{m=1}^{N-1} s_{n,t,m} \bmod R$
 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 $\forall t, \sigma_{n,t} = \sum_{m\in[N]} s_{m,t,n} \bmod R$
 8:     Send $(\sigma_{n,t})_{t\in[T]}$ to operator
 9: done
10: Operator computes $\boldsymbol{S} = \sum_{n\in[N]} \boldsymbol{\sigma}_n \bmod R$ (and broadcasts it to agents)
```

information about any of the agents, and sends it to the operator (line 8). The operator can finally compute the quantity $\boldsymbol{S} = \boldsymbol{x}^\top \mathbb{1}_N = \boldsymbol{\sigma}^\top \mathbb{1}_N$.

*Remark* 3.11. A drawback of the proposed SMC summation (Algorithm 3.2) is that each agent needs to exchange with all other nodes; therefore the communication overhead for each iteration and each node is $\mathcal{O}(N\ell)$, where $\ell$ is an upper bound on the length of the message (e.g., number of bits encoding constant $R$). Splitting the messages with the other $N-1$ nodes is required to obtain privacy guarantees against collusion of strictly less than $N-1$ nodes (see subsection 3.3). However, as detailed in [4, Protocol I] the SMC summation can be adapted by splitting secret numbers among only $k \in \{1, \dots, N\}$ members. In that case, the communication overhead would be reduced to $\mathcal{O}(k\ell)$, but privacy would only be protected against collusion of less than $k-1$ agents. This choice has to be made as a trade-off between computation overhead and privacy guarantees.

We sum up in Algorithm 3.3 below the procedure of generating a new constraint as stated in Theorem 3.3 from the output of APM in finite time (see Proposition 3.8) and in a privacy-preserving way using SMC.

To choose $B$ and $\varepsilon_{\text{cvg}}$ satisfying the conditions of Proposition 3.8 a priori, one has to know the value of $\underline{\nu}$. Although a conservative lower bound could be obtained by Diophantine arguments if we consider rationals as inputs of the algorithm, in practice it is easier and more efficient to proceed in an iterative manner for the value of $\varepsilon_{\text{cvg}}$. Indeed, one can start with $\varepsilon_{\text{cvg}}$ arbitrarily large so that APM will converge quickly,

**Algorithm 3.3.** Nonintrusive APM (NI-APM)

```
Require: Start with $\boldsymbol{y}^{(0)}$, $k=1$, $\varepsilon_{\text{cvg}}, \varepsilon_{\text{dis}}$, norm $\|.\|$ on $\mathbb{R}^{NT}$
 1: repeat
 2:     for each agent $n \in [N]$ do
 3:         $\boldsymbol{x}_n^{(k)} \leftarrow P_{\mathcal{X}_n}(\boldsymbol{y}_n^{(k-1)})$
 4:     done
 5:     Operator obtains $\boldsymbol{S}^{(k)} \leftarrow$SMCA($\boldsymbol{x}^{(k)}$) (cf. Algorithm 3.2)
 6:     and sends $\boldsymbol{\nu}^{(k)} := \frac{1}{N}(\boldsymbol{p} - \boldsymbol{S}^{(k)}) \in \mathbb{R}^T$ to agents $[N]$
 7:     for each agent $n \in [N]$ do
 8:         Compute $\boldsymbol{y}_n^{(k)} \leftarrow \boldsymbol{x}_n^{(k)} + \boldsymbol{\nu}^{(k)}$ ▷ from (3.4) and (3.12), $\boldsymbol{y}^{(k)} = P_{\mathcal{Y}_{\boldsymbol{p}}}(\boldsymbol{x}^{(k)})$
 9:     done
10:     $k \leftarrow k+1$
11: until $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\| < \varepsilon_{\text{cvg}}$
12: if $\|\boldsymbol{x}^{(k)} - \boldsymbol{y}^{(k)}\| \leqslant \varepsilon_{\text{dis}}$ then ▷ found a $\varepsilon_{\text{dis}}$-solution of the disaggregation problem
13:     Each agent adopts profile $\boldsymbol{x}_n^{(k)}$
14:     return Disag ← True
15: else ▷ have to find a valid constraint violated by $\boldsymbol{p}$
16:     Operator computes $\mathcal{T} \leftarrow \{t \in [T] \mid B\varepsilon_{\text{cvg}} < \nu_t^{(k)}\}$
17:     Operator computes $A_{\mathcal{T}} \leftarrow$ SMCA( $(\boldsymbol{x}_t^{(k)})_{t \in \mathcal{T}}$)
18:     if $A_{\mathcal{T}} - \sum_{t \in \mathcal{T}} \boldsymbol{p}_t < 0$ then
19:         return Disag ← False, $\mathcal{T}$, $A_{\mathcal{T}}$
20:     else ▷ need to run APM with higher precision
21:         Return to Line 1 with $\varepsilon_{\text{cvg}} \leftarrow \varepsilon_{\text{cvg}}/2$
22:     end
23: end
```

and then check if the cut obtained is violated by the current value of $\boldsymbol{p}$ (subsection 3.2): if it is not the case, we can continue the iterations of APM with convergence precision improved to $\varepsilon_{\text{cvg}}/2$ (subsection 3.2). Proposition 3.8 ensures that this loop terminates in finite time.

The parameter $\varepsilon_{\text{dis}} > 0$ (line 12 of Algorithm 3.3) has to be chosen a priori by the operator, depending on the precision required. In general in APM, $\boldsymbol{x}^\infty = \boldsymbol{y}^\infty$ will only be achieved in infinite time, so choosing $\varepsilon_{\text{dis}}$ strictly positive is required.

We end this section by summarizing in Algorithm 3.4 the global iterative procedure to compute an optimal and disaggregable resource allocation $\boldsymbol{p}$, solution of the initial problem (1.1), using iteratively NI-APM (Algorithm 3.3) and adding constraints as stated in Theorem 3.3.

Algorithm 3.4 iteratively calls NI-APM (Algorithm 3.3) and in case disaggregation is not possible (line 11), a new constraint is added (line 13), obtained from the quantity $A_{\mathcal{T}}$ defined in (3.7), to the feasible set of resource allocations $\mathcal{P}^{(s)}$ in problem (3.1). This constraint is an inequality on $\boldsymbol{p}$ and thus does not reveal significant individual information to the operator. The algorithm stops when disaggregation is possible (line 9). The termination of Algorithm 3.4 is ensured by the following property and the form of the constraints added (3.6).

PROPOSITION 3.12. *Algorithm* 3.4 *stops after a finite number of iterations, as at most* $2^T - 2$ *constraints (line* 13*) can be added to the master problem (line* 2*).*

The following Proposition 3.13 shows the correctness of our Algorithm 3.4.

PROPOSITION 3.13. *Let $B$ and $\varepsilon_{\text{cvg}}$ satisfy the conditions of Proposition* 3.8 *and* 3.9. *Then,*

(i) *if the problem* (1.1) *has no solution, Algorithm* 3.4 *exits at line* 4 *after at most* $2^T - 2$ *iterations;*

(ii) *otherwise, Algorithm* 3.4 *computes, after at most $s \leqslant 2^T - 2$ iterations, an aggregate solution $\boldsymbol{p}^{(s)} \in \mathcal{P}$, associated to individual profiles $(\boldsymbol{x}^*)_n =$ NI-APM$(\boldsymbol{p}^{(s)})$ such that*

**Algorithm 3.4.** Nonintrusive optimal disaggregation

```
Require: $s = 0$ , $\mathcal{P}^{(0)} = \mathcal{P}$ ; DISAG= FALSE
 1: while Not DISAG do
 2:     Solve $\min_{\boldsymbol{p} \in \mathcal{P}^{(s)}} f(\boldsymbol{p})$
 3:     if problem infeasible then
 4:         Exit
 5:     else
 6:         Compute $\boldsymbol{p}^{(s)} = \arg\min_{\boldsymbol{p} \in \mathcal{P}^{(s)}} f(\boldsymbol{p})$
 7:     end
 8:     DISAG← NI-APM($\boldsymbol{p}^{(s)}$)  (Algorithm 3.3)
 9:     if  DISAG then
10:         Operator adopts $\boldsymbol{p}^{(s)}$
11:     else
12:         Obtain $\mathcal{T}^{(s)}, A_{\mathcal{T}}^{(s)}$ from NI-APM($\boldsymbol{p}^{(s)}$)
13:         $\mathcal{P}^{(s+1)} \leftarrow \mathcal{P}^{(s)} \cap \{\boldsymbol{p} | \sum_{t \in \mathcal{T}^{(s)}} p_t \leqslant A_{\mathcal{T}}^{(s)}\}$
14:     end
15:     $s \leftarrow s + 1$
16: done
```

$$\boldsymbol{p}^{(s)} \in \mathcal{P}, \quad \forall n \in [N], \boldsymbol{x}_n^* \in \mathcal{X}_n, \quad \| \textstyle\sum_{n \in [N]} \boldsymbol{x}_n^* - \boldsymbol{p}^{(s)} \| \leqslant \varepsilon_{\text{dis}}, \quad \textit{and } f(\boldsymbol{p}^{(s)}) \leqslant f^* ,$$

*where $f^*$ is the optimal value of problem* (1.1).

*Proof.* The proof is immediate from Theorem 3.3, Proposition 3.8, and Proposition 3.9. □

*Remark* 3.14. The upper bound on the number of constraints added has no dependence on $N$ because, as stated in (2.6), once a subset of $[T]$ is chosen, the constraint we add in the algorithm is found by taking the minimum over the subsets of $[N]$.

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. Intuitively, we will only add constraints "supporting" the optimal allocation $\boldsymbol{p}$. Thus, Algorithm 3.4 is a method which enables the operator to compute a resource allocation $\boldsymbol{p}$ and the $N$ agents to adopt profiles $(\boldsymbol{x}_n)_n$, such that $(\boldsymbol{x}, \boldsymbol{p})$ solves the global problem (1.1), and the method ensures that both agent constraints (upper bounds $\overline{\boldsymbol{x}}_n$, lower bounds $\underline{\boldsymbol{x}}_n$, demand $E_n$) and disaggregate (individual) profile $\boldsymbol{x}_n$ (as well as the iterates $(\boldsymbol{x}^{(k)})_k$ and $(\boldsymbol{y}^{(k)})_k$ in NI-APM) are kept confidential by agent $n$ and cannot be induced by a third party (either the operator or any other agent $m \neq n$).

*Remark* 3.15. A natural approach to address problem (1.1) in a distributed way, assuming that both the cost function $\boldsymbol{p} \mapsto f(\boldsymbol{p})$ and the feasibility set $\mathcal{P}$ are convex, is to rely on Lagrangian based decomposition techniques. Examples of such methods are dual subgradient methods [9, Chapter 6], the auxiliary problem principle method [14], ADMM [19], [44], or bundle methods [30].

It is conceivable to develop a privacy-preserving implementation of those techniques, where Lagrangian multipliers associated to the (relaxed) aggregation constraint $\sum_n \boldsymbol{x}_n = \boldsymbol{p}$ would be updated using the SMC technique as described in Algorithm 3.2. However, those techniques usually ask for a strong convexity hypothesis: for instance, in ADMM, in order to keep the decomposition structure in agent by agent, a possibility is to use multiblocks ADMM with $N+1$ blocks ($N$ agents and the operator), which is known to converge in the condition that strong convexity of the cost function in at least $N$ of the $N+1$ variables holds [16]. The study of privacy-preserving implementations of Lagrangian decomposition methods is left for further work.

The advantage of Algorithm 3.4 proposed in this paper is that convergence is ensured (see Proposition 3.12) even if the cost function $\boldsymbol{p} \mapsto f(\boldsymbol{p})$ and the feasibility set $\mathcal{P}$ are not convex, which is the case in many practical situations (see subsection 5.2).

### 3.3. Privacy guarantees.

We next analyze the protection of privacy of the present optimization algorithm, considering different situations. For this analysis, we consider the paradigm of *honest, semihonest*, and *malicious* agents. The term *semihonest* (also called *honest but curious* [1], [37]) refers to a user who obeys the protocol but wishes to use the data obtained through the algorithm to infer private information. In contrast, a *malicious* user may even disobey the protocol.

*Malicious operator and honest agents.* As a first step, we consider an ideal model, in which we only analyze the loss of privacy caused by the algorithm, through communications between honest agents and a malicious (or semihonest) central operator.

We neglect, in this situation, the information leaks induced by interagent communications through the secure multiparty protocol.

Through Algorithm 3.4, the operator obtains the following information for each iteration $s \in \{1, \ldots, s^{\text{cv}}\}$ (where $s^{\text{cv}} < \infty$ is the number of iterations needed for convergence):

- the set $\mathcal{T}^{(s)}$ and the scalar $A^{(s)}_{\mathcal{T}}$ defining the cut;
- whenever NI-APM is called, the series $(S^{(s,k)})_{k=1}^{K_s}$ of the aggregated projections.

Let us denote by $\mathbb{I}^{\mathrm{op}} := \big\{(S^{(0,k)})_{k=1}^{K_0}, \mathcal{T}^{(0)}, A^{(0)}_{\mathcal{T}}, \dots, (S^{(s^{\mathrm{cv}},k)})_{k=1}^{K_{s^{\mathrm{cv}}}}, \mathcal{T}^{(s^{\mathrm{cv}})}, A^{(s^{\mathrm{cv}})}_{\mathcal{T}}\big\}$ the information obtained by the operator. The next proposition implies that, in this ideal model, the operator cannot discern the individual profiles or individual constraints.

PROPOSITION 3.16. *Consider two different instances of the resource allocation problem, differing only by a permutation of the agents, meaning that* $\forall n \in [N]$, $\mathcal{X}_n$ *is replaced by* $\mathcal{X}_{\sigma(n)}$, *for some permutation* $\sigma$ *of the elements of* $[N]$. *Then, the information* $\mathbb{I}^{op}$ *gotten by the operator (as well as the information* $(\boldsymbol{\nu}^{(k)})_k$ *sent by the operator to agents in Algorithm* 3.3*), and the sequence of cuts added in the master problem (Algorithm* 3.4*), are precisely the same in both instances.*

*Proof.* The only information transmitted by the agents to the central operator consists of the aggregate profiles $\boldsymbol{S}^{(k)} = \mathrm{SMCA}(\boldsymbol{x}^{(k)})$ and the sums $A_{\mathcal{T}}$ (line 5 and line 17 of Algorithm 3.3), for different values of the subset $\mathcal{T}$, corresponding to different Hoffman cuts. Each aggregate profile $\boldsymbol{S}^{(k)}$ is invariant by permutation of the agents. Each sum $A_{\mathcal{T}}$ is of the form $A_{\mathcal{T}} = \sum_{n\in[N],t\in\mathcal{T}} x_{nt}^{(K)}$, where $K$ is the iteration at which the violated inequality is found. Hence it is also invariant by a permutation of the agents. $\square$

To illustrate Proposition 3.16, suppose Alice and Bob live in a small town, in which there is a sports hall with two kinds of lessons every day, boxing, from 12h00 to 13h00, and dancing, from 18h00 to 19h00. Suppose Alice goes to the boxing lesson and that Bob goes to the dancing lesson. Suppose further that Alice and Bob do not consume any electric energy during their lessons. Then, by analyzing the successive aggregates $A_{\mathcal{T}}$, for different values of $\mathcal{T} \subset [T]$, the central operator may deduce that the global consumption decreases at the time of the lessons, implying that one agent boxes whereas another one dances. However, whether it is Alice or Bob who does boxing or dancing remains inaccessible to the operator, owing to the symmetry argument. This protection persists even with a malicious operator. This privacy property is stated formally as follows.

COROLLARY 3.17. *With* $N > 1$, *a malicious (or semihonest) operator using Algorithm* 3.4 *cannot infer the constraints* $\mathcal{X}_n$ *or profile* $\boldsymbol{x}_n$ *of a particular agent* $n \in [N]$ *with probability* 1.

*Remark* 3.18. Even if the identity of users is fully protected, in some specific cases the central operator could infer limited information about constraints or profile that an agent within the population may bear. Indeed, let us consider an illustrative example with $T = 2$, $N = 2$, and constraints parameters $(E_1, \overline{x}_{1,1}, \overline{x}_{1,2}) := (1, 2, 0)$, $(E_2, \overline{x}_{2,1}, \overline{x}_{2,2}) := (3, 1, 2)$, and $\underline{\boldsymbol{x}} := 0$ (all unknown to the operator). The operator knows a priori the aggregate quantities $E_1 + E_2 = 4$, $\overline{\boldsymbol{x}}_1 + \overline{\boldsymbol{x}}_2 = (3, 2)$, $\underline{\boldsymbol{x}}_1 + \underline{\boldsymbol{x}}_2 = (0, 0)$.

Before the algorithm, from the operator viewpoint, any parameter values satisfying the aggregate conditions above are possible. For instance, it is possible that agent 1 has the parameters $(E_1^*, \overline{x}^*_{1,1}, \overline{x}^*_{1,2}, \underline{x}^*_{1,1}, \underline{x}^*_{1,2}) := (3, 2, 2, 0, 0)$.

Now suppose that the first aggregate profile proposed by the operator is $\boldsymbol{p}^{(0)} := (3, 1)$. From Algorithm 3.4, he obtains the cut defined by $\mathcal{T}^{(0)} = \{1\}$ and $A^{(0)}_{\{1\}} = 2$. Thus, assuming the operator knows that $A^{(0)}_{\{1\}}$ is of the form $A^{(0)}_{\{1\}} = \min_{\mathcal{N}\subset[N]}\{\sum_{n\in\mathcal{N}} E_n + \sum_{t\in\mathcal{T},n\notin\mathcal{N}} \overline{x}_{n,t} - \sum_{t\notin\mathcal{T},n\in\mathcal{N}} \underline{x}_{n,t}\}$ for a subset $\mathcal{N}$, he infers the following conditions on the parameters:

$$\begin{array}{lcl} E_1+\overline{x}_{2,1}-\underline{x}_{1,2}=2 & & E_2+\overline{x}_{1,1}-\underline{x}_{2,2}=2 \\ \text{and } E_2+\overline{x}_{1,1}-\underline{x}_{2,2}\geqslant 2 & \text{or} & \text{and } E_1+\overline{x}_{2,1}-\underline{x}_{1,2}\geqslant 2\ . \end{array} \tag{3.13}$$

With these conditions, the parameter values $(E_1^*,\overline{x}_{1,1}^*,\overline{x}_{1,2}^*,\underline{x}_{1,1}^*,\underline{x}_{1,2}^*)$ are no longer possible for agent 1 (nor for agent 2 as the information the operator obtains is necessarily symmetric). Indeed, from the initial aggregate conditions, this implies that $(E_2^*,\overline{x}_{2,1}^*,\overline{x}_{2,2}^*,\underline{x}_{2,1}^*,\underline{x}_{2,2}^*) := (1,1,0,0,0)$, which is incompatible with (3.13).

It seems difficult to have an algorithm where the operator will not learn any information on the distribution of parameters. The kind of information leak illustrated in Remark 3.18 is also observed in other privacy-preserving algorithms. One such example is the privacy-preserving consensus method to compute an average value of numbers secretly owned by agents of [37]. Even if the secret number of a particular agent cannot be learned by another agent (see [37, Theorem 3]), at the end of the procedure, each agent has obtained new information on the joint distribution of initial numbers of other agents, as each agent obtains the average value Avg[0] of those numbers.

Fortunately, in most cases, an operator using Algorithm 3.4 will only learn very limited information on the distribution of constraints and profiles of users: for this, it is instructive to estimate the information leak caused by the algorithm by comparing the information sent to the central operator with the one needed to encode the instance. Suppose, for simplicity, that all the numbers manipulated by the algorithm (in particular the bounds $\underline{x}_{n,t}$ and $\bar{x}_{n,t}$ are encoded by fixed (say, double) precision numbers. Then, the total number of bits needed to encode the instance is $O(NT)$. However, it follows from Proposition 3.13 that the number of bits received by a malicious central operator is $O(2^T)$, hence, in the regime $N \gg 2^T$ (large number of agents), the information leak is asymptotically negligible. Moreover, we give in subsection 5.2 a real example with $N = 2^8$ agents, $T = 24$, in which the algorithm terminates after 194 iterations. Here, the information leak consists of 194 doubles, to be compared with the encoding size of the instance, $NT + N = 12544$ doubles.

*Semihonest agents.* Let us now consider agents that are only semihonest, meaning that they still obey the protocol, but wish to exploit the data received in the algorithm to infer information about the other users.

A semihonest agent (or an external eavesdropper) aiming to learn the profile $\boldsymbol{x}_n$ of agent $n$ has to intercept all the communications between $n$ and all $N-1$ other agents (to learn $(s_{n,t,m})_{m\neq n}$ and $(s_{m,t,n})_{m\neq n}$) and to the operator (to learn $\sigma_n$) during the SMC protocol in order to succeed (otherwise, this semihonest agent has access just to random numbers). Besides, if we think of collusion of semihonest agents aiming to learn the profile of a specific agent, we can observe that the collusion must involve $N-1$ agents (all except one) to succeed in learning anything.

Indeed, in the SMC summation protocol (Algorithm 3.2), the messages $s_{n,t,j}$ sent by agent $n$ to agent $j$, encoding shares of the consumption $x_{nt}$ of agent $n$ at different times, are all uniformly distributed random variables, because the sum of independent uniform random variables modulo $R$ is still uniform. Similarly, the messages $\sigma_{n,t}$ sent by agent $n$ to the central operator are uniformly distributed random variables. Furthermore, there are no correlations between the random variables received by an agent at different iterations. This entails that, even if some agents keep the same consumption profile over time, a semihonest agent cannot learn the profiles of the other agents by comparing the information received during different iterations of the SMC protocol.

We refer the reader to [4, Protocol I], where a finer variant of SMC (secure split protocol), splitting secret numbers only among $k < n$ randomly chosen players, is analyzed.

Alternative SMC methods to the proposed protocol Algorithm 3.2 exist to compute the sum $\sum_{n\in[N]} y_n$, while keeping $y_n$ secret to user $n$, for instance [13]. Other techniques could also be considered such as consensus-based aggregation algorithms [22].

From the above paragraph, one deduces that a collusion of less than $(N-1)$ semihonest agents cannot obtain more information than the operator from an execution of Algorithm 3.4: indeed, the only additional information that the collusion obtains is composed of the sequences $\{\boldsymbol{\nu}^{(k)}\}$ obtained from each execution of Algorithm 3.3. Thus, the symmetry argument given in Proposition 3.16 applies to give a privacy guarantee similar to Corollary 3.17.

COROLLARY 3.19. *Using Algorithm* 3.4 *with* $N > 1$*, a collusion of less than* $(N-1)$ *malicious (or semihonest) agents cannot infer the constraints* $\mathcal{X}_n$ *or profile* $\boldsymbol{x}_n$ *of a particular agent* $n$ *(which is not in the collusion) with probability* 1.

*Malicious agents and robustness.* The proposed procedure is not robust against malicious agents, i.e., agents lying about their consumption to jeopardize the system. Indeed, if agents lie about their consumptions or feasibility constraints, Algorithm 3.4 will not converge to the global optimum. However, we still have the privacy guarantee given in Corollary 3.17: malicious agents lying to the operator in order to learn information about other agents can only learn *global* information and will not be able to learn *individual* information (profile or constraints) about a specific agent.

A more detailed privacy analysis of our method and of its possible refinements can be an avenue for further work.

In the next section, we focus on the convergence rate of APM in the particular case of transportation constraints and give an explicit bound on the geometric rate stated in Theorem 3.2.

**3.4. Complexity analysis of APM in the transportation case.** In this section we analyze the speed of convergence of the APM described in Algorithm 3.1 on the sets $\mathcal{X}$ and $\mathcal{Y}_{\boldsymbol{p}}$ defined in section 2.

A general result in [12] gives an upper bound of the sequences generated by APM on $\mathcal{X}$ and $\mathcal{Y}$ if these two sets are semialgebraic. In particular, it establishes the geometric convergence for polyhedral sets. However, as stated in [34], given two particular polyhedral sets $\mathcal{X}$ and $\mathcal{Y}$, it is not straightforward to deduce an explicit rate of convergence from their result.

The authors in [34] established in a particular case a geometric convergence with an explicit upper bound on the convergence rate. They consider APM on two sets $P$ and $Q$, where $P$ is a linear subspace and $Q$ is a product of base polytopes of submodular functions.

In this section, we also establish an explicit upper bound on the convergence rate of APM in the transportation case, that is, with $\mathcal{X}$ and $\mathcal{Y}_{\boldsymbol{p}}$ defined in (2.2) and (1.2b).

THEOREM 3.20. *For the two sets* $\mathcal{X}$ *and* $\mathcal{Y}_{\boldsymbol{p}}$*, the sequence of alternate projections converges to* $\boldsymbol{x}^* \in \mathcal{X}$*,* $\boldsymbol{y}^* \in \mathcal{X}^P$ *satisfying* $\|\boldsymbol{x}^* - \boldsymbol{y}^*\| = \inf_{\boldsymbol{x}\in\mathcal{X},\boldsymbol{y}\in\mathcal{Y}_{\boldsymbol{p}}} \|\boldsymbol{x}-\boldsymbol{y}\|$*, at the geometric rate*

$$\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^*\| \leqslant 2\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\| \times \left(1 - \tfrac{4}{N(T+1)^2(T-1)}\right)^k,$$

*and the analogous inequalities hold for* $(\boldsymbol{y}^{(k)})_k$.

For the remainder of this section, we will just use $\mathcal{Y}$ to denote $\mathcal{Y}_{\boldsymbol{p}}$, as $\boldsymbol{p}$ remains fixed during APM. For the result stated in Theorem 3.20 above, we use several lemmas from [34].

*Proof.* First, we use the fact stated in [34] that APM on subspaces $U$ and $V$ converge with geometric rate $c_F(U,V)^2$, where the rate is given by the square of the cosine of the Friedrichs angle between $U$ and $V$, given by

$$c_F(U,V) = \sup\left\{u^T v \mid u \in U \cap (U\cap V)^{\perp}, v \in V \cap (U \cap V)^{\perp}, \|u\| \leqslant 1, \|v\| \leqslant 1\right\}.$$

An intuitive generalization of this result for polyhedra $\mathcal{X}$ and $\mathcal{Y}$, considering all affine subspaces supporting the faces of $\mathcal{X}$ and $\mathcal{Y}$, is given in [34].

LEMMA 3.21 (see [34]). *For APM on polyhedra $\mathcal{X}$ and $\mathcal{Y}$ in $\mathbb{R}^D$, the convergence is geometric with rate bounded by the square of the maximal cosine of Friedrichs angle between subspaces supporting faces of $\mathcal{X}$ and $\mathcal{Y}$:*

$$\max_{\boldsymbol{x},\boldsymbol{y}} c_F\big(\operatorname{aff}_0(\mathcal{X}_{\boldsymbol{x}}), \operatorname{aff}_0(\mathcal{Y}_{\boldsymbol{y}})\big), \tag{3.14}$$

*where, for any $\boldsymbol{x} \in \mathbb{R}^D$, $\mathcal{X}_{\boldsymbol{x}} := \arg\max_{\boldsymbol{v}\in\mathcal{X}} \boldsymbol{x}^\top \boldsymbol{v}$ is the face of $\mathcal{X}$ generated by direction $\boldsymbol{x}$ and $\operatorname{aff}_0(C) = \operatorname{aff}(C) - \boldsymbol{c}$ for some $\boldsymbol{c} \in C$ denotes the subspace supporting the affine hull of $C$, for $C = \mathcal{X}_{\boldsymbol{x}}$ or $C = \mathcal{Y}_{\boldsymbol{y}}$.*

In the rest of the proof, we bound the quantity (3.14) for our polyhedra $\mathcal{X}$ and $\mathcal{Y}$.

For this, we use the space $\mathbb{R}^{NT} = \mathbb{R}^T \times \cdots \times \mathbb{R}^T$, where the $(n-1)T+1$ to $nT$ entries correspond to the profile of agent $n$, for $1 \leqslant n \leqslant N$. As in [34], we use a result connecting angles between subspaces and the eigenvalues of matrices giving the directions of these spaces.

LEMMA 3.22 (see [34]). *If $A$ and $B$ are matrices with orthonormal rows with the same number of columns, then*

- *if all the singular values of $AB^\top$ are equal to one, then $c_F\big(\operatorname{Ker} A,\ \operatorname{Ker} B\big) = 0$;*
- *otherwise, $c_F\big(\operatorname{Ker} A,\ \operatorname{Ker} B\big)$ is equal to the largest singular value of $AB^\top$ among those that are smaller than one.*

We are left with finding a matrix representation of the faces of polyhedra $\mathcal{X}$ and $\mathcal{Y}$ and, then, bounding the corresponding singular values.

In our case, the polyhedra $\mathcal{Y}$ is an affine subspace $\mathcal{Y} = \{\boldsymbol{x} \in \mathbb{R}^{NT} \mid A\boldsymbol{x} = \sqrt{N}^{-1}\boldsymbol{p}\}$ where

$$A := \sqrt{N}^{-1} J_{1,N} \otimes I_T,$$

where $\otimes$ denotes the Kronecker product. The matrix $A$ has orthonormal rows and the linear subspace associated to $\mathcal{Y}$ is equal to $\operatorname{Ker}(A)$.

Obtaining a matrix representation of the faces of $\mathcal{X}$ is more complex. The faces of $\mathcal{X}$ are obtained by considering, for each $n \in [N]$, subsets of the time periods that are at lower or upper bound (resp., $\underline{\mathcal{T}}_n$ and $\overline{\mathcal{T}}_n$, with $\underline{\mathcal{T}}_n \cap \overline{\mathcal{T}}_n = \emptyset$). Considering a collection of such subsets, a face of $\mathcal{X}$ can be written as

$$\begin{aligned}\mathcal{A}_{(\overline{\mathcal{T}}_n, \underline{\mathcal{T}}_n)_n} := \Big\{(\boldsymbol{x})_{n,t} \mid & \forall n,\ \textstyle\sum_t x_{n,t} = E_n \text{ and} \\ & \forall t \in \underline{\mathcal{T}}_n, x_{n,t} = \underline{x}_{n,t}, \text{ and } \forall t \in \overline{\mathcal{T}}_n, x_{n,t} = \overline{x}_{n,t}\Big\}.\end{aligned}$$

For some particular collection of subsets $(\overline{\mathcal{T}}_n, \underline{\mathcal{T}}_n)_n$, the set $\mathcal{A}_{(\overline{\mathcal{T}}_n, \underline{\mathcal{T}}_n)_n}$ might be empty. The linear subspace associated to $\mathcal{A}_{(\overline{\mathcal{T}}_n, \underline{\mathcal{T}}_n)_n}$ is given by $\{\boldsymbol{x} \in \mathbb{R}^{NT} | \; B\boldsymbol{x} = 0\} = \mathrm{Ker}(B)$, where the $N$ first rows of $B$, corresponding to the constraints $\sum_t x_{n,t} = E_n$, are given before orthonormalization by

$$\sqrt{T}^{-1} I_N \otimes J_{1,T},$$

and the matrix $B$ has $b := \sum_n |\mathcal{T}_n|$ more rows, where $\mathcal{T}_n := \underline{\mathcal{T}_n} \cup \overline{\mathcal{T}_n}$, corresponding to the saturated bounds. Each of this row is given by the unit vector $\boldsymbol{e}_{n,t}^\top \in \mathbb{R}^{NT}$ for $n \in [N]$, $t \in \mathcal{T}_n$, which already gives an orthonormalized family of (unit) vectors. Therefore, a simple orthonormalized matrix $B \in \mathcal{M}_{N+b,NT}(\mathbb{R})$ giving the direction of $\mathcal{A}_{(\overline{\mathcal{T}}_n, \underline{\mathcal{T}}_n)_n}$ is given by

$$B := \Big( \mathrm{diag} \left( \sqrt{T - |\mathcal{T}_1|}^{-1} \mathbf{1}_{\mathcal{T}_1^c}^\top, \dots, \sqrt{T - |\mathcal{T}_N|}^{-1} \mathbf{1}_{\mathcal{T}_N^c}^\top \right) \quad \Big| \quad \mathrm{diag}\left( B_{\mathcal{T}_1}, \dots, B_{\mathcal{T}_N} \right) \Big)^\top ,$$

where $\mathbf{1}_{\mathcal{T}_n^c} \in \mathbb{R}^T$ is the vector where the indices in $\mathcal{T}_n^c$ are equal to 1 and 0 otherwise, and $B_{\mathcal{T}_n} := \sum_{\substack{1 \leqslant k \leqslant |\mathcal{T}_n| \\ \mathcal{T}_n = \{t_1, \dots, t_{|\mathcal{T}_n|}\}}} E_{kt_k}$ is the matrix $|\mathcal{T}_n| \times T$ with indices of $\mathcal{T}_n$. We obtain the double product:

$$\begin{aligned} \left(AB^\top\right) \left(BA^\top\right) &= \frac{1}{N} \left( \textstyle\sum_n \dfrac{\mathbb{1}_{k \notin \mathcal{T}_n \wedge \ell \notin \mathcal{T}_n}}{T - |\mathcal{T}_n|} \right)_{1 \leqslant k, \ell \leqslant T} + \frac{1}{N} \sum_n B_{\mathcal{T}_n}^\top B_{\mathcal{T}_n} \\ &= \frac{1}{N} \left( \textstyle\sum_n \dfrac{\mathbb{1}_{\{k,\ell\} \subset \mathcal{T}_n^c}}{T - |\mathcal{T}_n|} \right)_{1 \leqslant k, \ell \leqslant T} + \frac{1}{N} \sum_{1 \leqslant t \leqslant T} \left( \textstyle\sum_n \mathbb{1}_{t \in \mathcal{T}_n} \right) E_{t,t} \ . \end{aligned}$$

We observe that

- if $t_0 \in \bigcap_{i=1}^N \mathcal{T}_n$, then $\boldsymbol{e}_{t_0}$ is an eigenvector associated to eigenvalue $\lambda_{t_0} = 1$;
- the vector $\mathbf{1}_{\bar{\mathcal{T}}} := (\mathbb{1}_{t \notin \cap_n \mathcal{T}_n})_{t \in [T]} \in \mathbb{R}^T$, where $\bar{\mathcal{T}} := \cup_n \mathcal{T}_n^c$, is an eigenvector associated to eigenvalue $\lambda = 1$. Indeed, if we denote by $\mathcal{N}_\theta = \{n \in [N] | \theta \in \mathcal{T}_n\}$, then $[\mathbf{1}_{\bar{\mathcal{T}}}]_\theta = 1 \Leftrightarrow \mathcal{N}_\theta^c \neq \emptyset$, and for each $\theta \in \bar{\mathcal{T}}$,

$$\begin{aligned} \left[ \left(AB^\top\right) \left(BA^\top\right) \right]_\theta \mathbf{1}_{\bar{\mathcal{T}}} &= \frac{1}{N} \left( \sum_{i \in \mathcal{N}_\theta^c} \sum_t \frac{\mathbb{1}_{t \notin \mathcal{T}_n}}{T - |\mathcal{T}_n|} [\mathbf{1}_{\bar{\mathcal{T}}}]_t + \sum_n \mathbb{1}_{\theta \in \mathcal{T}_n} [\mathbf{1}_{\bar{\mathcal{T}}}]_\theta \right) \\ &= \frac{1}{N} \left( \sum_{i \in \mathcal{N}_\theta^c} \frac{T - |\mathcal{T}_n|}{T - |\mathcal{T}_n|} 1 + \sum_{i \in \mathcal{N}_\theta} 1 \times [\mathbf{1}_{\bar{\mathcal{T}}}]_\theta \right) \\ &= \frac{|\mathcal{N}_\theta^c| + |\mathcal{N}_\theta| [\mathbf{1}_{\bar{\mathcal{T}}}]_\theta}{N} = [\mathbf{1}_{\bar{\mathcal{T}}}]_\theta \ . \end{aligned}$$

To bound the other eigenvalues of the matrix $(AB^\top)(BA^\top)$, we rely on spectral graph theory arguments. Consider the weighted graph $\mathcal{G} = ([T], \mathcal{E})$ whose vertices are the time periods $[T]$ and each edge $(k, \ell) \in [T] \times [T]$ with $k \neq \ell$ has weight $S_{k,\ell} = \frac{1}{N} \sum_n \frac{\mathbb{1}_{\{k,\ell\} \subset \mathcal{T}_n^c}}{T - |\mathcal{T}_n|}$ (if this quantity is zero, then there is no edge between $k$ and $\ell$).

The matrix $P := I_T - (AB^\top)(BA^\top)$ verifies for each $k \in [T]$

$$\sum_{\ell\neq k} -P_{k,\ell} = \sum_{\ell\neq k} \frac{1}{N} \sum_n \frac{\mathbb{1}_{\{k,\ell\}\subset\mathcal{T}_n^c}}{T-|\mathcal{T}_n|} = \frac{1}{N}\sum_n \frac{\mathbb{1}_{k\in\mathcal{T}_n^c}(T-|\mathcal{T}_n|-1)}{T-|\mathcal{T}_n|} \tag{3.15}$$

$$= \frac{1}{N}\sum_n (1-\mathbb{1}_{k\in\mathcal{T}_n}) - \frac{1}{N}\sum_n \frac{\mathbb{1}_{k\in\mathcal{T}_n^c}}{T-|\mathcal{T}_n|} = P_{kk} \ , \tag{3.16}$$

which shows that $P$ is the Laplacian matrix of graph $\mathcal{G}$. As $\mathrm{Sp}(AB^\top BA^\top) = 1 - \mathrm{Sp}(P)$, we want to have a lower bound on the least eigenvalue of $P$ greater than 0, which we denote by $\lambda_1$.

By rearranging the indices of $[T]$ in two blocks $\bar{\mathcal{T}}$ and $\bar{\mathcal{T}}^c$, we observe that $P$ can be written as a block diagonal matrix $P = \mathrm{diag}(P_{\bar{\mathcal{T}}}, 0_{\bar{\mathcal{T}}^c})$. As we are only interested in the positive eigenvalues of $\mathcal{P}$, we can therefore study the linear application associated to $P$ restricted to the subspace $\mathrm{Vect}(e_t)_{t\in\bar{\mathcal{T}}}$.

As $\mathbf{1}_{\bar{\mathcal{T}}}$ is an eigenvector of $P$ associated to $\lambda_0 = 0$, from the minmax theorem, we have

$$\lambda_1 = \min_{u\perp\mathbf{1}_{\bar{\mathcal{T}}}, u\neq 0} \frac{u^\top P u}{u^\top u} \ . \tag{3.17}$$

Let us consider an eigenvector $u$ realizing (3.17). Let $u_{t^*} := \max_t u_t$ and $u_{s^*} := \min_t u_t$ and let $d_{s^*,t^*}$ be the distance between $s^*$ and $t^*$ in $\mathcal{G}$, and let ($s^*$-$t^*$) denote a shortest path from $s^*$ to $t^*$ in $\mathcal{G}$. As $P$ is a Laplacian matrix, we have

$$\begin{aligned} u^\top P u &= \frac{1}{2}\sum_{k,\ell\in\bar{\mathcal{T}}} -P_{k,\ell}(u_k-u_\ell)^2 \geqslant \frac{1}{2}\sum_{\{k,\ell\}\in(s^*\text{-}t^*)} -P_{k,\ell}(u_k-u_\ell)^2 \\ &\geqslant \min_{k,\ell\in(s^*\text{-}t^*)}(-P_{k,\ell})\frac{(u_{t^*}-u_{s^*})^2}{d_{s^*,t^*}} \ , \end{aligned} \tag{3.18}$$

where the last inequality is obtained from the Cauchy–Schwarz inequality.

Let us write the path $(s^*\text{-}t^*) = (t_0, t_1, \dots, t_d)$. As $(s^*\text{-}t^*)$ is a shortest path, for each $k \in \{0, d-1\}$, the edge $(t_k, t_{k+1})$ exists so there exists $n \in [N]$ such that $\{t_k, t_{k+1}\} \subset \mathcal{T}_n^c$. Moreover, for each $n$, we have $\mathcal{T}_n^c \cap \{t_0, \dots, t_{k-1}, t_{k+2}, \dots, t_d\} = \emptyset$; otherwise we could "shortcut" the path $(s^*\text{-}t^*)$. Thus we have $|\mathcal{T}_n| \geqslant d-1$. We obtain

$$-P_{t_k,t_{k+1}} = \frac{1}{N}\sum_n \frac{\mathbb{1}_{\{t_k,t_{k+1}\}\subset\mathcal{T}_n^c}}{T-|\mathcal{T}_n|} \geqslant \frac{1}{N(T-d+1)} \ .$$

On the other hand, we have $(u_{t^*} - u_{s^*}) \geqslant u_{t^*} + \frac{u_{t^*}}{T-1} = \frac{T}{T-1}u_{t^*} \geqslant \frac{T}{(T-1)\sqrt{T}}\|u\|_2$.

Using these bounds and (3.18), we obtain

$$u^\top P u \geqslant \frac{(u_{t^*}-u_{s^*})^2}{N(T-d_{s^*,t^*}+1)d_{s^*,t^*}} \geqslant \frac{4T}{N(T+1)^2(T-1)^2}\|u\|_2^2 \geqslant \frac{4}{N(T+1)^2(T-1)}\|u\|_2^2.$$

Therefore, $\lambda_1 \geqslant \frac{4}{N(T+1)^2(T-1)} := \kappa_{N,T}$ and the greatest singular value lower than one of $(AB^\top)(BA^\top)$ is $1-\kappa_{N,T}$. We conclude by applying successively Lemmas 3.22 and 3.21 to obtain the convergence rate stated in Theorem 3.20. ∎

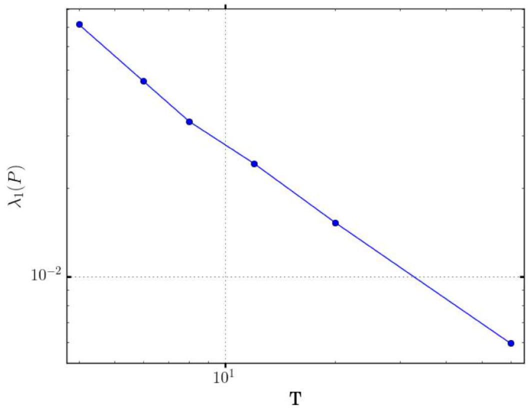


Fig. 3.2. *Evolution of the convergence rate, given as* $\lambda_1(P)$ *(lowest nonzero eigenvalue of* $P$*), with* $N = 6$ *and* $T \in \{4, 6, 8, 12, 20, 60\}$*. The worst convergence rate is evaluated by taking* $100 \times T$ *random draws of the sets* $\mathcal{T}_n \subset [T]$ *for each* $n$*, and evaluating the eigenvalue of the matrix.*

Figure 3.2 shows the convergence rate $\lambda_1(P)$ evaluated numerically for different values of $T$. The slope can be evaluated as around $-0.93$, which indicates that in practice the convergence rate is $\mathcal{O}(T^{-1})$, faster than the upper bound in $\mathcal{O}(T^{-3})$ established in Theorem 3.2.

**4. Generalization to polyhedral agents constraints.** In this section, we extend our results to a more general framework where for each $n \in [N]$, $\mathcal{X}_n$ is an arbitrary polyhedron, instead of having the particular structure given in (2.2). Let us now consider that $(\mathcal{X}_n)_n$ are polyhedra with, for each $n$,

$$\mathcal{X}_n = \{\boldsymbol{x}_n \in \mathbb{R}^T | A_n \boldsymbol{x}_n \leqslant \boldsymbol{b}_n\} \ , \tag{4.1}$$

with $A_n \in \mathcal{M}_{k_n,T}(\mathbb{R})$ with $k_n \in \mathbb{N}$. The disaggregation problem (1.2), *with* $\boldsymbol{p} \in \mathcal{P}$ *fixed*, is written

$$\min_{\boldsymbol{x} \in \mathbb{R}^{NT}} 0 \tag{4.2a}$$

$$\text{s.t. } A_0 \boldsymbol{x} = B\boldsymbol{p} \quad (\boldsymbol{\lambda}_0), \tag{4.2b}$$

$$A_n \boldsymbol{x}_n \leqslant \boldsymbol{b}_n \ \forall n \in [N] \quad (\boldsymbol{\lambda}_n), \tag{4.2c}$$

where $A_0 = J_{1,N} \otimes I_T$, $B = I_T$ (such that (4.2b) corresponds to the aggregation constraint $\sum_n \boldsymbol{x}_n = \boldsymbol{p}$) and $\boldsymbol{\lambda}_0 \in \mathbb{R}^T$, $(\boldsymbol{\lambda}_n)_{n\in[N]} \in \mathbb{R}_+^{\sum_n k_n}$ are the Lagrangian multipliers associated to (4.2b) and (4.2c).

With the polyhedral constraints (4.1), the graph representation of the disaggregation problem, as illustrated in Figure 2.1, is no longer valid. Consequently, one cannot directly apply Hoffman's theorem (Theorem 2.3) to obtain a characterization of disaggregation feasibility by inequalities on $\boldsymbol{p}$. However, using duality theory, Proposition 4.1 below also gives a characterization of disaggregation.

Proposition 4.1. *A profile* $\boldsymbol{p} \in \mathcal{P}$ *is disaggregable iff*

$$\forall \big(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}_1, \dots \boldsymbol{\lambda}_N\big) \in \Lambda, \quad \boldsymbol{\lambda}_0^\top B\boldsymbol{p} + \textstyle\sum_{n\in[N]} \boldsymbol{\lambda}_n^\top \boldsymbol{b}_n \geqslant 0 \ , \tag{4.3}$$

*where*

$$\Lambda := \{\boldsymbol{\lambda}_0 \in \mathbb{R}^{k_0}, \forall n \in [N], \boldsymbol{\lambda}_n \in \mathbb{R}_+^{k_n} \mid A_0^\top \boldsymbol{\lambda}_0 + A^\top (\boldsymbol{\lambda}_n)_n = 0\},$$

*with* $A := \operatorname{diag}(A_n)_{n\in[N]}$.

*Proof.* From strong duality, we have

$$\min_{\boldsymbol{x}\in\mathbb{R}^{NT}} \max_{\boldsymbol{\lambda}_0\in\mathbb{R}^{k_0},\boldsymbol{\lambda}_n\in\mathbb{R}_+^{k_n}} \boldsymbol{\lambda}_0^\top(A_0\boldsymbol{x}-B\boldsymbol{p}) + \sum_n \boldsymbol{\lambda}_n^\top(A_n\boldsymbol{x}_n-\boldsymbol{b}_n) \tag{4.4}$$

$$= \begin{array}{l} \displaystyle\max_{\boldsymbol{\lambda}_0\in\mathbb{R}^{k_0},\boldsymbol{\lambda}_n\in\mathbb{R}_+^{k_n}} -\boldsymbol{\lambda}_0^\top B\boldsymbol{p} - \sum_n \boldsymbol{\lambda}_n^\top \boldsymbol{b}_n \\ \text{s.t. } \boldsymbol{\lambda}_0^\top A_0 + (\boldsymbol{\lambda}_n)_n^\top A = 0. \end{array} \tag{4.5}$$

If the polytope $\mathcal{Y}_{\boldsymbol{p}} \cap \mathcal{X}$ given by the constraints of (4.2) is empty, then there is an infeasibility certificate $\boldsymbol{\lambda}^\top = (\boldsymbol{\lambda}_0^\top \ \boldsymbol{\lambda}_1^\top \dots \boldsymbol{\lambda}_N^\top) \in \mathbb{R}^T \times \prod_n \mathbb{R}_+^{k_n}$ such that

$$\boldsymbol{\lambda}_0^\top A_0 + (\boldsymbol{\lambda}_n^\top)_n A = 0 \text{ and } \boldsymbol{\lambda}_0^\top B\boldsymbol{p} + (\boldsymbol{\lambda}_n)_n^\top \boldsymbol{b} < 0 \ . \tag{4.6}$$

On the other hand, if $\mathcal{Y}_{\boldsymbol{p}} \cap \mathcal{X}$ is nonempty, then a solution to the dual problem (4.5) is bounded, which implies that $\forall \boldsymbol{\lambda} := (\boldsymbol{\lambda}_0, (\boldsymbol{\lambda}_n)_n) \in \Lambda, \boldsymbol{\lambda}_0^\top B\boldsymbol{p} + \sum_n \boldsymbol{\lambda}_n^\top \boldsymbol{b}_n \geqslant 0$. □

As opposed to Hoffman circulation's theorem where disaggregation is characterized by a finite number of inequalities, Proposition 4.1 involves a priori an infinite number of inequalities.

However, we know that the polyhedral cone $\Lambda$ can be represented by a finite number of generators (edges), that is, there exists $\Lambda^* := \{\boldsymbol{\lambda}^{*(1)}, \dots, \boldsymbol{\lambda}^{*(d)}\}$ such that

$$\Lambda = \left\{ \textstyle\sum_{1\leqslant i\leqslant d} \alpha_i \boldsymbol{\lambda}^{*(i)} \mid (\alpha_i)_i \in \mathbb{R}_+^d \right\} . \tag{4.7}$$

Thus, we obtain the following corollary to Proposition 4.1.

Corollary 4.2. *There exists a finite set $\Lambda^* \subset \Lambda$ such that, for any $\boldsymbol{p} \in \mathcal{P}$, $\boldsymbol{p}$ is disaggregable iff*

$$\forall (\boldsymbol{\lambda}_0, (\boldsymbol{\lambda}_n)_n) \in \Lambda^*, \quad \boldsymbol{\lambda}_0 B\boldsymbol{p} + \textstyle\sum_n \boldsymbol{\lambda}_n \boldsymbol{b}_n \geqslant 0 \ . \tag{4.8}$$

*Remark* 4.3. In the transportation case (2.2), we can write each agent constraint in the form $A_n\boldsymbol{x}_n \leqslant \boldsymbol{b}_n$ (the equality $\sum_t x_{n,t} = E_n$ is written as two inequalities), and Hoffman conditions (2.5) can be written in the form (4.8). Moreover, Theorem 3.3 ensures that one possibility for $\Lambda^*$ of Corollary 4.2 is to consider the collection of $2^T$ multipliers corresponding to the subsets $\mathcal{T} \subset [T]$ and $\mathcal{N}$ minimizing (2.6). We skip the details here for brevity.

As in the first part of the paper, we want to use APM to decompose problem (1.1) and, in the case where disaggregation is not possible, use the result of APM to generate an inequality (4.3) violated by the current profile $\boldsymbol{p}$.

In the case of impossible disaggregation, APM converges to the orbit $(\boldsymbol{y}^\infty, \boldsymbol{x}^\infty)$, and $\boldsymbol{\mu} := \boldsymbol{y}^\infty - \boldsymbol{x}^\infty$ defines a separating hyperplane $\bar{\boldsymbol{x}} + \boldsymbol{\mu}^\perp$ , where $\bar{\boldsymbol{x}} = \frac{\boldsymbol{y}^\infty + \boldsymbol{x}^\infty}{2}$, that satisfies, with $a := \bar{\boldsymbol{x}}.\boldsymbol{\mu}$ (note that $\bar{\boldsymbol{x}}$ can be replaced by any $\boldsymbol{y} \in [\boldsymbol{y}^\infty, \boldsymbol{x}^\infty]$),

$$\forall \boldsymbol{x} \in \mathcal{Y}_{\boldsymbol{p}};\ \boldsymbol{\mu}^\top \boldsymbol{x} > a, \qquad\qquad \forall \boldsymbol{x} \in \mathcal{X};\ a > \boldsymbol{\mu}^\top \boldsymbol{x},$$

which give lower bounds on the linear problems (the second one is decomposed because $A$ is a block-diagonal matrix, but it can also be written in one problem):

$$\begin{array}{l} \displaystyle\min_{\boldsymbol{x}\in\mathbb{R}^{NT}} \boldsymbol{\mu}^\top \boldsymbol{x} \\ A_0\boldsymbol{x} = B\boldsymbol{p} \ (\boldsymbol{\lambda}_0) \end{array} = \begin{array}{l} \displaystyle\max_{\boldsymbol{\lambda}_0\in\mathbb{R}^{k_0}} -\boldsymbol{\lambda}_0^\top B\boldsymbol{p} \\ \boldsymbol{\mu} = -A_0^\top \boldsymbol{\lambda}_0 \end{array} \tag{4.9}$$

$$\text{and} \quad \forall n \in [N], \quad \begin{array}{l} \displaystyle\max_{\boldsymbol{x}\in\mathbb{R}^{NT}} \boldsymbol{\mu}_n \boldsymbol{x}_n \\ A_n\boldsymbol{x}_n \leqslant \boldsymbol{b}_n \ (\boldsymbol{\lambda}_n) \end{array} = \begin{array}{l} \min_{\boldsymbol{\lambda}_n\in\mathbb{R}_+^{k_n}} \boldsymbol{b}_n^\top \boldsymbol{\lambda}_n \\ \boldsymbol{\mu}_n = \boldsymbol{\lambda}_n^\top A_n. \end{array} \tag{4.10}$$

**Algorithm 4.1.** Nonintrusive optimal disaggregation with polyhedral constraints

```
Require: Start with $\Lambda^{(0)} = \{\}$, $k = 0$, DISAG= false
while not DISAG do
    get solution $\boldsymbol{p}^{(k)}$ of problem $\min_{\boldsymbol{p}\in\mathcal{P}}\{f(\boldsymbol{p}) \mid \boldsymbol{\lambda}_0^\top B\boldsymbol{p} + \boldsymbol{\lambda}^\top \boldsymbol{b} \geqslant 0,\ \forall \boldsymbol{\lambda} \in \Lambda^{(k)}\}$
    get $\boldsymbol{\mu}^{(k)} = \boldsymbol{y}^\infty - \boldsymbol{x}^\infty \leftarrow \mathrm{APM}(\mathcal{Y}_{\boldsymbol{p}^{(k)}}, \mathcal{X})$
    if $\boldsymbol{\mu}^{(k)} \neq 0$ then
        obtain $\boldsymbol{\lambda}_0^{(k)} \leftarrow \max_{\boldsymbol{\lambda}_0 \in \mathbb{R}^{k_0}}\{-\boldsymbol{\lambda}_0^\top B\boldsymbol{p}^{(k)} \mid \boldsymbol{\mu}^{(k)} = -A_0^\top \boldsymbol{\lambda}_0\}$
        obtain for each $n$, $\boldsymbol{\lambda}_n^{(k)} \leftarrow \max_{\boldsymbol{\lambda}_n \geqslant 0}\{\boldsymbol{b}_n^\top \boldsymbol{\lambda}_n \mid \boldsymbol{\mu}_n^{(k)} = \boldsymbol{\lambda}_n^\top A_n\}$
        add $\Lambda^{(k+1)} = \Lambda^{(k)} \cup \{(\boldsymbol{\lambda}_0^{(k)}, \boldsymbol{\lambda}^{(k)})\}$
    else
        Return DISAG = true, $\boldsymbol{p}^{(k)}$ as optimal solution
    end
    $k \leftarrow k + 1$
done
```

Strong duality on these problems implies that there exist $\boldsymbol{\lambda}_0$ and $\boldsymbol{\lambda}$ such that

$$\boldsymbol{\mu} = -A_0^\top \boldsymbol{\lambda}_0 \text{ and } a < -\boldsymbol{\lambda}_0^\top B\boldsymbol{p}, \qquad \boldsymbol{\mu} = \boldsymbol{\lambda}^\top A \text{ and } a > \boldsymbol{b}^\top \boldsymbol{\lambda}\ . \tag{4.11}$$

Thus, we obtain $(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}) \in \Lambda$ satisfying (4.6), that is, $\boldsymbol{\lambda}_0^\top B\boldsymbol{p} + \boldsymbol{b}^\top \boldsymbol{\lambda} < 0$, and we can use this to add a new valid additional inequality on $\boldsymbol{p}$ of form (4.3) that will change the current profile $\boldsymbol{p}$:

$$\boldsymbol{\lambda}_0^\top B\boldsymbol{p} + \boldsymbol{\lambda}^\top \boldsymbol{b} \geqslant 0. \tag{4.12}$$

In Algorithm 4.1, we summarize the proposed decomposition of problem (1.1). This is a generalization of the decomposition principle used for Algorithm 3.4.

*Remark* 4.4. We use the fact that $\boldsymbol{\mu} = \boldsymbol{y}^\infty - \boldsymbol{x}^\infty$ although, as before, we only have an *approximation* of this quantity. The approximation has to be precise enough to ensure that the solution obtained verifies $\boldsymbol{\lambda}_0^\top B\boldsymbol{p} + \boldsymbol{b}^\top \boldsymbol{\lambda} < 0$. In practice, one can proceed as in the transportation case and Algorithm 3.3 using a large $\varepsilon_{\mathrm{cvg}}$ as stopping criteria in APM, then compute $(\boldsymbol{\lambda}_0, \boldsymbol{\lambda}) \in \Lambda$ and check if $\boldsymbol{\lambda}_0^\top B\boldsymbol{p} + \boldsymbol{b}^\top \boldsymbol{\lambda} < 0$. If this is not the case, restart with $\varepsilon_{\mathrm{cvg}} = \varepsilon_{\mathrm{cvg}}/2$.

*Remark* 4.5. When $\mathcal{Y}_{\boldsymbol{p}} = \{\boldsymbol{x} \in \mathbb{R}^{NT} | A_0\boldsymbol{x} = B_0\boldsymbol{p}\} = \{\boldsymbol{x} | \sum_n \boldsymbol{x}_n = \boldsymbol{p}\}$, we can obtain a nonintrusive version of APM on $\mathcal{Y}_{\boldsymbol{p}}$ and $\mathcal{X}$, similar to Algorithm 3.3. In this case, (4.11) ensures that we have $\boldsymbol{\mu}_{n,t} = -[\boldsymbol{\lambda}_0]_t$ for each $n \in [N]$, and $\boldsymbol{\lambda}_0$ is fixed by $\boldsymbol{\mu}$. The only difference with the transportation case for a nonintrusive APM in the general polyhedral case is in the way of computing the valid constraint violated by $\boldsymbol{p}$. Thus, lines 16 to 19 of Algorithm 3.3 have to be replaced by Algorithm 4.2.

*Remark* 4.6 (link with Benders' decomposition). In this generalized case, we obtain an algorithm related to Benders' decomposition [8] (recall that in our specific case (4.2), the cost function does not involve the variable $\boldsymbol{x}$ but only variable $\boldsymbol{p}$).

The difference between the proposed Algorithm 4.1 and Benders' decomposition lies in the way of generating the new cut. Benders' decomposition would directly solve the dual problem (4.5), $\max_{\boldsymbol{\lambda}}\{-\boldsymbol{\lambda}_0^\top B_0\boldsymbol{p} - (\boldsymbol{\lambda}_n)_n^\top \boldsymbol{b} \mid \boldsymbol{\lambda}_0 A_0 + (\boldsymbol{\lambda}_n)_n A = 0\}$, and obtain a cut if it is unbounded. However, this problem involves the constraints of all users (through $A$ and $\boldsymbol{b}$), and it is not straightforward to obtain a method to solve this problem in a decentralized and efficient way.

**Algorithm 4.2.** Modification of lines 16–19 of Algorithm 3.3 for NI-APM with polyhedral constraints

```
16: for each agent $n \in [N]$ do
17:     compute $M_n$ optimal value of (4.10).
18: done
19: Operator computes $M \leftarrow \mathrm{SMCA}((M_n)_n)$
20: if $-\boldsymbol{\nu}.\boldsymbol{p} + M < 0$ then
21:     return Disag $\leftarrow$ False, $-\boldsymbol{\nu}, M$
```

## 5. Numerical examples.

**5.1. An illustrative example with T=4.** In this section we illustrate the iterations of the method proposed in this paper on an example with $T = 4$ and $N = 3$. Assuming that we have to satisfy the aggregate constraint $\sum_t p_t = \sum_n E_n$, we can use the projections on this affine space of solutions of master problems $(\boldsymbol{p}^{(s)})_s$ to visualize them in dimension 3.

One can wonder if, in the transportation case, applying Algorithm 3.4 or Algorithm 4.1 will always lead to the same cuts and solutions: the answer is no, as shown by the instance considered in this section, for which Algorithm 3.4 converges in three iterations and Algorithm 4.1 needs four iterations.

We consider the problem (1.1) with agents constraints (2.2) with parameters $\underline{\boldsymbol{x}} := 0$ and

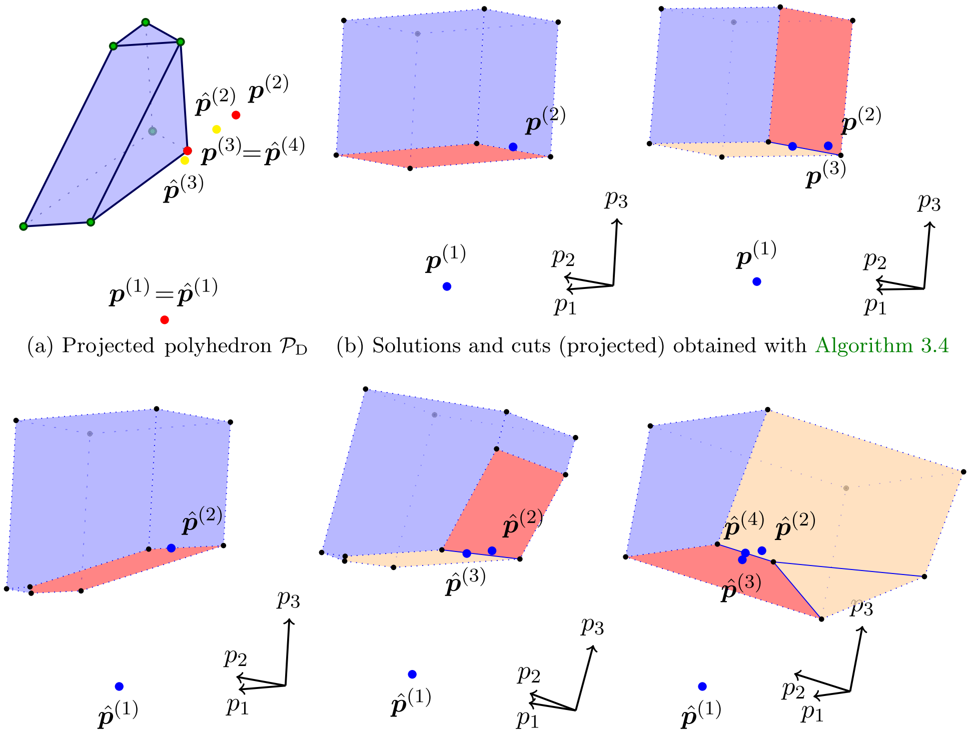
(5.1)
$$\overline{\boldsymbol{x}} := \begin{array}{ll} [0.8, 0.2, 0.7, 0.1], & E_1 = 1.8, \\ [0.5, 0.1, 0.3, 0.6], & E_2 = 0.4, \\ [0.1, 0.1, 0.7, 0.2], & E_3 = 1.1, \end{array} \qquad \forall \boldsymbol{p} \in \mathbb{R}^4, f(\boldsymbol{p}) := \sum_{1 \leqslant t \leqslant 4} 0.8 \times p_t + 0.1 \times p_t^2 \ .$$

Considering the aggregate equality constraint $\sum_{1\leqslant t \leqslant 4} p_t = \sum_{1 \leqslant n \leqslant 3} E_n = 3.3$, we use the canonical projection of four-dimensional vectors into the three-dimensional space $(p_1, p_2, p_3)$ to visualize the cuts and solutions. In this example, there exist $2^T - 2 = 14$ nontrivial Hoffman inequalities characterizing disaggregation from Theorem 2.3. The projection of the obtained polytope $\mathcal{P}_{\mathrm{D}}$, as defined in (2.1), is represented in Figure 5.1(a). One can remark that this polytope has only six facets. Our Algorithm 3.4 applied on this instance with $\varepsilon_{\mathrm{dis}} = 10^{-3}$ and $\varepsilon_{\mathrm{cvg}} = 10^{-5}$ converges in three iterations, with successive solutions of the master problem (3.1) and cuts added:

$$\begin{aligned} &\boldsymbol{p}^{(1)} = [1., 0.4, 1., 0.9] && \overset{\text{cut}}{\longrightarrow} && p_1 + p_2 + p_4 \leqslant 1.9, \\ &\boldsymbol{p}^{(2)} = [0.75, 0.4, 1.4, 0.75] && \overset{\text{cut}}{\longrightarrow} && p_2 + p_3 + p_4 \leqslant 2.4, \\ &\boldsymbol{p}^{(3)} = [0.9, 0.4, 1.4, 0.6] \ . \end{aligned}$$

Figure 5.1(b) represents in the projection space the three successive solutions and the two generated cuts (in red for each iteration).

On the other hand, applying Algorithm 4.1 with the same precision parameters $(\varepsilon_{\mathrm{dis}}, \varepsilon_{\mathrm{cvg}})$, there are three cuts generated and four resolutions of the master problem needed for convergence, given by (we refer the reader to Remark 4.4 for the numerical precision obtained in the values)


$\hat{\boldsymbol{p}}^{(2)}$
$\boldsymbol{p}^{(2)}$
$\boldsymbol{p}^{(3)}=\hat{\boldsymbol{p}}^{(4)}$
$\hat{\boldsymbol{p}}^{(3)}$
$\boldsymbol{p}^{(1)}=\hat{\boldsymbol{p}}^{(1)}$
$\boldsymbol{p}^{(2)}$
$\boldsymbol{p}^{(1)}$
$p_3$
$p_2$
$p_1$
$\boldsymbol{p}^{(2)}$
$\boldsymbol{p}^{(3)}$
$\boldsymbol{p}^{(1)}$
$p_3$
$p_2$
$p_1$
$\hat{\boldsymbol{p}}^{(2)}$
$\hat{\boldsymbol{p}}^{(1)}$
$p_3$
$p_2$
$p_1$
$\hat{\boldsymbol{p}}^{(2)}$
$\hat{\boldsymbol{p}}^{(3)}$
$\hat{\boldsymbol{p}}^{(1)}$
$p_3$
$p_2$
$p_1$
$\hat{\boldsymbol{p}}^{(4)}$
$\hat{\boldsymbol{p}}^{(2)}$
$\hat{\boldsymbol{p}}^{(3)}$
$\hat{\boldsymbol{p}}^{(1)}$
$p_3$
$p_2$
$p_1$

(a) Projected polyhedron $\mathcal{P}_{\mathrm{D}}$ (b) Solutions and cuts (projected) obtained with Algorithm 3.4

(c) Solutions and cuts (projected) obtained with Algorithm 4.1

FIG. 5.1. *Illustration of the iterations of the proposed decomposition method. The cut* $p_3 \geqslant 1.4$*, which is added at first for Algorithm* 3.4*, is only added at the third iteration of Algorithm* 4.1*.*

$$
\begin{aligned}
&\hat{\boldsymbol{p}}^{(1)} = [1., 0.4, 1., 0.9] && \xrightarrow{\text{cut}} \quad -0.25p_1 - 0.25p_2 + 1.0p_3 - 0.5p_4 \geqslant 0.75,\\
&\hat{\boldsymbol{p}}^{(2)} = [0.8097, 0.4, 1.3984, 0.6919] && \xrightarrow{\text{cut}} \quad 1.0p_1 - 0.509p_2 + 0.018p_3 - 0.509p_4 \geqslant 0.4161,\\
&\hat{\boldsymbol{p}}^{(3)} = [0.9062, 0.4, 1.3823, 0.6115] && \xrightarrow{\text{cut}} \quad -0.333p_1 - 0.333p_2 + 1.0p_3 - 0.333p_4 \geqslant 0.7666,\\
&\hat{\boldsymbol{p}}^{(4)} = [0.9, 0.4, 1.4, 0.6]\ .
\end{aligned}
$$

The four successive solutions and the three added cuts are represented in the three-dimensional space in Figure 5.1(c): we observe that the last cut needed to obtain the convergence of Algorithm 4.1 corresponds to the first one added with Algorithm 3.4.

Due to the strict convexity of the cost function $\boldsymbol{p} \mapsto f(\boldsymbol{p})$, the final solution obtained is the same, a unique aggregated optimal solution of (1.1). The four successive solutions and the three added cuts are represented in the three-dimensional space in Figure 5.1(c): we observe that the last cut needed to obtain the convergence of Algorithm 4.1 corresponds to the first one added with Algorithm 3.4.

**5.2. A nonconvex example: Management of a microgrid.** In this section, we illustrate the proposed method on a larger-scale practical example from energy. We consider an electricity microgrid [28] composed of $N$ electricity consumers with flexible appliances (such as electric vehicles or water heaters), a photovoltaic (PV) power plant, and a conventional generator. The operator of the microgrid aims at satisfying the demand constraints of consumers over a set of time periods $\mathcal{T} = \{1, \dots, T\}$, while minimizing the energy cost for the community. We have the following characteristics:

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

$$f(p^g) = \alpha_k + c_k p^g, \text{ if } p^g \in \mathcal{I}_k := [\theta_{k-1},\ \theta_k[,\ k = 1\dots K, \text{ where } \theta_0 := 0 \text{ and } \theta_K := \overline{p}^g;$$

- each agent $n \in [N]$ 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 $\underline{\boldsymbol{x}}_n, \overline{\boldsymbol{x}}_n$. These parameters are confidential because they could, for instance, contain some information on agent $n$ habits.

The master problem (3.1) can be written as the following MILP (5.2):

$$\text{(5.2a)} \qquad \min_{\boldsymbol{p},\boldsymbol{p}^g,(\boldsymbol{p}_k^g),(\boldsymbol{b}_k),\boldsymbol{b}^{\mathrm{ON}},\boldsymbol{b}^{\mathrm{ST}}} \sum_{t\in[T]} \left( \alpha_1 b_t^{\mathrm{ON}} + \sum_k c_k p_{k\,t}^g + C^{\mathrm{ST}} b_t^{\mathrm{ST}} \right),$$

$$\text{(5.2b)} \qquad p_t^g = \textstyle\sum_{k=1}^K p_{k,t}^g \ \forall t \in [T],$$

$$\text{(5.2c)} \qquad b_{k,t}(\theta_k - \theta_{k-1}) \leqslant p_{k,t}^g \leqslant b_{k-1,t}(\theta_k - \theta_{k-1}) \ \forall 1 \leqslant k \leqslant K,\ \forall t \in [T],$$

$$\text{(5.2d)} \qquad b_t^{\mathrm{ST}} \geqslant b_t^{\mathrm{ON}} - b_{t-1}^{\mathrm{ON}} \ \forall t \in \{2, \dots, T\},$$

$$\text{(5.2e)} \qquad \underline{p}^g b_t^{\mathrm{ON}} \leqslant p_t^g \leqslant \overline{p}^g b_t^{\mathrm{ON}} \ \forall t \in \mathcal{T},$$

$$\text{(5.2f)} \qquad b_t^{\mathrm{ON}}, b_t^{\mathrm{ST}}, b_{1,t}, \dots, b_{K-1,t} \in \{0,1\} \ \forall t \in [T],$$

$$\text{(5.2g)} \qquad \boldsymbol{p} \leqslant \boldsymbol{p}^{\mathrm{PV}} + \boldsymbol{p}^g,$$

$$\text{(5.2h)} \qquad \boldsymbol{p}^\top \mathbb{1}_T = \boldsymbol{E}^\top \mathbb{1}_N,$$

$$\text{(5.2i)} \qquad \underline{\boldsymbol{x}}^\top \mathbb{1}_N \leqslant \boldsymbol{p} \leqslant \overline{\boldsymbol{x}}^\top \mathbb{1}_N\ .$$

In this formulation (5.2b)–(5.2c), where $b_{0,t} := 1$ and $b_{K,t} := 0$, are a mixed integer formulation of the generation cost function $f$. One can show that the Boolean variable $b_{k,t}$ is equal to one iff $p_t^g \geqslant \theta_k$ for each $k \in \{1, \dots, K-1\}$. Note that only $\alpha_1$ appears in (5.2a) because of the continuity of $f$.

Constraints (5.2d)–(5.2e) ensure the on/off and starting constraints of the power plant, (5.2g) ensures that the power allocated to consumption is not above the total production, and (5.2h)–(5.2i) are the aggregated feasibility conditions already referred to in (2.3). The nonconvexity of (5.2) comes from the existence of starting costs and constraints of minimal power, which makes it necessary to use Boolean state variables $b^{\mathrm{ST}}, b^{\mathrm{ON}}$.

We simulate the problem described above for different values of $N \in \{2^4, 2^5, 2^6, 2^7, 2^8\}$ and 100 instances with random parameters for each value of $N$. A scaling factor $\kappa_N = N/20$ is applied on parameters to ensure that production capacity is large enough to meet consumer demand. The parameters are chosen as follows:

- $T = 24$ (hours of a day);
- production costs: $K = 3$, $\theta = [0, 70, 100, 300]\kappa_N, \boldsymbol{c} = [0.2, 0.4, 0.5]$, $\underline{p}^g = 50\kappa_N, \overline{p}^g = 300\kappa_N$, $\alpha_1 = 4$, and $C^{\mathrm{ST}} = 15$;
- photovoltaic: $p_t^{\mathrm{PV}} = [50(1 - \cos(\frac{(t-6)2\pi}{16}) + \mathcal{U}([0,10])]\ \kappa_N$ for $t \in \{6, \dots, 20\}$, $p_t^{\mathrm{PV}} = 0$ otherwise;
- consumption parameters drawn randomly with $\underline{x}_{n,t} \sim \mathcal{U}([0,10])$, $\overline{x}_{n,t} \sim \mathcal{U}([0,5]) + \underline{x}_{n,t}$, and $E_n \sim \mathcal{U}([\mathbb{1}_T^\top \underline{\boldsymbol{x}}_n, \mathbb{1}_T^\top \overline{\boldsymbol{x}}_n])$, so that individual feasibility $(\mathcal{X}_n \neq \emptyset)$ is ensured.

Table 5.1
*Number of subproblems solved (average on* 100 *instances).*

| $N =$ | $2^4$ | $2^5$ | $2^6$ | $2^7$ | $2^8$ |
|---|---|---|---|---|---|
| # master | 193.6 | 194.1 | 225.5 | 210.9 | 194.0 |
| # projs. | 9507 | 15367 | 24319 | 26538 | 26646 |

We implement Algorithm 3.4 using Python 3.5. The MILP (5.2) 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 3.3), we use $\varepsilon_{\text{dis}} = 0.01$ with the operator norm defined by $|||\boldsymbol{x}||| = \max_{n\in[N]} \sum_t |x_{n,t}|$ (to avoid the $\sqrt{N}$ factor in the convergence criteria appearing with $\|.\|_2$) and start with $\varepsilon_{\text{cvg}} = 0.1$. The largest instances took around 10 minutes to be solved in this configuration and without parallel implementation. As the CPU time needed depends 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.

Table 5.1 gives the number of master problems solved and the total number of projections computed, on average over the 100 instances for each value of $N$.

One observes that the number of master problems (5.2) solved (number of "cuts" added) remains almost constant when $N$ increases. In all instances, this number is way below the upper bound of $2^{24} > 1,6 \times 10^7$ possible constraints (see Proposition 3.12), which suggests that only a limited 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 3.3, line 1, over all calls of APM in the instance) increases in a sublinear way which is even better that one could expect from the upper bound given in Theorem 3.20.

## 6. Conclusion.

We provided a nonintrusive algorithm that enables us to compute an optimal resource allocation, find a solution of a—possibly nonconvex—optimization problem, and assign to each agent an individual profile satisfying a global demand and lower and upper bounds constraints.e Our method uses local projections and works in a distributed fashion. Hence, the resolution of the problem is still efficient even in the case of a very large number of agents. The method is also 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 for this work. Section 4 generalizes the procedure to arbitrary polyhedral constraints for agents. However, the number of constraints (cuts) added to the master problem is not proved to be finite as done in the transportation case. Proving that only a finite number of constraints can be added (maybe up to a refinement procedure of the current constraint obtained) will enable us to have a termination result for the algorithm in the general polyhedral case. In the transportation case, we showed the geometric convergence of APM with a rate linear in the number of agents. Moreover, the number of cuts added in the procedure is finite but the upper bound that we have remains exponential. In practice however, the number of constraints to consider remains small, as seen in section 5. A thinner upper bound on the number of cuts added in the algorithm in this case would constitute an interesting result.

## Appendix A. Proof of Proposition 3.4.

*Proof of item* (i). Let us write the stationarity conditions associated to problem (3.3):

$$\forall n \in [N], \forall t \in [T], \quad 0 = (x_{n,t} - y_{n,t}) - \lambda_n - \underline{\mu}_{n,t} + \overline{\mu}_{n,t} \quad \text{and} \quad y_{n,t} = x_{n,t} + \nu_t \ . \tag{A.1}$$

By summing the latter equalities on $t \in [T]$ and $n \in [N]$, we obtain the equalities

$$\sum_t \nu_t = \sum_t y_{n,t} - E_n \ \forall n \in [N], \qquad\qquad p_t = \sum_n x_{n,t} + N\nu_t \ \forall t \in [T] \ . \tag{A.2}$$

Moreover, by summing over $t \in \mathcal{T}_n^\circ$ the first series of equalities in (A.1), and by using the complementary slackness conditions, which entail that $\underline{\mu}_{n,t} = \overline{\mu}_{n,t} = 0$ for $t \in \mathcal{T}_n^\circ$, we get

$$|\mathcal{T}_n^\circ|\lambda_n = E_n - \sum_{t \in \underline{\mathcal{T}}_n} \underline{x}_{n,t} - \sum_{t \in \mathcal{T}_n^\circ} y_{n,t} - \sum_{t \in \overline{\mathcal{T}}_n} \overline{x}_{n,t} \quad \forall n \in [N] \ , \tag{A.3}$$

where we define for each $n \in [N]$

$$\mathcal{T}_n^\circ := \{t \mid \underline{x}_{n,t} < x_{n,t} < \overline{x}_{n,t}\}, \quad \underline{\mathcal{T}}_n = \{t \mid x_{n,t} = \underline{x}_{n,t}\} \ \text{ and } \ \overline{\mathcal{T}}_n = \{t \mid x_{n,t} = \overline{x}_{n,t}\} \ .$$

From (A.2) and Assumption 2 giving the equality $\sum_n E_n = \sum_t p_t$, we obtain $\sum_t \nu_t = 0$ and

$$\forall n \in [N], \ \textstyle\sum_{t \in [T]} y_{n,t} = E_n. \tag{A.4}$$

Let us show that $\forall t \in \mathcal{T}, \nu_t > 0$. For $t \in \mathcal{T}$, there exists $n$ s.t. $y_{n,t} > \overline{x}_{n,t}$. From (A.1) we get

$$\nu_t = y_{n,t} - x_{n,t} \geqslant y_{n,t} - \overline{x}_{n,t} > 0 \ . \tag{A.5}$$

Suppose by contradiction that there exists $n \notin \mathcal{N}$ and $\hat{t} \in \mathcal{T}$ such that $x_{n,\hat{t}} < \overline{x}_{n,\hat{t}}$. We obtain from complementary slackness conditions and the first equality in (A.1)

$$x_{n,\hat{t}} \geqslant y_{n,\hat{t}} + \lambda_n = x_{n,\hat{t}} + \nu_{\hat{t}} + \lambda_n, \quad \text{which implies that} \quad \lambda_n < 0 \ . \tag{A.6}$$

From this, we obtain $\overline{\mathcal{T}}_n \subset \mathcal{T}$: indeed, for $t \in \overline{\mathcal{T}}_n$, we have $y_{n,t} + \lambda_n \geqslant \overline{x}_{n,t}$, which gives

$$y_{n,t} \geqslant \overline{x}_{n,t} - \lambda_n > \overline{x}_{n,t}, \ \text{which implies } t \in \mathcal{T} \ .$$

From the condition (A.4) and as $\nu_t > 0$ for each $t \in \overline{\mathcal{T}}_n$ because $\overline{\mathcal{T}}_n \subset \mathcal{T}$ and (A.5), we get

$$0 = \sum_{t \in [T]} (y_{n,t} - x_{n,t}) = \sum_{t \in \underline{\mathcal{T}}_n} (y_{n,t} - \underline{x}_{n,t}) + \sum_{t \in \mathcal{T}_n^\circ} (-\lambda_n) + \sum_{t \in \overline{\mathcal{T}}_n} \nu_t, \ \text{ i.e.,}$$

$$\sum_{t \in \underline{\mathcal{T}}_n} (y_{n,t} - \underline{x}_{n,t}) = \sum_{t \in \mathcal{T}_n^\circ} \lambda_n - \sum_{t \in \overline{\mathcal{T}}_n} \nu_t,$$

which is strictly negative: this implies that there exists $t' \in \underline{\mathcal{T}}_n$ such that $y_{n,t'} < \underline{x}_{n,t'}$. Necessarily, $t' \notin \mathcal{T}$ because $\nu_{t'} = y_{n,t'} - x_{n,t'} < \underline{x}_{n,t'} - \underline{x}_{n,t'} = 0$. Then, as we have

$\sum_{m\in[N]} y_{m,t'} = p_{t'} \geqslant \sum_m \underline{x}_{m,t'}$, there exists $m \in [N]$ such that $y_{m,t'} > \underline{x}_{m,t'}$. If $\lambda_m \leqslant 0$, and as $x_{m,t'} = y_{m,t'} - \nu_{t'} > \underline{x}_{m,t'}$, we get

$$x_{m,t'} = \min(\overline{x}_{m,t'}, y_{m,t'} + \lambda_m) \leqslant y_{m,t'} + \lambda_m \leqslant y_{m,t'} = x_{m,t'} + \nu_{t'} < x_{m,t'}\ ,$$

which is impossible, thus $\lambda_m > 0$. Now, we observe that $\mathcal{T}_n^\circ \subset \mathcal{T}$. Indeed, otherwise, if $t'' \in \mathcal{T}_n^\circ \cap \mathcal{T}^c$, we have $\nu_{t''} = -\lambda_n > 0$ and $x_{m,t''} = y_{m,t''} - \nu_{t''} < y_{m,t''} < \overline{x}_{m,t''}$, and thus we get

$$x_{m,t''} = \max(\underline{x}_{m,t''}, y_{m,t''} + \lambda_m) \geqslant y_{m,t''} + \lambda_m > y_{m,t''} = x_{m,t''} + \nu_{t''} + \lambda_m > x_{m,t''}\ ,$$

which is impossible, thus $\mathcal{T}_n^\circ \subset \mathcal{T}$.

Finally, since $\overline{\mathcal{T}}_n^c \neq \emptyset$, consider $t_0 \in \arg\min_{t\notin\overline{\mathcal{T}}_n}\{\overline{x}_{n,t} - y_{n,t}\}$. By (A.3), we obtain

$$y_{n,t_0} + \lambda_n < \overline{x}_{n,t_0} \iff E_n - \sum_{t\in\underline{\mathcal{T}}_n} \underline{x}_{n,t} - \sum_{t\in\mathcal{T}_n^\circ} y_{n,t} - \sum_{t\in\overline{\mathcal{T}}_n} \overline{x}_{n,t} < |\mathcal{T}_n^\circ|(\overline{x}_{n,t_0} - y_{n,t_0}) \tag{A.7}$$

and thus

$$\begin{aligned}
E_n - \sum_{t\in\mathcal{T}^c} \underline{x}_{n,t} - \sum_{t\in\mathcal{T}} \overline{x}_{n,t} &= E_n - \sum_{t\in\underline{\mathcal{T}}_n} \underline{x}_{n,t} + \sum_{t\in\underline{\mathcal{T}}_n\cap\mathcal{T}} \underline{x}_{n,t} - \sum_{t\in\overline{\mathcal{T}}_n\cup\mathcal{T}_n^\circ} \overline{x}_{n,t} - \sum_{t\in\underline{\mathcal{T}}_n\cap\mathcal{T}} \overline{x}_{n,t} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\text{as } \overline{\mathcal{T}}_n \cup \mathcal{T}_n^\circ \subset \mathcal{T}) \\
&< \sum_{t\in\mathcal{T}_n^\circ} (\overline{x}_{n,t_0} - y_{n,t_0}) - (\overline{x}_{n,t} - y_{n,t}) + \sum_{\underline{\mathcal{T}}_n\cap\mathcal{T}} (\underline{x}_{n,t} - \overline{x}_{n,t}) \quad (\text{from (A.7)}) \qquad \text{(A.8)} \\
&\leqslant 0 \quad (\text{from the definition of } t_0 \text{ and } \underline{x}_{n,t} \leqslant \overline{x}_{n,t}), \qquad \text{(A.9)}
\end{aligned}$$

which contradicts $n \notin \mathcal{N}$. Thus we have shown that $\forall n \notin \mathcal{N}, \forall t \in \mathcal{T},\ x_{n,t} = \overline{x}_{n,t}$. From (A.1) and as $\forall t \in \mathcal{T}, \nu_t > 0$ from (A.5), we obtain $\forall n \notin \mathcal{N}, \forall t \in \mathcal{T},\ y_{n,t} = x_{n,t} + \nu_t > \overline{x}_{n,t}$. This terminates the proof for item (i). □

*Proof of item* (ii). Let us consider $\mathcal{T}' := \{t | \nu_t > 0\}$. We already showed that $\mathcal{T} \subset \mathcal{T}'$ in (A.5). To prove the other inclusion and obtain item (ii), we observe that, considering $n \notin \mathcal{N}$ (nonempty as shown independently in item (v)) and considering $\mathcal{T}'$ instead of $\mathcal{T}$, all the facts established in the proof of item (i) above also hold: by contradiction of $\forall t \in \mathcal{T}'$, $x_{n,t} = \overline{x}_{n,t}$, we first obtain $\lambda_n < 0$ as in (A.6). Then, as for $t'' \in \mathcal{T}_n^\circ$, we have $\nu_{t''} = -\lambda_n > 0$, we get $\mathcal{T}_n^\circ \cap \mathcal{T}'^c = \emptyset$. Finally the same sequence of inequalities as (A.8)–(A.9) shows a contradiction. Consequently, for each $t \in \mathcal{T}'$, $x_{n,t} = \overline{x}_{n,t}$ and $y_{n,t} = x_{n,t} + \nu_t > \overline{x}_{n,t}$, thus $t \in \mathcal{T}$ and $\mathcal{T}' \subset \mathcal{T}$. □

*Proof of item* (iii). Suppose on the contrary that there exists $n \in \mathcal{N}$ such that $\lambda_n \geqslant 0$. For $t \in \mathcal{T}_n^\circ$, we have $\nu_t = -\lambda_n \leqslant 0$, thus, $\mathcal{T}_n^\circ \subset \mathcal{T}^c$. Then, if $t \in \mathcal{T}$ and if $x_{n,t} < \overline{x}_{n,t}$, we would have

$$x_{n,t} = \max(\underline{x}_{n,t}, y_{n,t} + \lambda_n) \geqslant x_{n,t} + 0 + \nu_t > x_{n,t}\ ,$$

which is impossible, thus $x_{n,t} = \overline{x}_{n,t}$, and $\mathcal{T} \subset \overline{\mathcal{T}}_n$. As we show independently in item (v) that $\mathcal{T} \neq \emptyset$, we know $\overline{\mathcal{T}}_n \neq \emptyset$. Let us consider $t_0 \in \arg\min_{t\notin\underline{\mathcal{T}}_n}\{y_{n,t} - \underline{x}_{n,t}\}$. By (A.3), we obtain

$$y_{n,t_0} + \lambda_n > \underline{x}_{n,t_0} \iff E_n - \sum_{t\in\underline{\mathcal{T}}_n} \underline{x}_{n,t} - \sum_{t\in\mathcal{T}_n^\circ} y_{n,t} - \sum_{t\in\overline{\mathcal{T}}_n} \overline{x}_{n,t} > |\mathcal{T}_n^\circ|(\underline{x}_{n,t_0} - y_{n,t_0}) \tag{A.10}$$

and thus

$$E_n - \sum_{t\in\mathcal{T}^c} \underline{x}_{n,t} - \sum_{\mathcal{T}} \overline{x}_{n,t} = E_n - \sum_{t\in\underline{\mathcal{T}}_n} \underline{x}_{n,t} - \sum_{t\in\mathcal{T}^c\cap\overline{\mathcal{T}}_n} \underline{x}_{n,t} - \sum_{t\in\mathcal{T}_n^\circ} \underline{x}_{n,t} - \sum_{t\in\overline{\mathcal{T}}_n} \overline{x}_{n,t} + \sum_{t\in\mathcal{T}^c\cap\overline{\mathcal{T}}_n} \overline{x}_{n,t}$$

$$\text{(as } \mathcal{T}\subset\overline{\mathcal{T}}_n)$$

$$> \sum_{t\in\mathcal{T}_n^\circ} \big((y_{n,t}-\underline{x}_{n,t}) - (y_{n,t_0}-\underline{x}_{n,t_0})\big) + \sum_{t\in\mathcal{T}^c\cap\overline{\mathcal{T}}_n} \overline{x}_{n,t} - \underline{x}_{n,t} \quad \text{(from (A.10))} \tag{A.11}$$

$$\geqslant 0 \qquad \text{(from the definition of } t_0 \text{ and } \underline{x}_{n,t}\leqslant\overline{x}_{n,t}), \tag{A.12}$$

which contradicts $n\in\mathcal{N}$ and terminates the proof of item (iii). □

*Proof of item* (iv). From (ii), we know that $\mathcal{T}^c=\{t|\nu_t\leqslant 0\}$; thus, if $t\notin\mathcal{T}$ and $n\in\mathcal{N}$, if $x_{n,t}>\underline{x}_{n,t}$, then we would have $x_{n,t}\leqslant y_{n,t}+\lambda_n = x_{n,t}+\nu_t+\lambda_n < x_{n,t}$, which is a contradiction. □

*Proof of item* (v). From $\sum_t \nu_t = 0$, we see that if $\mathcal{T}=\emptyset$, then this means that $\nu_t=0\ \forall t\in[T]$, and thus $\boldsymbol{y}=\boldsymbol{x}$, which is a contradiction. Thus there exists $t_0$ such that $\nu_{t_0}>0$ and because $\sum_{t\in\mathcal{T}}\nu_t=0$, there exists $t_0'$ such that $\nu_{t_0'}<0$.

If $\mathcal{N}=\emptyset$, then using (i), we would have $\forall n,\ y_{n,t_0}>\overline{x}_{n,t_0}$ and thus $p_{t_0}>\sum_{n\in[N]}\overline{x}_{n,t_0}$, which contradicts the aggregate upper bound constraint $\forall t,\ p_t\leqslant\sum_{n\in[N]}\overline{x}_{n,t}$.

If $\mathcal{N}^c=\emptyset$, then using (iv), we would have $\forall n,\ y_{n,t_0'}<\underline{x}_{n,t_0'}$ and thus $p_{t_0'}<\sum_{n\in[N]}\underline{x}_{n,t_0'}$, which contradicts the aggregate lower bound constraint $\forall t,\ p_t\geqslant\sum_{n\in[N]}\underline{x}_{n,t}$. □

## Appendix B. Proof of Lemma 3.10.

*Proof of item* (i). From $\boldsymbol{x}^{(K)}=P_{\mathcal{X}}(\boldsymbol{y}^{(K-1)})$ and $\boldsymbol{y}^{(K)}=P_{\mathcal{Y}}(\boldsymbol{x}^{(K)})$, we obtain, similarly to (A.1),

$$\forall n\in[N],\forall t\in[T],\ 0=\left(x_{n,t}^{(K)}-y_{n,t}^{(K-1)}\right)-\lambda_n^{(K)}-\underline{\mu}_{n,t}^{(K)}+\overline{\mu}_{n,t}^{(K)} \text{ and } y_{n,t}^{(K)}=x_{n,t}^{(K)}+\nu_t^{(K)}, \tag{B.1}$$

where the Lagrangian multipliers $\lambda_n^{(K)},\underline{\mu}_{n,t}^{(K)},\overline{\mu}_{n,t}^{(K)}$ (resp., $\nu_t^{(K)}$) are associated to the quadratic problem characterizing the projections $P_{\mathcal{X}}(\boldsymbol{y}^{(K-1)})$ (resp., $\boldsymbol{y}^{(K)}=P_{\mathcal{Y}}(\boldsymbol{x}^{(K)})$). We obtain equalities similar to (A.2), (A.3). We proceed as for Proposition 3.4(i) and suppose that there exists $n\notin\mathcal{N}$ and $\hat{t}\in\mathcal{T}$ such that $x_{n,\hat{t}}<\overline{x}_{n,\hat{t}}$. Then, as $\|\boldsymbol{y}^{(K)}-\boldsymbol{y}^\infty\|\leqslant\frac{\varepsilon_{\text{cvg}}}{1-\rho}$ and $\sum_{t\in[T]}\boldsymbol{y}_t^{(K)}=\sum_{t\in[T]}\boldsymbol{y}_t^\infty$, we have for each $n\in[N],t\in[T]$, $|y_{n,t}^{(K)}-y_{n,t}^\infty|\leqslant\frac{\varepsilon_{\text{cvg}}}{2(1-\rho)}$, and thus we get

$$\begin{aligned}&\overline{x}_{n,\hat{t}}\geqslant x_{n,\hat{t}}^{(K)}\geqslant y_{n,\hat{t}}^{(K-1)}+\lambda_n^{(K)}\geqslant y_{n,\hat{t}}^\infty-\tfrac{\varepsilon_{\text{cvg}}}{2(1-\rho)}+\lambda_n^{(K)}=x_{n,\hat{t}}^\infty+\nu_{\hat{t}}^\infty-\tfrac{\varepsilon_{\text{cvg}}}{2(1-\rho)}+\lambda_n^{(K)}\\ &\Longrightarrow \lambda_n^{(K)}<\tfrac{\varepsilon_{\text{cvg}}}{2(1-\rho)}\upsilon-\nu_{\hat{t}}^\infty<\tfrac{B\varepsilon_{\text{cvg}}}{2}-2B\varepsilon_{\text{cvg}}=-\tfrac{3}{2}B\varepsilon_{\text{cvg}}\ ,\end{aligned} \tag{B.2}$$

as $\nu_{\hat{t}}^\infty\geqslant\underline{\nu}>2B\varepsilon_{\text{cvg}}$. Let us now consider $t'\in\mathcal{T}_n^{\circ(K)}\cup\overline{\mathcal{T}}_n^{(K)}$; then,

$$\nu_{t'}^{(K)}=y_{n,t'}^{(K)}-x_{n,t'}^{(K)}\geqslant y_{n,t'}^{(K)}-y_{n,t'}^{(K)}-\lambda_n^{(K)}>-\frac{\varepsilon_{\text{cvg}}}{2}+\frac{3}{2}B\varepsilon_{\text{cvg}}>B\varepsilon_{\text{cvg}}+\frac{\varepsilon_{\text{cvg}}}{2}(B-1)\geqslant B\varepsilon_{\text{cvg}}\ , \tag{B.3}$$

which shows that $t' \in \mathcal{T}^{(K)} = \mathcal{T}^{\infty}$ and thus $\mathcal{T}_n^{\circ(K)} \cup \overline{\mathcal{T}}_n^{(K)} \subset \mathcal{T}$. Then, the same sequence of inequalities as (A.7), (A.8), (A.9) applied to $\boldsymbol{y}^{(K-1)}$ gives a contradiction to $n \notin \mathcal{N}$. □

*Proof of item* (ii). The proof of item (ii) is symmetric to the one of item (i): if we suppose that there exists $n \in \mathcal{N}$ and $\hat{t} \notin \mathcal{T}$ such that $x_{n,\hat{t}}^{(K)} > \underline{x}_{n,\hat{t}}$, we obtain, symmetrically to (B.2), that $\lambda_n^{(K)} \geqslant -\frac{\varepsilon_{\mathrm{cvg}}}{2(1-\rho)}$. Then, considering $t' \in \underline{\mathcal{T}}_n^{(K)} \cup \mathcal{T}_n^{\circ(K)}$, we show, symmetrically to (B.3), that $\nu_{t'}^{(K)} < B\varepsilon_{\mathrm{cvg}}$, i.e., $t' \notin \mathcal{T}$ and thus $\underline{\mathcal{T}}_n^{(K)} \cup \mathcal{T}_n^{\circ(K)} \subset \mathcal{T}^c$. We conclude by obtaining a contradiction to $n \in \mathcal{N}$ by the same sequence of inequalities as (A.10), (A.11), (A.12). □

**Acknowledgments.** We thank the referees for their comments, leading to improvements of this paper. We thank Daniel Augot for very helpful discussions concerning secure multiparty computations and cryptographic protocols.

## REFERENCES

[1] E. A. Abbe, A. E. Khandani, and A. W. Lo, *Privacy-preserving methods for sharing financial risk exposures*, Amer. Econ. Rev., 102 (2012), pp. 65–70.

[2] Y. P. Aneja and K. P. Nair, *Bicriteria transportation problem*, Management Sci., 25 (1979), pp. 73–78.

[3] M. F. Anjos, A. Lodi, and M. Tanneau, *A decentralized framework for the optimal coordination of distributed energy resources*, IEEE Trans. Pow. Sys., 34 (2018), pp. 349–359.

[4] M. Atallah, M. Bykova, J. Li, K. Frikken, and M. Topkara, *Private collaborative forecasting and benchmarking*, in Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society, 2004, pp. 103–114.

[5] H. H. Bauschke and J. M. Borwein, *On the convergence of Von Neumann's alternating projection algorithm for two sets*, Set-Valued Anal., 1 (1993), pp. 185–212.

[6] H. H. Bauschke and J. M. Borwein, *Dykstra's alternating projection algorithm for two sets*, J. Approx. Theory, 79 (1994), pp. 418–443.

[7] H. H. Bauschke, J. Chen, and X. Wang, *A Bregman projection method for approximating fixed points of quasi-Bregman nonexpansive mappings*, Appl. Anal., 94 (2015), pp. 75–84.

[8] J. F. Benders, *Partitioning procedures for solving mixed variables programming problems*, Numer. Math., 4 (1962), pp. 238–252.

[9] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1999.

[10] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall Englewood Cliffs, NJ, 1989.

[11] E. D. Bolker, *Transportation polytopes*, J. Combin. Theory Ser. B, 13 (1972), pp. 251–262.

[12] J. M. Borwein, G. Li, and L. Yao, *Analysis of the convergence rate for the cyclic projection algorithm applied to basic semialgebraic convex sets*, SIAM J. Optim., 24 (2014), pp. 498–527.

[13] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu, *Tools for privacy preserving distributed data mining*, ACM SIGKDD Explorations Newsletter, 4 (2002), pp. 28–34.

[14] G. Cohen and D. L. Zhu, *Decomposition and coordination methods in large scale optimization problems: The nondifferentiable case and the use of augmented lagrangians*, Adv. Large Scale Syst. 1, (1984), pp. 203–266.

[15] W. J. Cook, W. Cunningham, W. Pulleyblank, and A. Schrijver, *Combinatorial Optimization*, Springer, New York, 2009.

[16] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, *Parallel multi-block admm with o (1/k) convergence*, J. Sci. Comput., 71 (2017), pp. 712–736.

[17] R. L. Dykstra, *An algorithm for restricted least squares regression*, J. Amer. Statist. Assoc., 78 (1983), pp. 837–842.

[18] D. Gale, *A theorem on flows in networks*, Pacific J. Math., 7 (1957), pp. 1073–1082.

[19] R. Glowinski and A. Marroco, *Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires*, ESAIM, 9 (1975), pp. 41–76.

[20] O. Goldreich, S. Micali, and A. Wigderson, *How to play any mental game*, in Proceedings of the 19 Annual ACM Symposium on Theory of Computing, New York, 1987, pp. 218–229.

[21] L. Gubin, B. T. Polyak, and E. Raik, *The method of projections for finding the common point of convex sets*, USSR Comput. Math. Math. Phys., 7 (1967), pp. 1–24.

[22] J. He, L. Cai, P. Cheng, J. Pan, and L. Shi, *Consensus-based data-privacy preserving data aggregation*, IEEE Trans. Automat. Control, 64 (2019), pp. 5222–5229.

[23] A. J. Hoffman, *Some recent applications of the theory of linear inequalities to extremal combinatorial analysis*, in Proceedings of Symposia on Applied Mathematics, 1960, pp. 113–127.

[24] B. A. Huberman, E. Adar, and L. R. Fine, *Valuating privacy*, IEEE Security Privacy, 3 (2005), pp. 22–25.

[25] P. Jacquot, O. Beaude, P. Benchimol, S. Gaubert, and N. Oudjane, *A privacy-preserving disaggregation algorithm for non-intrusive management of flexible energy*, in Proceedings of the IEEE 58th Conference on Decision and Control, 2019.

[26] P. Jacquot, O. Beaude, S. Gaubert, and N. Oudjane, *Analysis and implementation of an hourly billing mechanism for demand response management*, IEEE Trans. Smart Grid, 10 (2019), pp. 4265–4278.

[27] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright, *A new privacy-preserving distributed k-clustering algorithm*, in Proceedings of the 2006 SIAM International Conference on Data Mining, SIAM, Philadelphia, 2006, pp. 494–498.

[28] F. Katiraei, R. Iravani, N. Hatziargyriou, and A. Dimeas, *Microgrids management*, IEEE Power Energy Magazine, 6 (2008).

[29] K. K. Lai, K. Lam, and W. K. Chan, *Shipping container logistics and allocation*, J. Oper. Res. Soc., 46 (1995), pp. 687–697.

[30] C. Lemaréchal, A. Nemirovskii, and Y. Nesterov, *New variants of bundle methods*, Math. Program., 69 (1995), pp. 111–147.

[31] P.-Y. R. Ma, E. Y. S. Lee, and M. Tsuchiya, *A task allocation model for distributed computing systems*, IEEE Trans. Computers, 100 (1982), pp. 41–47.

[32] F. L. Müller, J. Szabó, O. Sundström, and J. Lygeros, *Aggregation and disaggregation of energetic flexibility from distributed energy resources*, IEEE Trans. Smart Grid, 10 (2019), pp. 1205–1214.

[33] J. Munkres, *Algorithms for the assignment and transportation problems*, SIAM J. Appl. Math., 5 (1957), pp. 32–38.

[34] R. Nishihara, S. Jegelka, and M. I. Jordan, *On the convergence rate of decomposable submodular function minimization*, in Proceedings of NIPS, 2014, pp. 640–648.

[35] D. P. Palomar and M. Chiang, *A tutorial on decomposition methods for network utility maximization*, IEEE J. Sel. Areas Commun., 24 (2006), pp. 1439–1451.

[36] A. Rais and A. Viana, *Operations research in healthcare: A survey*, Int. Trans. Oper. Res., 18 (2011), pp. 1–31.

[37] M. Ruan, H. Gao, and Y. Wang, *Secure and privacy-preserving consensus*, IEEE Trans. Automat. Control, 64 (2019), pp. 4035–4049.

[38] K. Seong, M. Mohseni, and J. M. Cioffi, *Optimal resource allocation for OFDMA downlink systems*, in Proceedings of the International Symposium on Information Theory, IEEE, 2006, pp. 1394–1398.

[39] R.-h. Shi, Y. Mu, H. Zhong, J. Cui, and S. Zhang, *Secure multiparty quantum computation for summation and multiplication*, Scientific Reports, 6 (2016), pp. 1–9.

[40] J. Von Neumann, *Functional Operators: Measures and Integrals*, Vol. 1, Princeton University Press, Princeton, NJ, 1950.

[41] L. Xiao and S. Boyd, *Optimal scaling of a gradient method for distributed resource allocation*, J. Optim. Theory. Appl., 129 (2006), pp. 469–488.

[42] L. Xiao, M. Johansson, and S. P. Boyd, *Simultaneous routing and resource allocation via dual decomposition*, IEEE Trans. Comm., 52 (2004), pp. 1136–1144.

[43] A. C. Yao, *How to generate and exchange secrets*, in Proceedings of the 27th SFCS, 1986, pp. 162–167.

[44] H. Yu and M. J. Neely, *A simple parallel algorithm with an $O(1/t)$ convergence rate for general convex programs*, SIAM J. Optim., 27 (2017), pp. 759–783.

[45] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, *Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey*, Sensors, 12 (2012), pp. 16838–16866.

[46] M. Zulhasnine, C. Huang, and A. Srinivasan, *Efficient resource allocation for device-to-device communication underlaying LTE network*, in Proceedings of WiMob, IEEE, 2010, pp. 368–375.