A Privacy-Preserving Disaggregation Algorithm for Nonconvex Optimization based on Alternate Projections

P. Jacquot$^1,2$, O. Beaude$^1$, P. Benchimol$^1$, S. Gaubert$^2$, N. Oudjane$^1$

$^1$EDF Lab Saclay

$^2$Inria and CMAP, École polytechnique

Lundi 7 octobre 2019

Journée de rentrée du CMAP
Introduction and Context

Two main issues:
- Dimension: hundreds or thousands of users/consumers
- Privacy: users may not want to disclose individual constraints and consumption profiles to big brother.
Two main issues:

- **dimension**: hundreds or thousands of users/consumers;
- **privacy**: users may not want to disclose individual constraints and consumption profiles to big brother.

**Microgrid Operator**

1. \[1\]
2. \[2\]
\[n\]
\[N\]
Introduction and Context

Two main issues:

- **dimension**: hundreds or thousands of users/consumers;

Privacy-Preserving Disaggregation Algorithm
Introduction and Context

Two main issues:

- **dimension**: hundreds or thousands of users/consumers;
- **privacy**: users may not want to disclose individual constraints and consumption profiles to big brother.

Microgrid Operator

1

2

n

N
Problem Formulation

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

\[
\min_{\mathbf{x} \in \mathbb{R}^{N \times T}, \mathbf{p} \in \mathbb{R}^T} f(\mathbf{p}) \quad (1a)
\]

\[
\mathbf{p} \in \mathcal{P} \quad \text{operator constraints} \quad (1b)
\]

\[
\sum_{n \in \mathcal{N}} x_{n,t} = p_t, \quad \forall t \in \mathcal{T} \quad \text{disaggregation} \quad (1c)
\]

\[
\mathbf{x}_n \in \mathcal{X}_n, \quad \forall n \in \mathcal{N} \quad \text{agents constraints} \quad (1d)
\]
Problem Formulation

\[
\begin{align*}
\min_{x \in \mathbb{R}^{N \times T}, \; p \in \mathbb{R}^T} & \quad f(p) \\
p & \in \mathcal{P} \quad \text{operator constraints} \\
\sum_{n \in \mathcal{N}} x_{n,t} = p_t, \; \forall t \in \mathcal{T} & \quad \text{disaggregation} \\
x_n & \in \mathcal{X}_n, \; \forall n \in \mathcal{N} \quad \text{agents constraints} \\
\end{align*}
\]

with \( \mathcal{X}_n \) \text{def} = \{ x_n \in \mathbb{R}^T \mid \sum_t x_{n,t} = E_n \quad \text{and} \quad \forall t, x_{n,t} \leq x_{n,t} \leq \bar{x}_{n,t} \}
Problem Formulation

\[
\min_{x \in \mathbb{R}^{N \times T}, \ p \in \mathbb{R}^T} f(p)
\]

\[p \in \mathcal{P}\]

\[\sum_{n \in \mathcal{N}} x_{n,t} = p_t, \ \forall t \in \mathcal{T}\]

\[x_n \in \mathcal{X}_n, \ \forall n \in \mathcal{N}\]

with \(\mathcal{X}_n \overset{\text{def}}{=} \{x_n \in \mathbb{R}^T \mid \sum_t x_{n,t} = E_n\ \text{and} \ \forall t, x_{n,t} \leq x_{n,t} \leq \overline{x}_{n,t}\}\)

**Resource allocation problems:** many applications in energy, logistics, distributed computing, healthcare...
distributed problems are usually addressed by Lagrangian decomposition approaches ...
- distributed problems are usually addressed by **Lagrangian decomposition** approaches . . .
- which requires strong duality / convexity hypothesis!
- distributed problems are usually addressed by Lagrangian decomposition approaches . . .
- which requires strong duality / convexity hypothesis!
- a lot of problems have non convex constraints/ cost functions : our method does not require convexity.
Two subproblems

Our method considers two subproblems iteratively:

\[ \text{Master Problem} \]
\[
\min_{p \in \mathbb{R}^T} f(p)
\]
\[\text{s.t. } p \in P(s) , \]

\[ \text{Disaggregation Problem} \]
\[
\text{Find } x = (x_n)_{n \in N} \in Y_p \cap X_p(s)
\]
\[\text{where } Y_p \text{ def } = \{ y \in \mathbb{R}^{NT} | \sum_{n \in N} y_n = p \}, \]
\[\text{and } X_p \text{ def } = \prod_{n \in N} X_n. \]
Two subproblems

Our method considers two subproblems iteratively:

**Master Problem**

\[
\begin{align*}
\min_{p \in \mathbb{R}^T} & \quad f(p) \\
\text{s.t.} & \quad p \in \mathcal{P}(s),
\end{align*}
\]

where \( \mathcal{P}(s) \subset \mathcal{P} \)
Two subproblems

Our method considers two subproblems iteratively:

**Master Problem**

\[
\begin{align*}
\min_{p \in \mathbb{R}^T} & \quad f(p) \\
\text{s.t.} & \quad p \in \mathcal{P}^{(s)},
\end{align*}
\]

where \(\mathcal{P}^{(s)} \subset \mathcal{P}\)

**Disaggregation Problem**

\[
\begin{align*}
\text{Find} & \quad x = (x_n)_{n \in \mathcal{N}} \in \mathcal{Y}_p \cap \mathcal{X} \\
\text{where} & \quad \mathcal{Y}_p \overset{\text{def}}{=} \{ y \in \mathbb{R}^{NT} \mid \sum_{n \in \mathcal{N}} y_n = p \} \\
& \quad \text{and} \quad \mathcal{X} \overset{\text{def}}{=} \prod_{n \in \mathcal{N}} X_n.
\end{align*}
\]
Two subproblems

Our method considers two subproblems iteratively:

**Master Problem**

\[
\begin{align*}
\min_{p \in \mathbb{R}^T} & \quad f(p) \\
\text{s.t.} & \quad p \in P^{(s)} , \\
\text{where} & \quad P^{(s)} \subset P 
\end{align*}
\]

**Disaggregation Problem**

\[
\begin{align*}
\text{Find } & \quad x = (x_n)_{n \in \mathcal{N}} \in \mathcal{Y}_p \cap \mathcal{X} \\
\text{where } & \quad \mathcal{Y}_p \overset{\text{def}}{=} \{ y \in \mathbb{R}^{NT} | \sum_{n \in \mathcal{N}} y_n = p \} \\
\text{and } & \quad \mathcal{X} \overset{\text{def}}{=} \prod_{n \in \mathcal{N}} \mathcal{X}_n . 
\end{align*}
\]
Our method considers two subproblems iteratively:

**Master Problem**

$$\min_{p \in \mathbb{R}^T} f(p)$$

s.t. \( p \in \mathcal{P}^{(s)} \),

where \( \mathcal{P}^{(s)} \subset \mathcal{P} \)

**Disaggregation Problem**

Find \( x = (x_n)_{n \in \mathcal{N}} \in \mathcal{Y}_p \cap \mathcal{X} \)

where \( \mathcal{Y}_p \overset{\text{def}}{=} \{ y \in \mathbb{R}^{NT} | \sum_{n \in \mathcal{N}} y_n = p \} \)

and \( \mathcal{X} \overset{\text{def}}{=} \prod_{n \in \mathcal{N}} \mathcal{X}_n \).
Disaggregation Feasibility

Characterizing \( \mathcal{Y}_p \cap \mathcal{X} = \{ x \in \mathcal{X} \mid \sum_{n \in \mathcal{N}} x_n = p \} \)

Necessary aggregated constraints:

\[
\sum_t p_t = \sum_n E_n \quad \text{and} \quad \forall t, \quad \sum_n x_{n,t} \leq p_t \leq \sum_n x_{n,t} .
\]

Not sufficient!
Disaggregation Feasibility

Characterizing $\mathcal{Y}_p \cap \mathcal{X} = \{ x \in \mathcal{X} | \sum_{n \in \mathcal{N}} x_n = p \}$

Necessary aggregated constraints:

$$\sum_{t} p_t = \sum_{n} E_n \quad \text{and} \quad \forall t, \sum_{n} x_{n,t} \leq p_t \leq \sum_{n} \overline{x}_{n,t}.$$ 

Not sufficient!

Diagram:
Disaggregation Feasibility

Characterizing $\mathcal{Y}_p \cap \mathcal{X} = \{ \mathbf{x} \in \mathcal{X} | \sum_{n \in \mathcal{N}} x_n = p \}$

**Theorem (Hoffman Circulation’s Theorem)**

Disaggregation is feasible (i.e. $\mathcal{X} \cap \mathcal{Y}_p \neq \emptyset$) iff for any $\mathcal{T}_{in} \subset \mathcal{T}, \mathcal{N}_{in} \subset \mathcal{N}$:

$$\sum_{t \notin \mathcal{T}_{in}} p_t \leq \sum_{t \notin \mathcal{T}_{in}, n \in \mathcal{N}_{in}} \bar{x}_{n,t} - \sum_{t \in \mathcal{T}_{in}, n \notin \mathcal{N}_{in}} x_{n,t} + \sum_{n \notin \mathcal{N}_{in}} E_n.$$  \hspace{1cm} (2)
\[ \mathcal{X} = \prod_n \mathcal{X}_n \quad \text{and} \quad \mathcal{Y} = \mathcal{Y}_p = \{ x \in \mathbb{R}^{NT} \mid \sum_{n \in \mathcal{N}} x_n = p \} \]

**Require:** \( y^{(0)} \), \( k = 0 \), \( \varepsilon_{\text{cvg}} \), \( \| . \| \)

repeat
\[
\begin{align*}
\mathbf{x}^{(k+1)} & \leftarrow P_{\mathcal{X}}(y^{(k)}) \\
\mathbf{y}^{(k+1)} & \leftarrow P_{\mathcal{Y}}(\mathbf{x}^{(k+1)}) \\
k & \leftarrow k + 1
\end{align*}
\]
until \( \| y^{(k)} - y^{(k-1)} \| < \varepsilon_{\text{cvg}} \)
Alternate Projections Algorithm

\[ \mathcal{X} = \prod_n \mathcal{X}_n \quad \text{and} \quad \mathcal{Y} = \mathcal{Y}_p = \{ \mathbf{x} \in \mathbb{R}^{NT} \mid \sum_{n \in N} x_n = p \} \]

Require: \( y^{(0)} \), \( k = 0 \), \( \varepsilon_{\text{cvg}} \), \( \| \cdot \| \)

repeat
\[
\begin{align*}
    \mathbf{x}^{(k+1)} & \leftarrow P_{\mathcal{X}}(y^{(k)}) \\
    \mathbf{y}^{(k+1)} & \leftarrow P_{\mathcal{Y}}(\mathbf{x}^{(k+1)}) \\
    k & \leftarrow k + 1
\end{align*}
\]
until \( \| y^{(k)} - y^{(k-1)} \| < \varepsilon_{\text{cvg}} \)
Alternate Projections Algorithm

\[ \mathcal{X} = \prod_n \mathcal{X}_n \quad \text{and} \quad \mathcal{Y} = \mathcal{Y}_p = \{ \mathbf{x} \in \mathbb{R}^{NT} \mid \sum_{n \in \mathcal{N}} x_n = p \} \]

Require: \( y^{(0)} \), \( k = 0 \), \( \varepsilon_{\text{cvg}} \), \( \| \cdot \| \)

repeat
\[
\begin{align*}
\mathbf{x}^{(k+1)} &\leftarrow P_{\mathcal{X}}(y^{(k)}) \\
\mathbf{y}^{(k+1)} &\leftarrow P_{\mathcal{Y}}(\mathbf{x}^{(k+1)}) \\
k &\leftarrow k + 1
\end{align*}
\]
until \( \| y^{(k)} - y^{(k-1)} \| < \varepsilon_{\text{cvg}} \)
Alternate Projections Algorithm

\[ x = \prod_n x_n \quad \text{and} \quad \mathcal{Y} = \mathcal{Y}_p = \{ x \in \mathbb{R}^{NT} \mid \sum_{n \in \mathcal{N}} x_n = p \} \]

**Require:** \( y^{(0)}, k = 0, \varepsilon_{\text{cvg}}, \| \cdot \| \)

repeat

\[
\begin{align*}
x^{(k+1)} & \leftarrow P_{\mathcal{X}}(y^{(k)}) \\
y^{(k+1)} & \leftarrow P_{\mathcal{Y}}(x^{(k+1)}) \\
k & \leftarrow k + 1
\end{align*}
\]

until \( \| y^{(k)} - y^{(k-1)} \| < \varepsilon_{\text{cvg}} \)
\( \mathcal{X} = \prod_n \mathcal{X}_n \quad \text{and} \quad \mathcal{Y} = \mathcal{Y}_p = \{ \mathbf{x} \in \mathbb{R}^{NT} \mid \sum_{n \in \mathcal{N}} x_n = p \} \)

**Require:** \( y^{(0)}, k = 0, \varepsilon_{\text{cvg}}, \| \cdot \| \)

repeat
\( x^{(k+1)} \leftarrow P_{\mathcal{X}}(y^{(k)}) \)
\( y^{(k+1)} \leftarrow P_{\mathcal{Y}}(x^{(k+1)}) \)
\( k \leftarrow k + 1 \)
until \( \| y^{(k)} - y^{(k-1)} \| < \varepsilon_{\text{cvg}} \)
Alternate Projections Algorithm

\[ x = \prod_n x_n \quad \text{and} \quad \mathcal{Y} = \mathcal{Y}_p = \{ x \in \mathbb{R}^{NT} \mid \sum_{n \in N} x_n = p \} \]

**Require:** \( y^{(0)}, k = 0, \varepsilon_{\text{cvg}}, \|\cdot\| \)

repeat
\[
\begin{align*}
x^{(k+1)} & \leftarrow P_x(y^{(k)}) \\
y^{(k+1)} & \leftarrow P_y(x^{(k+1)}) \\
k & \leftarrow k + 1
\end{align*}
\]
until \( \|y^{(k)} - y^{(k-1)}\| < \varepsilon_{\text{cvg}} \)
\[ x = \prod_n x_n \quad \text{and} \quad \mathcal{Y} = \mathcal{Y}_p = \{ x \in \mathbb{R}^{NT} \mid \sum_{n \in \mathcal{N}} x_n = p \} \]

**Require:** \( y^{(0)} \), \( k = 0 \), \( \varepsilon_{\text{cvg}} \), \( \| . \| \)

repeat

\[
\begin{align*}
x^{(k+1)} & \leftarrow P_x(y^{(k)}) \\
y^{(k+1)} & \leftarrow P_{\mathcal{Y}}(x^{(k+1)}) \\
k & \leftarrow k + 1
\end{align*}
\]

until \( \| y^{(k)} - y^{(k-1)} \| < \varepsilon_{\text{cvg}} \)
Theorem (Gubin, Polyak, 1967)

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

$$x^{(k)} \xrightarrow[k \to \infty]{} x^\infty, \quad y^{(k)} \xrightarrow[k \to \infty]{} y^\infty; \quad (3a)$$

$$\|x^\infty - y^\infty\|_2 = \min_{x \in \mathcal{X}, y \in \mathcal{Y}} \|x - y\|_2. \quad (3b)$$

In particular, if $\mathcal{X} \cap \mathcal{Y} \neq \emptyset$, then $(x^{(k)})_k$ and $(y^{(k)})_k$ converge to a same point $x^\infty \in \mathcal{X} \cap \mathcal{Y}$. 
For the sets $\mathcal{X}$ and $\mathcal{Y}$ defined above, and if $\mathcal{X} \cap \mathcal{Y} = \emptyset$, the following sets given by the limit orbit $(x^\infty, y^\infty)$ defined in Theorem 2:

$$T_0 \triangleq \{ t | p(t) > \sum_{n \in \mathbb{N}} x_n^\infty, t \} \quad (4a)$$

$$N_0 \triangleq \{ n | E_n - \sum_{t \in T_0} x_n, t - \sum_{t \in T_0} x_n, t < 0 \} \quad (4b)$$

This cut can be reformulated in terms of $1^\top N x^\infty$ as:

$$A_{T_0} < \sum_{t \in T_0} p(t) \quad (6)$$

$A_{T_0} \triangleq \sum_{t \in T_0} \sum_{n \in \mathbb{N}} x_n^\infty, t$. 

Paulin Jacquot (EDF - Inria - CMAP)
Theorem (Cut generation from APM limit iterates)

For the sets $X$ and $Y$ defined above, and if $X \cap Y = \emptyset$, the following sets given by the limit orbit $(x^\infty, y^\infty)$ defined in Theorem 2:

\[ T_0 \overset{\text{def}}{=} \{ t \mid p_t > \sum_{n \in \mathcal{N}} x^\infty_{n,t} \} \quad (4a) \]

\[ N_0 \overset{\text{def}}{=} \{ n \mid E_n - \sum_{t \in T_0} x^n_{n,t}, t - \sum_{t \in T_0} x^n_{n,t} < 0 \} \quad (4b) \]
Theorem (Cut generation from APM limit iterates)

For the sets $\mathcal{X}$ and $\mathcal{Y}$ defined above, and if $\mathcal{X} \cap \mathcal{Y} = \emptyset$, the following sets given by the limit orbit $(\mathbf{x}^\infty, \mathbf{y}^\infty)$ defined in Theorem 2:

$$\mathcal{T}_0 \overset{\text{def}}{=} \{ t \mid p_t > \sum_{n \in \mathcal{N}} x_{n,t}^\infty \} \quad (4a)$$

$$\mathcal{N}_0 \overset{\text{def}}{=} \{ n \mid E_n - \sum_{t \notin \mathcal{T}_0} x_{n,t} - \sum_{t \in \mathcal{T}_0} x_{n,t} < 0 \} \quad (4b)$$
Theorem (Cut generation from APM limit iterates)

For the sets $X$ and $Y$ defined above, and if $X \cap Y = \emptyset$, the following sets given by the limit orbit $(x^\infty, y^\infty)$ defined in Theorem 2:

$$T_0 \overset{\text{def}}{=} \{ t \mid p_t > \sum_{n \in N} x_{n,t}^\infty \}$$  \hspace{1cm} (4a) \\

$$N_0 \overset{\text{def}}{=} \{ n \mid E_n - \sum_{t \notin T_0} x_{n,t} - \sum_{t \in T_0} \overline{x}_{n,t} < 0 \}$$  \hspace{1cm} (4b) \\

define a "Hoffman cut" violated by $p$, that is:

$$\sum_{n \in N_0} E_n + \sum_{t \in T_0, n \notin N_0} \overline{x}_{n,t} - \sum_{t \notin T_0, n \in N_0} x_{n,t} < \sum_{t \in T_0} p_t.$$  \hspace{1cm} (5)
Theorem (Cut generation from APM limit iterates)

For the sets $\mathcal{X}$ and $\mathcal{Y}$ defined above, and if $\mathcal{X} \cap \mathcal{Y} = \emptyset$, the following sets given by the limit orbit $(x^\infty, y^\infty)$ defined in Theorem 2:

$$
\mathcal{T}_0 \text{ def } = \{ t \mid p_t > \sum_{n \in \mathcal{N}} x_{n,t}^\infty \} \quad (4a)
$$

$$
\mathcal{N}_0 \text{ def } = \{ n \mid E_n - \sum_{t \notin \mathcal{T}_0} x_{n,t} - \sum_{t \in \mathcal{T}_0} x_{n,t} < 0 \} \quad (4b)
$$

define a “Hoffman cut” violated by $p$, that is:

$$
\sum_{n \in \mathcal{N}_0} E_n + \sum_{t \in \mathcal{T}_0, n \notin \mathcal{N}_0} x_{n,t} - \sum_{t \notin \mathcal{T}_0, n \in \mathcal{N}_0} x_{n,t} < \sum_{t \in \mathcal{T}_0} p_t . \quad (5)
$$

This cut can be reformulated in terms of $1^\top_N x^\infty$ as:

$$
A_{\mathcal{T}_0} < \sum_{t \in \mathcal{T}_0} p_t \quad \text{with} \quad A_{\mathcal{T}_0} \text{ def } = \sum_{t \in \mathcal{T}_0} \sum_{n \in \mathcal{N}} x_{n,t}^\infty . \quad (6)
$$
Linear convergence of APM in our case

**Theorem**

For the sets $\mathcal{X}$ and $\mathcal{Y}$ defined above, the two subsequences of AP $(x^{(k)})_k$ and $(y^{(k)})_k$ converge at a geometric rate to $x^\infty \in \mathcal{X}$, $y^\infty \in \mathcal{Y}$, with:

$$\|x^{(k)} - x^\infty\|_2 \leq 2\|x^{(0)} - x^\infty\|_2 \times \rho_N^k$$

where $\rho_N \overset{\text{def}}{=} 1 - \frac{4}{N(T + 1)^2(T - 1)} < 1$

Same inequalities hold for the convergence of $y^{(k)}$ to $y^\infty$. 
Lemma (Nishihara et al, 2014)

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

$$c_F(U, V) = \sup\{u^T v \mid \|u\| \leq 1, \|v\| \leq 1$$
$$u \in U \cap (U \cap V)^\perp, v \in V \cap (U \cap V)^\perp\}.$$
Lemma (Nishihara et al, 2014)

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

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

Lemma (Nishihara et al, 2014)

Let $A$ and $B$ be matrices with orthonormal rows and with equal numbers of columns and $\Lambda_{sv}(AB^T)$ the set of singular values of $AB^T$. Then:

- if $\Lambda_{sv}(AB^T) = \{1\}$, then $c_F(\text{Ker}(A), \text{Ker}(B)) = 0$ ;
- Otherwise, $c_F(\text{Ker}(A), \text{Ker}(B)) = \max_{\lambda < 1}\{\lambda \in \Lambda_{sv}(AB^T)\}.$
Convergence rate: sketch of proof - 2:

\[ \mathcal{Y} \text{ is affine subspace } \mathcal{Y} = \{ x \in \mathbb{R}^{NT} \mid Ax = \sqrt{N}^{-1} 1_T \} \text{ with } \overrightarrow{\mathcal{Y}} = \text{Ker}(A) \text{ and } A \overset{\text{def}}{=} \sqrt{N}^{-1} J_{1,N} \otimes I_T. \]
Convergence rate: sketch of proof - 2:

- \( \mathcal{Y} \) is affine subspace \( \mathcal{Y} = \{ x \in \mathbb{R}^{N_T} | Ax = \sqrt{N}^{-1} 1_T \} \) with \( \overrightarrow{y} = \text{Ker}(A) \) and 
  \( A \overset{\text{def}}{=} \sqrt{N}^{-1} J_{1,N} \otimes I_T. \)
- Faces of \( \mathcal{X} \) are subsets of the collection of affine subspaces indexed by 
  \( (\overline{T}_n, \underline{T}_n) \subset \mathcal{T}^N \) (with \( \overline{T} \cap \underline{T} = \emptyset \)):
  \[ A(\overline{T}_n, \underline{T}_n) \overset{\text{def}}{=} \{ (x)_{nt} | \forall n, \ x_n^T 1_T = E_n \text{ and } \forall t \in \overline{T}_n, x_n,t = \underline{x}_n,t, \text{ and } \forall t \in \underline{T}_n, x_n,t = \overline{x}_n,t \}. \]

  Direction is \( \text{Ker}(B) \), with \( [B]_{[N]} \overset{\text{def}}{=} \sqrt{T}^{-1} I_N \otimes J_{1,T}. \)
Convergence rate: sketch of proof - 2:

- $\mathcal{Y}$ is affine subspace $\mathcal{Y} = \{ x \in \mathbb{R}^{NT} | Ax = \sqrt{N^{-1}} 1_T \}$ with $\overrightarrow{\mathcal{Y}} = \text{Ker}(A)$ and $A \overset{\text{def}}{=} \sqrt{N^{-1}} J_{1,N} \otimes I_T$.

- Faces of $\mathcal{X}$ are subsets of the collection of affine subspaces indexed by $(\mathcal{T}_n, \mathcal{T}_n)_n \subset \mathcal{T}^N$ (with $\mathcal{T} \cap \mathcal{T} = \emptyset$):
  
  \[
  \mathcal{A}(\mathcal{T}_n, \mathcal{T}_n)_n \overset{\text{def}}{=} \{ (x)_{nt} | \forall n, \ x_n^T 1_T = E_n \text{ and } \forall t \in \mathcal{T}_n, x_{n,t} = x_{n,t}, \text{ and } \forall t \in \mathcal{T}_n, x_{n,t} = \bar{x}_{n,t} \}.
  \]

  Direction is $\text{Ker}(B)$, with $[B]_N \overset{\text{def}}{=} \sqrt{T^{-1}} I_N \otimes J_{1,T}$.

- We denote by $K_n \overset{\text{def}}{=} \text{card}(\mathcal{T}_n)$. Renormalizing $B$, we show:

  \[
  S := (AB^\top)(A^\top B) = \frac{1}{N} \left( \sum_n \frac{1}{T - K_n} \right)_{k,\ell} + \frac{1}{N} \sum_{1 \leq t \leq T} \left( \sum_n 1_{t \in \mathcal{T}_n} \right) E_{t,t}.
  \]
Convergence rate: sketch of proof - 2:

- $\mathcal{Y}$ is affine subspace $\mathcal{Y} = \{ x \in \mathbb{R}^{NT} \mid Ax = \sqrt{N}^{-1}1_T \}$ with $\overrightarrow{\mathcal{Y}} = \text{Ker}(A)$ and $A \overset{\text{def}}{=} \sqrt{N}^{-1} J_{1,N} \otimes I_T$.

- Faces of $\mathcal{X}$ are subsets of the collection of affine subspaces indexed by $(\overline{T}_n, \underline{T}_n)_n \subset \mathcal{T}^N$ (with $\overline{T} \cap \underline{T} = \emptyset$):

  $$\mathcal{A}_{(\overline{T}_n, \underline{T}_n)} \overset{\text{def}}{=} \left\{ (x)_{nt} \mid \forall n, \ x_n^T 1_T = E_n \text{ and } \forall t \in \overline{T}_n, x_{n,t} = x_{n,t}, \text{ and } \forall t \in \underline{T}_n, x_{n,t} = \overline{x}_{n,t} \right\}.$$ 

- Direction is $\text{Ker}(B)$, with $[B][N] \overset{\text{def}}{=} \sqrt{T}^{-1} I_N \otimes J_{1,T}$.

- We denote by $K_n \overset{\text{def}}{=} \text{card}(\mathcal{T}_n)$. Renormalizing $B$, we show:

  $$S := (AB^\top)(A^\top B) = \frac{1}{N} \left( \sum_n \frac{1}{T} \right) \left( \mathcal{A}_{(\overline{T}_n, \underline{T}_n)} \right) + \frac{1}{N} \sum_{1 \leq t \leq T} \left( \sum_n 1_{t \in \mathcal{T}_n} \right) E_{t,t}.$$ 

- Denote $\overline{T} \overset{\text{def}}{=} \bigcup_n T_n^c$ and $P \overset{\text{def}}{=} I_T - S$. Then $P = \text{diag}(P_{\overline{T}}, 0_{\overline{T}^c})$ 
   → restrict to $\text{Vect}(e_t)_{t \in \overline{T}}$ to find $\lambda_1(P)$ (least positive eigval)
Consider graph $\mathcal{G} = (\bar{T}, \mathcal{E})$ whose vertices set is $\bar{T}$ and edge $(k, \ell)$ has weight $S_{k,\ell} = \frac{1}{N} \sum_n \frac{1_{\{k,\ell\} \subseteq T_n^c}}{T_n^c - K_n}$. One can show that $\sum_{\ell \neq k} -P_{k,\ell} = P_{kk} \rightarrow P$ is Laplacian matrix of $\mathcal{G}$.
Consider graph $G = (\bar{T}, \mathcal{E})$ whose vertices set is $\bar{T}$ and edge $(k, \ell)$ has weight $S_{k,\ell} = \frac{1}{N} \sum_{n} \mathbb{1}_{\{k,\ell\} \in \mathcal{T}_n} \frac{1}{T-K_n}$. One can show that $\sum_{\ell \neq k} -P_{k,\ell} = P_{kk} \rightarrow P$ is Laplacian matrix of $G$.

Using Laplacian property and Cauchy-Schwartz, $\forall u \perp 1$:

$$u^\top Pu \geq \min_{k, \ell \in (s^*-t^*)} (-P_{k,\ell}) \frac{(u_{t^*} - u_{s^*})^2}{d_{s^*,t^*}} \geq \frac{4T\|u\|_2^2}{N(T+1)^2(T-1)^2}$$

where $u_{t^*} := \max_t u_t$, $u_{s^*} := \min_t u_t$ and $d_{s^*,t^*}$ distance in $G$, and $(s^*-t^*)$ a path from $s^*$ to $t^*$. 

Convergence rate: sketch of proof - 3:
Consider graph $\mathcal{G} = (\bar{T}, \mathcal{E})$ whose vertices set is $\bar{T}$ and edge $(k, \ell)$ has weight $S_{k,\ell} = \frac{1}{N} \sum_{n} \frac{1_{\{k,\ell\} \subset \mathcal{T}_n}}{T-K_n}$. One can show that $\sum_{\ell \neq k} -P_{k,\ell} = P_{kk} \rightarrow P$ is Laplacian matrix of $\mathcal{G}$.

Using Laplacian property and Cauchy-Schwartz, $\forall u \perp 1$:

$$u^\top Pu \geq \min_{k,\ell \in (s^*-t^*)} (-P_{k,\ell}) \frac{(u_{t^*} - u_{s^*})^2}{d_{s^*,t^*}} \geq \frac{4T\|u\|^2}{N(T+1)^2(T-1)^2}$$

where $u_{t^*} := \max_t u_t$, $u_{s^*} := \min_t u_t$ and $d_{s^*,t^*}$ distance in $\mathcal{G}$, and $(s^*-t^*)$ a path from $s^*$ to $t^*$.

As $1$ is an eigenvector of $P$ associated to $\lambda_0 = 0$, from the minmax theorem, we get $\lambda_1(P) \geq \frac{4}{N(T+1)^2(T-1)} := 1 - \rho NT$.
back to the two subproblems...

**Master Problem**

\[
\min_{p \in \mathbb{R}^T} f(p)
\]

s.t. \( p \in \mathcal{P}(s) \)

**Disaggregation Problem**

\[
\text{Find } x \in \mathcal{Y}_p \cap (\prod_n \mathcal{X}_n)
\]
back to the two subproblems...

**Master Problem**

\[
\min_{p \in \mathbb{R}^T} f(p)
\]

s.t. \( p \in \mathcal{P}(s) \)

**Disaggregation Problem**

**Find** \( x \in \mathcal{Y}_p \cap (\prod_n \mathcal{X}_n) \)
back to the two subproblems...

**Master Problem**
\[
\min_{p \in \mathbb{R}^T} f(p)
\]
\[
\text{s.t. } p \in \mathcal{P}^{(s)}
\]

**Disaggregation Problem**
\[
\text{Find } x \in \mathcal{Y}_p \cap (\prod_n \mathcal{X}_n)
\]

Paulin Jacquot (EDF - Inria - CMAP)  
Privacy-Preserving Disaggregation Algo  
Lundi 7 octobre 2019  14 / 21
back to the two subproblems...

**Master Problem**

\[
\min_{p \in \mathbb{R}^T} f(p) \\
\text{s.t. } p \in \mathcal{P}^{(s)}
\]

**Disaggregation Problem**

Find \( x \in \mathcal{Y}_p \cap (\prod_n \mathcal{X}_n) \)

\[
\mathcal{P}^{(s+1)} = \mathcal{P}^{(s)} \cap \{ p \mid A_{\mathcal{T}_0} < \sum_{t \in \mathcal{T}_0} p_t \}
\]
To non-intrusive projections...

- projections $x_n = P_{X_n}(y_n)$ can be computed locally;
To non-intrusive projections...

- projections $x_n = P_{x_n}(y_n)$ can be computed locally;
- projection $y = P_{\mathcal{Y}_p}(x)$ on $\mathcal{Y}_p = \{x \in \mathbb{R}^{NT} \mid \sum_n x_n = p\}$ is explicit (affine space):
  \[\forall n, \quad y_n = x_n + \frac{1}{N} \left( p - (\sum_m x_m) \right) ;\]
To non-intrusive projections...

- projections $x_n = P_{X_n}(y_n)$ can be computed locally;
- projection $y = P_{Y_p}(x)$ on $Y_p = \{ x \in \mathbb{R}^{NT} \mid \sum_n x_n = p \}$ is explicit (affine space):
  \[
  \forall n, \quad y_n = x_n + \frac{1}{N} \left( p - (\sum_m x_m) \right);
  \]
- only requires the aggregate quantity $\sum_n x_n$, then local operations;
To non-intrusive projections...

- projections \( x_n = P_{\mathcal{X}_n}(y_n) \) can be computed locally;
- projection \( y = P_{\mathcal{Y}_p}(x) \) on \( \mathcal{Y}_p = \{ x \in \mathbb{R}^{NT} \mid \sum_n x_n = p \} \) is explicit (affine space):
  \[
  \forall n, \quad y_n = x_n + \frac{1}{N} \left( p - (\sum_m x_m) \right);
  \]
- only requires the aggregate quantity \( \sum_n x_n \), then local operations;

How can we compute \( \sum_n x_n \) without disclosing profiles to Big Brother?
Issues: transmission of profiles for projection

In APM, agents still have to provide profiles \((x_n^{(k)})_n\)

→ **Secure Multiparty Computation** (SMC) principle

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

1: **for** each agent \(n \in \mathcal{N} \) **do**
2: Draw \(\forall t, (s_{n,t,m})_{m=1}^{N-1} \in \mathcal{U}([0, A]^{N-1})\)
3: and set \(\forall t, s_{n,t,N} \overset{\text{def}}{=} x_{n,t} - \sum_{m=1}^{N-1} s_{n,t,m}\)
4: Send \((s_{n,t,m})_{t \in T} \) to agent \(m \in \mathcal{N}\)
5: **done**

6: **for** each agent \(n \in \mathcal{N} \) **do**
7: Compute \(\forall t, \sigma_{n,t} = \sum_{m \in \mathcal{N}} s_{m,t,n}\)
8: Send \((\sigma_{n,t})_{t \in T} \) to operator
9: **done**
10: Operator computes \(S = \sum_{n \in \mathcal{N}} \sigma_n\)
Issues: transmission of profiles for projection

In APM, agents still have to provide profiles \((x_n^{(k)})_n\)

→ **Secure Multiparty Computation** (SMC) principle

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

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

In APM, agents still have to provide profiles \( (x_n^{(k)})_n \)

→ **Secure Multiparty Computation** (SMC) principle

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

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

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

1. **while** Not $\text{DISAG}$  **do**
Combining all elements...

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

1: while Not \( \text{DISAG} \) do
2: Compute \( \mathbf{p}^{(s)} = \arg \min_{\mathbf{p} \in \mathcal{P}_{cs}^{(s)}} f(\mathbf{p}) \)

---

**Master Problem**

\[
\begin{align*}
\min_{\mathbf{p} \in \mathbb{R}^T} & \quad f(\mathbf{p}) \\
\text{s.t.} & \quad \mathbf{p} \in \mathcal{P}^{(s)}
\end{align*}
\]
Combining all elements...

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

1: **while** Not \( \text{DISAG} \) **do**
2: \hspace{1em} Compute \( p^{(s)} = \arg \min_{p \in \mathcal{P}_{cs}^{(s)}} p \)
3: \hspace{1em} \( \text{DISAG} \leftarrow \text{APM}(p^{(s)}) \)

**Master Problem**

\[
\min_{p \in \mathbb{R}^T} f(p)
\]

s.t. \( p \in \mathcal{P}^{(s)} \)

**Disaggregation Pb**

Find \( x \in \mathcal{Y}_p \cap (\prod_n \mathcal{X}_n) \)
Combining all elements...

Require: $s = 0$, $\mathcal{P}^{(0)} = \mathcal{P}$; Disag = False
1: while Not Disag do
2: Compute $p^{(s)} = \arg\min_{p \in \mathcal{P}_{cs}^{(s)}} p$
3: Disag ← APM($p^{(s)}$)
4: if Disag then
5: Operator adopts $p^{(s)}$
6: else
7: Obtain $T^{(s)}$ from APM($p^{(s)}$)
8: $\mathcal{P}^{(s+1)}$ ← $\mathcal{P}^{(s)} \cap \{ p \mid \sum_{t \in T^{(s)}} p_t \leq A^{(s)}_T \}$
9: end
10: $s$ ← $s + 1$
11: done
Combining all elements...

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

1. while Not $\text{DISAG}$ do
2. Compute $p^{(s)} = \arg \min_{p \in \mathcal{P}_{cs}^{(s)}} p$
3. $\text{DISAG} \leftarrow \text{APM}(p^{(s)})$
4. if $\text{DISAG}$ then
5. Operator adopts $p^{(s)}$
6. else
7. Obtain $\mathcal{T}^{(s)}_0, A^{(s)}_{\mathcal{T}^{(s)}_0}$ from $\text{APM}(p^{(s)})$
8. $\mathcal{P}^{(s+1)} \leftarrow \mathcal{P}^{(s)} \cap \{ p | \sum_{t \in \mathcal{T}^{(s)}_0} p_t \leq A^{(s)}_{\mathcal{T}^{(s)}_0} \}$
9. end

**Master Problem**

$$\min_{p \in \mathbb{R}^T} f(p) \quad \text{s.t. } p \in \mathcal{P}^{(s)}$$

**Disaggregation Pb**

Find $x \in \mathcal{Y}_p \cap (\prod_n x_n)$
Combining all elements...

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

1: while Not $\text{DISAG}$ do
2: Compute $\mathbf{p}^{(s)} = \arg \min_{\mathbf{p} \in \mathcal{P}_{\text{cs}}^{(s)}} \mathbf{p}$
3: $\text{DISAG} \leftarrow \text{APM}(\mathbf{p}^{(s)})$
4: if $\text{DISAG}$ then
5: Operator adopts $\mathbf{p}^{(s)}$
6: else
7: Obtain $\mathcal{T}_0^{(s)}$, $A_{\mathcal{T}_0}^{(s)}$ from APM($\mathbf{p}^{(s)}$)
8: $\mathcal{P}^{(s+1)} \leftarrow \mathcal{P}^{(s)} \cap \{ \mathbf{p} \mid \sum_{t \in \mathcal{T}_0^{(s)}} p_t \leq A_{\mathcal{T}_0}^{(s)} \}$
9: end
10: $s \leftarrow s + 1$
11: done

Master Problem

$$\begin{align*}
\min_{\mathbf{p} \in \mathbb{R}^T} & \quad f(\mathbf{p}) \\
\text{s.t.} & \quad \mathbf{p} \in \mathcal{P}^{(s)}
\end{align*}$$

Disaggregation Pb

Find $\mathbf{x} \in \mathcal{Y}_\mathbf{p} \cap (\prod_n \mathcal{X}_n)$
Proposition

The procedure stops after a finite number of iterations, as at most $2^T$ constraints can be added to the master problem.
Proposition

The procedure stops after a finite number of iterations, as at most $2^T$ constraints can be added to the master problem.

**Issue**: we need the limit $x^\infty$ of the APM sequence to obtain the cut.
**Termination condition: number of cuts**

**Proposition**

The procedure stops after a finite number of iterations, as at most $2^T$ constraints can be added to the master problem.

**Issue:** we need the limit $x^\infty$ of the APM sequence to obtain the cut. but in practice we can stop in finite time and obtain the approximated same cut!
Illustrative example in dimension $T = 4$ (with $\sum_t p_t = \sum_n E_n$)

**Master Problem**

$$\min_{\mathbf{p} \in \mathbb{R}^T} f(\mathbf{p})$$

s.t. $\mathbf{p} \in \mathcal{P}^{(s)}$

**Disaggregation Problem**

Find $\mathbf{x} \in \mathcal{Y}_\mathbf{p} \cap (\prod_n \mathcal{X}_n)$
Illustrative example in dimension \( T = 4 \) (with \( \sum_t p_t = \sum_n E_n \))

\[
\begin{aligned}
\text{Master Problem} & \\
\min_{p \in \mathbb{R}^T} f(p) \\
\text{s.t. } p & \in \mathcal{P}^{(s)}
\end{aligned}
\]

\[
\begin{aligned}
\text{Disaggregation Pb} & \\
\text{Find } x \in \mathcal{Y}_p \cap (\prod_n \mathcal{X}_n)
\end{aligned}
\]
Illustrative example in dimension $T = 4$ (with $\sum_t p_t = \sum_n E_n$)

Master Problem

$$\min_{p \in \mathbb{R}^T} f(p)$$

subject to $p \in \mathcal{P}^{(s)}$

Disaggregation Problem

Find $x \in \mathcal{Y}_p \cap (\prod_n \chi_n)$
Illustrative example in dimension \( T = 4 \) (with \( \sum_t p_t = \sum_n E_n \))

**Master Problem**

\[
\begin{align*}
\min_{p \in \mathbb{R}^T} & \quad f(p) \\
\text{s.t.} & \quad p \in \mathcal{P}^{(s)}
\end{align*}
\]

**Disaggregation Pb**

Find \( x \in \mathcal{Y}_p \cap (\prod_n \mathcal{X}_n) \)

Feasible!
Illustrative example in dimension $T = 4$ (with $\sum_t p_t = \sum_n E_n$)

**Master Problem**

$$\min_{p \in \mathbb{R}^T} f(p)$$

s.t. $p \in \mathcal{P}(s)$

**Disaggregation Problem**

Find $x \in \mathcal{Y}_p \cap (\prod_n \mathcal{X}_n)$

Feasible!
The method computes a resource allocation $p$ and $N$ individual agents profiles $(x_n)_n$, such that $(x, p)$ solves the global (nonconvex) problem, while keeping private:

Further work and extensions: a variant of the algorithm can deal individual constraints $X_n \rightarrow$ arbitrary (polyhedral) set, resolving local LPs to get a cut; can we generalize the direct obtention of the cut for other polyhedra? analysis on the maximal number of constraints added (polynomial bound?).
The method computes a resource allocation $p$ and $N$ individual agents profiles $(x_n)_n$, such that $(x, p)$ solves the global (nonconvex) problem, while keeping private:

1. agent constraints (upper bounds $\bar{x}_n$, lower bounds $\underline{x}_n$, demand $E_n$);
Conclusion

The method computes a resource allocation $p$ and $N$ individual agents profiles $(x_n)_n$, such that $(x, p)$ solves the global (nonconvex) problem, while keeping private:

1. agent constraints (upper bounds $\overline{x}_n$, lower bounds $\underline{x}_n$, demand $E_n$);
2. individual profile $x_n$ (as well as the iterates $(x^{(k)})_k$ and $(y^{(k)})_k$ in APM).

Further work and extensions:

- A variant of the algorithm can deal with individual constraints $x_n \rightarrow$ arbitrary (polyhedral) set, resolving local LPs to get a cut;
- Can we generalize the direct obtention of the cut for other polyhedra?
- Analysis on the maximal number of constraints added (polynomial bound?).

Paulin Jacquot (EDF - Inria - CMAP)
Conclusion

The method computes a resource allocation $\mathbf{p}$ and $N$ individual agents profiles $(\mathbf{x}_n)_n$, such that $(\mathbf{x}, \mathbf{p})$ solves the global (nonconvex) problem, while keeping private:

1. agent constraints (upper bounds $\overline{x}_n$, lower bounds $\underline{x}_n$, demand $E_n$);
2. individual profile $\mathbf{x}_n$ (as well as the iterates $(\mathbf{x}^{(k)})_k$ and $(\mathbf{y}^{(k)})_k$ in APM).

Further work and extensions:
The method computes a resource allocation $\mathbf{p}$ and $N$ individual agents profiles $(\mathbf{x}_n)_n$, such that $(\mathbf{x}, \mathbf{p})$ solves the global (nonconvex) problem, while keeping private:

1. agent constraints (upper bounds $\overline{x}_n$, lower bounds $\underline{x}_n$, demand $E_n$);
2. individual profile $\mathbf{x}_n$ (as well as the iterates $(\mathbf{x}^{(k)})_k$ and $(\mathbf{y}^{(k)})_k$ in APM).

**Further work and extensions:**

- a variant of the algorithm can deal individual constraints $\mathcal{X}_n \rightarrow$ arbitrary (polyhedral) set, resolving local LPs to get a cut;
The method computes a resource allocation $p$ and $N$ individual agents profiles $(x_n)_n$, such that $(x,p)$ solves the global (nonconvex) problem, while keeping private:

1. agent constraints (upper bounds $\bar{x}_n$, lower bounds $\underline{x}_n$, demand $E_n$);
2. individual profile $x_n$ (as well as the iterates $(x^{(k)}_k)$ and $(y^{(k)}_k)$ in APM).

**Further work and extensions:**

- a variant of the algorithm can deal individual constraints $x_n \rightarrow$ arbitrary (polyhedral) set, resolving local LPs to get a cut;
- can we generalize the direct obtention of the cut for other polyhedra?
Conclusion

The method computes a resource allocation $p$ and $N$ individual agents profiles $(x_n)_n$, such that $(x, p)$ solves the global (nonconvex) problem, while keeping private:

1. agent constraints (upper bounds $\overline{x}_n$, lower bounds $\underline{x}_n$, demand $E_n$);
2. individual profile $x_n$ (as well as the iterates $(x^{(k)})_k$ and $(y^{(k)})_k$ in APM).

Further work and extensions:

- a variant of the algorithm can deal individual constraints $\mathcal{X}_n \rightarrow$ arbitrary (polyhedral) set, resolving local LPs to get a cut;
- can we generalize the direct obtention of the cut for other polyhedra ?
- analysis on the maximal number of constraints added (polynomial bound ?).
THANKS!
