# 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]

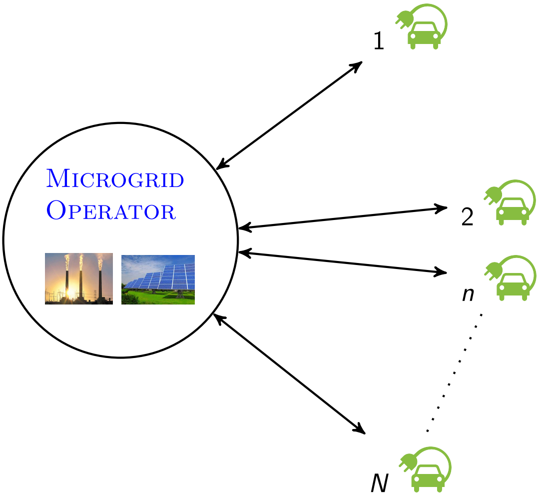
[1]EDF Lab Saclay

[2]Inria and CMAP, École polytechnique

Lundi 7 octobre 2019

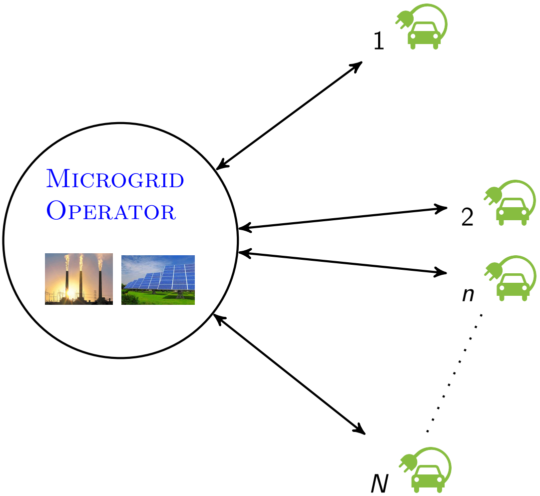
Journée de rentrée du **CMAP**

# Introduction and Context



# Introduction and Context

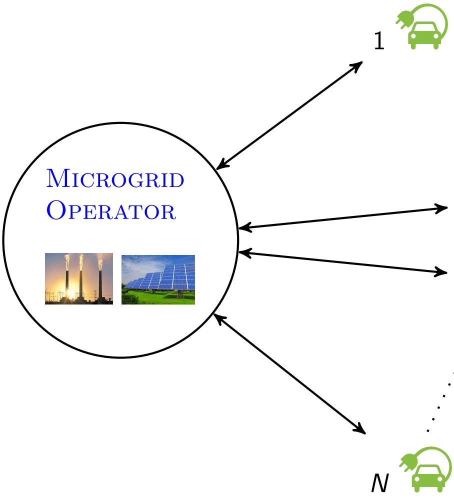
**Two main issues:**



# Introduction and Context

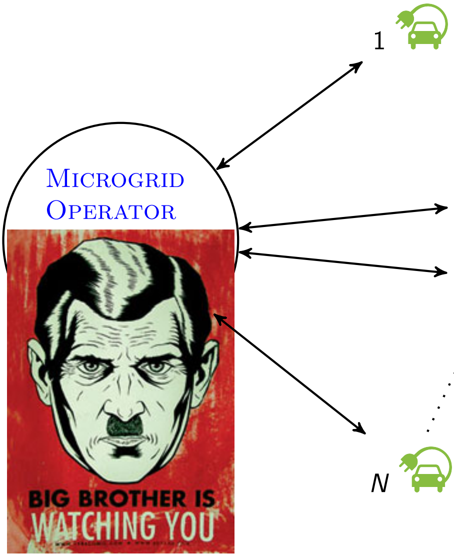
**Two main issues:**

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



**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.



# Problem Formulation

$$\min_{\boldsymbol{x}\in\mathbb{R}^{N\times T},\ \boldsymbol{p}\in\mathbb{R}^T} f(\boldsymbol{p}) \tag{1a}$$

$$\boldsymbol{p} \in \mathcal{P} \tag{1b}$$

$$\sum_{n\in\mathcal{N}} x_{n,t} = p_t,\ \forall t \in \mathcal{T} \tag{1c}$$

$$\boldsymbol{x}_n \in \mathcal{X}_n,\ \forall n \in \mathcal{N} \tag{1d}$$

# Problem Formulation

$$\min_{\boldsymbol{x}\in\mathbb{R}^{N\times T},\ \boldsymbol{p}\in\mathbb{R}^T} f(\boldsymbol{p}) \tag{1a}$$

$$\boldsymbol{p} \in \mathcal{P} \qquad \text{operator constraints} \tag{1b}$$

$$\sum_{n\in\mathcal{N}} x_{n,t} = p_t,\ \forall t \in \mathcal{T} \qquad \text{disaggregation} \tag{1c}$$

$$\boldsymbol{x}_n \in \mathcal{X}_n,\ \forall n \in \mathcal{N} \qquad \text{agents constraints} \tag{1d}$$

# Problem Formulation

$$\min_{\boldsymbol{x}\in\mathbb{R}^{N\times T},\ \boldsymbol{p}\in\mathbb{R}^T} f(\boldsymbol{p}) \tag{1a}$$

$$\boldsymbol{p} \in \mathcal{P} \qquad \text{operator constraints} \tag{1b}$$

$$\sum_{n\in\mathcal{N}} x_{n,t} = p_t,\ \forall t \in \mathcal{T} \qquad \text{disaggregation} \tag{1c}$$

$$\boldsymbol{x}_n \in \mathcal{X}_n,\ \forall n \in \mathcal{N} \qquad \text{agents constraints} \tag{1d}$$

with $\mathcal{X}_n \stackrel{\text{def}}{=} \left\{\boldsymbol{x}_n \in \mathbb{R}^T \mid \sum_t x_{n,t} = E_n \quad \text{and} \quad \forall t, \underline{x}_{n,t} \leq x_{n,t} \leq \overline{x}_{n,t}\right\}$

# Problem Formulation

$$\min_{\boldsymbol{x}\in\mathbb{R}^{N\times T},\ \boldsymbol{p}\in\mathbb{R}^T} f(\boldsymbol{p}) \tag{1a}$$

$$\boldsymbol{p} \in \mathcal{P} \quad \text{operator constraints} \tag{1b}$$

$$\sum_{n\in\mathcal{N}} x_{n,t} = p_t,\ \forall t \in \mathcal{T} \quad \text{disaggregation} \tag{1c}$$

$$\boldsymbol{x}_n \in \mathcal{X}_n,\ \forall n \in \mathcal{N} \quad \text{agents constraints} \tag{1d}$$

with $\mathcal{X}_n \overset{\text{def}}{=} \left\{\boldsymbol{x}_n \in \mathbb{R}^T \mid \sum_t x_{n,t} = E_n \quad \text{and} \quad \forall t, \underline{x}_{n,t} \leq x_{n,t} \leq \overline{x}_{n,t}\right\}$

RESSOURCE 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:

# Two subproblems

Our method considers two subproblems iteratively:

MASTER PROBLEM

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

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

# Two subproblems

Our method considers two subproblems iteratively:

**Master Problem**

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

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

**Disaggregation Problem**

$$\text{Find } \boldsymbol{x} = (\boldsymbol{x}_n)_{n\in\mathcal{N}} \in \mathcal{Y}_{\boldsymbol{p}} \cap \mathcal{X}$$

where $\mathcal{Y}_{\boldsymbol{p}} \stackrel{\text{def}}{=} \{\boldsymbol{y} \in \mathbb{R}^{NT} | \sum_{n\in\mathcal{N}} \boldsymbol{y}_n = \boldsymbol{p}\}$

and $\mathcal{X} \stackrel{\text{def}}{=} \prod_{n\in\mathcal{N}} \mathcal{X}_n$ .

# Two subproblems

Our method considers two subproblems iteratively:

**Master Problem**

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

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

$\boldsymbol{p}^{(s)}$ →

**Disaggregation Problem**

$$\text{Find } \boldsymbol{x} = (\boldsymbol{x}_n)_{n\in\mathcal{N}} \in \mathcal{Y}_{\boldsymbol{p}} \cap \mathcal{X}$$

$$\text{where } \mathcal{Y}_{\boldsymbol{p}} \overset{\text{def}}{=} \{\boldsymbol{y} \in \mathbb{R}^{NT} | \sum_{n\in\mathcal{N}} \boldsymbol{y}_n = \boldsymbol{p}\}$$

$$\text{and } \mathcal{X} \overset{\text{def}}{=} \prod_{n\in\mathcal{N}} \mathcal{X}_n \ .$$

# Two subproblems

Our method considers two subproblems iteratively:

**Master Problem**

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

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

$\boldsymbol{p}^{(s)}$ →

← $\mathcal{P}^{(s+1)}$

**Disaggregation Problem**

$$\text{Find } \boldsymbol{x} = (\boldsymbol{x}_n)_{n\in\mathcal{N}} \in \mathcal{Y}_{\boldsymbol{p}} \cap \mathcal{X}$$

$$\text{where } \mathcal{Y}_{\boldsymbol{p}} \stackrel{\text{def}}{=} \{\boldsymbol{y} \in \mathbb{R}^{NT} | \sum_{n\in\mathcal{N}} \boldsymbol{y}_n = \boldsymbol{p}\}$$

$$\text{and } \mathcal{X} \stackrel{\text{def}}{=} \prod_{n\in\mathcal{N}} \mathcal{X}_n .$$

# Disaggregation Feasibility

Characterizing $\mathcal{Y}_{\boldsymbol{p}} \cap \mathcal{X} = \{\boldsymbol{x} \in \mathcal{X} | \ \sum_{n\in\mathcal{N}} \boldsymbol{x}_n = \boldsymbol{p}\}$
Necessary aggregated constraints:

$$\sum_t p_t = \sum_n E_n \ \text{ and } \ \forall t, \ \sum_n \underline{x}_{n,t} \leq p_t \leq \sum_n \overline{x}_{n,t} \ .$$

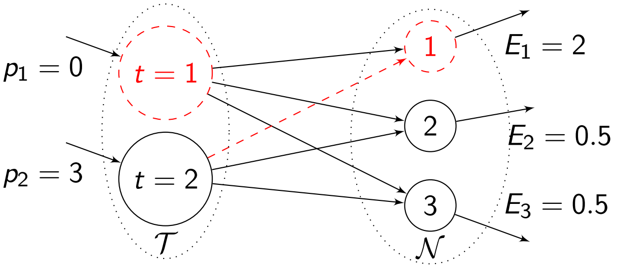
Not sufficient!

# Disaggregation Feasibility

Characterizing $\mathcal{Y}_{\boldsymbol{p}} \cap \mathcal{X} = \{\boldsymbol{x} \in \mathcal{X} | \sum_{n \in \mathcal{N}} \boldsymbol{x}_n = \boldsymbol{p}\}$
Necessary aggregated constraints:

$$\sum_t p_t = \sum_n E_n \quad \text{and} \quad \forall t, \ \sum_n \underline{x}_{n,t} \leq p_t \leq \sum_n \overline{x}_{n,t} \ .$$

Not sufficient!



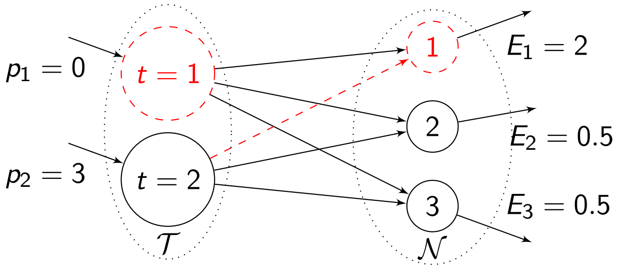
# Disaggregation Feasibility

Characterizing $\mathcal{Y}_{\boldsymbol{p}} \cap \mathcal{X} = \{\boldsymbol{x} \in \mathcal{X} | \sum_{n\in\mathcal{N}} \boldsymbol{x}_n = \boldsymbol{p}\}$

## Theorem (Hoffman Circulation's Theorem)

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

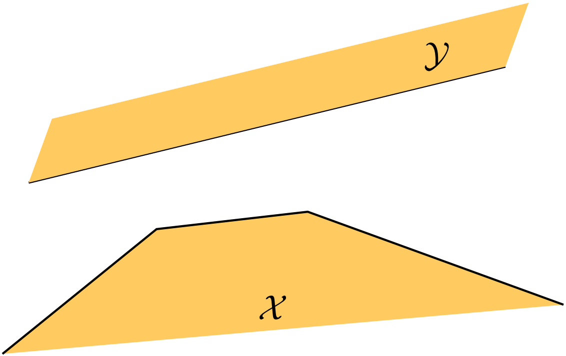
$$\sum_{t\notin\mathcal{T}_{\mathrm{in}}} p_t \leq \sum_{t\notin\mathcal{T}_{\mathrm{in}}, n\in\mathcal{N}_{\mathrm{in}}} \overline{x}_{n,t} - \sum_{t\in\mathcal{T}_{\mathrm{in}}, n\notin\mathcal{N}_{\mathrm{in}}} \underline{x}_{n,t} + \sum_{n\notin\mathcal{N}_{\mathrm{in}}} E_n. \quad (2)$$



# Alternate Projections Algorithm

$\mathcal{X} = \prod_n \mathcal{X}_n$ and $\mathcal{Y} = \mathcal{Y}_{\boldsymbol{p}} = \{\boldsymbol{x} \in \mathbb{R}^{NT} \mid \sum_{n \in \mathcal{N}} \boldsymbol{x}_n = \boldsymbol{p}\}$

**Require:** $\boldsymbol{y}^{(0)}$, $k = 0$ , $\varepsilon_{\text{cvg}}$, $\|.\|$
**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** $\left\|\boldsymbol{y}^{(k)} - \boldsymbol{y}^{(k-1)}\right\| < \varepsilon_{\text{cvg}}$



# Alternate Projections Algorithm

$\mathcal{X} = \prod_n \mathcal{X}_n$ and $\mathcal{Y} = \mathcal{Y}_{\boldsymbol{p}} = \{\boldsymbol{x} \in \mathbb{R}^{NT} \mid \sum_{n \in \mathcal{N}} \boldsymbol{x}_n = \boldsymbol{p}\}$

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

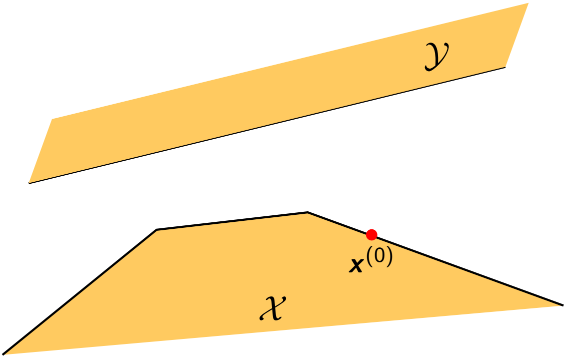
**repeat**

$\boldsymbol{x}^{(k+1)} \leftarrow P_{\mathcal{X}}(\boldsymbol{y}^{(k)})$

$\boldsymbol{y}^{(k+1)} \leftarrow P_{\mathcal{Y}}(\boldsymbol{x}^{(k+1)})$

$k \leftarrow k + 1$

**until** $\left\|\boldsymbol{y}^{(k)} - \boldsymbol{y}^{(k-1)}\right\| < \varepsilon_{\text{cvg}}$



# Alternate Projections Algorithm

$\mathcal{X} = \prod_n \mathcal{X}_n$ and $\mathcal{Y} = \mathcal{Y}_{\boldsymbol{p}} = \{\boldsymbol{x} \in \mathbb{R}^{NT} \mid \sum_{n \in \mathcal{N}} \boldsymbol{x}_n = \boldsymbol{p}\}$

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

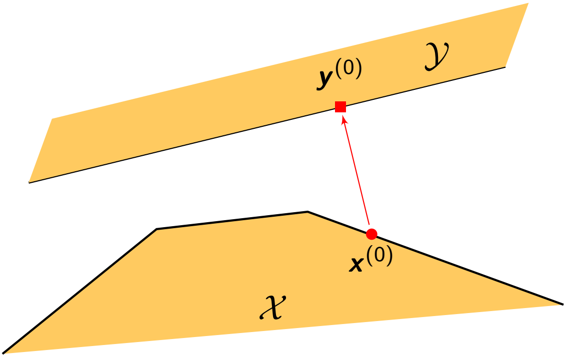
**repeat**

$\boldsymbol{x}^{(k+1)} \leftarrow P_{\mathcal{X}}(\boldsymbol{y}^{(k)})$

$\boldsymbol{y}^{(k+1)} \leftarrow P_{\mathcal{Y}}(\boldsymbol{x}^{(k+1)})$

$k \leftarrow k + 1$

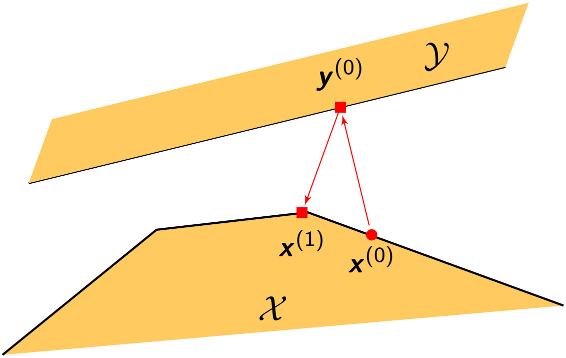
**until** $\left\|\boldsymbol{y}^{(k)} - \boldsymbol{y}^{(k-1)}\right\| < \varepsilon_{\text{cvg}}$



# Alternate Projections Algorithm

$\mathcal{X} = \prod_n \mathcal{X}_n$ and $\mathcal{Y} = \mathcal{Y}_{\boldsymbol{p}} = \{\boldsymbol{x} \in \mathbb{R}^{NT} \mid \sum_{n\in\mathcal{N}} \boldsymbol{x}_n = \boldsymbol{p}\}$

**Require:** $\boldsymbol{y}^{(0)}$, $k = 0$ , $\varepsilon_{\text{cvg}}$, $\|.\|$
  **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** $\left\|\boldsymbol{y}^{(k)} - \boldsymbol{y}^{(k-1)}\right\| < \varepsilon_{\text{cvg}}$



# Alternate Projections Algorithm

$\mathcal{X} = \prod_n \mathcal{X}_n$ and $\mathcal{Y} = \mathcal{Y}_{\boldsymbol{p}} = \{\boldsymbol{x} \in \mathbb{R}^{NT} \mid \sum_{n\in\mathcal{N}} \boldsymbol{x}_n = \boldsymbol{p}\}$

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

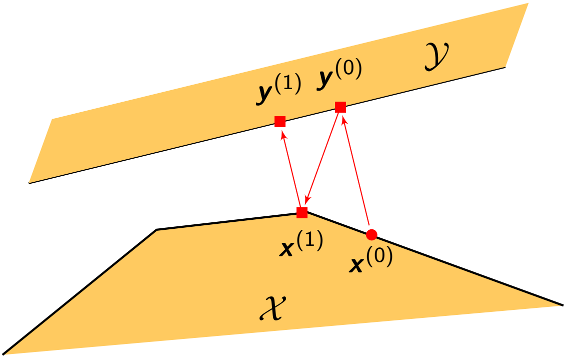
**repeat**

$\boldsymbol{x}^{(k+1)} \leftarrow P_{\mathcal{X}}(\boldsymbol{y}^{(k)})$

$\boldsymbol{y}^{(k+1)} \leftarrow P_{\mathcal{Y}}(\boldsymbol{x}^{(k+1)})$

$k \leftarrow k + 1$

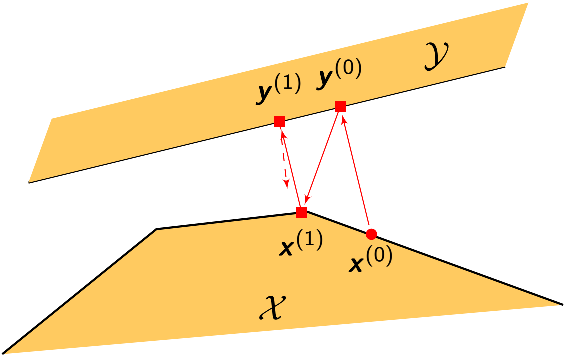
**until** $\left\|\boldsymbol{y}^{(k)} - \boldsymbol{y}^{(k-1)}\right\| < \varepsilon_{\text{cvg}}$



# Alternate Projections Algorithm

$\mathcal{X} = \prod_n \mathcal{X}_n$ and $\mathcal{Y} = \mathcal{Y}_{\boldsymbol{p}} = \{\boldsymbol{x} \in \mathbb{R}^{NT} \mid \sum_{n \in \mathcal{N}} \boldsymbol{x}_n = \boldsymbol{p}\}$

**Require:** $\boldsymbol{y}^{(0)}$, $k = 0$ , $\varepsilon_{\text{cvg}}$, $\|.\|$
  **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** $\left\|\boldsymbol{y}^{(k)} - \boldsymbol{y}^{(k-1)}\right\| < \varepsilon_{\text{cvg}}$



# Alternate Projections Algorithm

$$\mathcal{X} = \prod_n \mathcal{X}_n \quad \text{and} \quad \mathcal{Y} = \mathcal{Y}_{\boldsymbol{p}} = \{\boldsymbol{x} \in \mathbb{R}^{NT} \mid \sum_{n\in\mathcal{N}} \boldsymbol{x}_n = \boldsymbol{p}\}$$

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

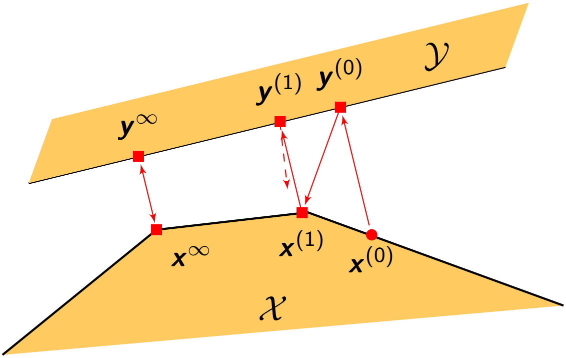
**repeat**

$\boldsymbol{x}^{(k+1)} \leftarrow P_{\mathcal{X}}(\boldsymbol{y}^{(k)})$

$\boldsymbol{y}^{(k+1)} \leftarrow P_{\mathcal{Y}}(\boldsymbol{x}^{(k+1)})$

$k \leftarrow k + 1$

**until** $\left\|\boldsymbol{y}^{(k)} - \boldsymbol{y}^{(k-1)}\right\| < \varepsilon_{\text{cvg}}$



## Theorem (Gubin, Polyak, 1967)

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

$$\boldsymbol{x}^{(k)} \underset{k\to\infty}{\longrightarrow} \boldsymbol{x}^\infty \ , \quad \boldsymbol{y}^{(k)} \underset{k\to\infty}{\longrightarrow} \boldsymbol{y}^\infty; \tag{3a}$$

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

*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 $\boldsymbol{x}^\infty \in \mathcal{X} \cap \mathcal{Y}$.*

## 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 $(\boldsymbol{x}^\infty, \boldsymbol{y}^\infty)$ defined in Theorem 2:*

(4a)

(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 $(\boldsymbol{x}^\infty, \boldsymbol{y}^\infty)$ defined in Theorem 2:*

$$\mathcal{T}_0 \stackrel{\text{def}}{=} \{t | p_t > \sum_{n\in\mathcal{N}} x^\infty_{n,t}\} \quad (4a)$$

$$(4b)$$

## 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 $(\boldsymbol{x}^\infty, \boldsymbol{y}^\infty)$ defined in Theorem 2:*

$$\mathcal{T}_0 \stackrel{\text{def}}{=} \{t | p_t > \textstyle\sum_{n\in\mathcal{N}} x^\infty_{n,t}\} \tag{4a}$$

$$\mathcal{N}_0 \stackrel{\text{def}}{=} \{n \,| E_n - \textstyle\sum_{t\notin\mathcal{T}_0} \underline{x}_{n,t} - \sum_{t\in\mathcal{T}_0} \overline{x}_{n,t} < 0\} \tag{4b}$$

## 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 $(\boldsymbol{x}^\infty, \boldsymbol{y}^\infty)$ defined in Theorem 2:*

$$\mathcal{T}_0 \overset{\text{def}}{=} \{t | p_t > \textstyle\sum_{n\in\mathcal{N}} x^\infty_{n,t}\} \tag{4a}$$

$$\mathcal{N}_0 \overset{\text{def}}{=} \{n \,| E_n - \textstyle\sum_{t\notin\mathcal{T}_0} \underline{x}_{n,t} - \sum_{t\in\mathcal{T}_0} \overline{x}_{n,t} < 0\} \tag{4b}$$

*define a "Hoffman cut" violated by $\boldsymbol{p}$, that is:*

$$\sum_{n\in\mathcal{N}_0} E_n + \sum_{t\in\mathcal{T}_0, n\notin\mathcal{N}_0} \overline{x}_{n,t} - \sum_{t\notin\mathcal{T}_0, n\in\mathcal{N}_0} \underline{x}_{n,t} < \sum_{t\in\mathcal{T}_0} p_t \ . \tag{5}$$

## 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 $(\boldsymbol{x}^\infty, \boldsymbol{y}^\infty)$ defined in Theorem 2:*

$$\mathcal{T}_0 \stackrel{\text{def}}{=} \{t | p_t > \textstyle\sum_{n\in\mathcal{N}} x^\infty_{n,t}\} \tag{4a}$$

$$\mathcal{N}_0 \stackrel{\text{def}}{=} \{n \,| E_n - \textstyle\sum_{t\notin\mathcal{T}_0} \underline{x}_{n,t} - \sum_{t\in\mathcal{T}_0} \overline{x}_{n,t} < 0\} \tag{4b}$$

*define a "Hoffman cut" violated by $\boldsymbol{p}$, that is:*

$$\sum_{n\in\mathcal{N}_0} E_n + \sum_{t\in\mathcal{T}_0, n\notin\mathcal{N}_0} \overline{x}_{n,t} - \sum_{t\notin\mathcal{T}_0, n\in\mathcal{N}_0} \underline{x}_{n,t} < \sum_{t\in\mathcal{T}_0} p_t \ . \tag{5}$$

*This cut can be reformulated in terms of $\mathbb{1}_N^\top \boldsymbol{x}^\infty$ as:*

$$\mathrm{A}_{\mathcal{T}_0} < \sum_{t\in\mathcal{T}_0} p_t \ \textit{with}\ \mathrm{A}_{\mathcal{T}_0} \stackrel{\text{def}}{=} \sum_{t\in\mathcal{T}_0} \sum_{n\in\mathcal{N}} x^\infty_{n,t}. \tag{6}$$

# Linear convergence of APM in our case

**Theorem**

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

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

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

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

# Some Ingredients...

## Lemma (Nishihara et al, 2014)

*For APM on polyhedra $\mathcal{X}$ and $\mathcal{Y}$, the sequences $(\boldsymbol{x}^{(k)})_k$ and $(\boldsymbol{y}^{(k)})_k$ 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:*

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

# Some Ingredients...

## Lemma (Nishihara et al, 2014)

*For APM on polyhedra $\mathcal{X}$ and $\mathcal{Y}$, the sequences $(\mathbf{x}^{(k)})_k$ and $(\mathbf{y}^{(k)})_k$ 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 $\Lambda_{\mathrm{sv}}(AB^{\top})$ the set of singular values of $AB^{\top}$. Then:*

- *if $\Lambda_{\mathrm{sv}}(AB^{\top}) = \{1\}$, then $c_F(\mathrm{Ker(A)}, \mathrm{Ker(B)}) = 0$ ;*
- *Otherwise, $c_F(\mathrm{Ker(A)}, \mathrm{Ker(B)}) = \max_{\lambda<1}\{\lambda \in \Lambda_{\mathrm{sv}}(\mathrm{AB}^{\top})\}$.*

# Convergence rate: sketch of proof - 2:

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

# Convergence rate: sketch of proof - 2:

- $\mathcal{Y}$ is affine subspace $\mathcal{Y} = \{\boldsymbol{x} \in \mathbb{R}^{NT} \; | A\boldsymbol{x} = \sqrt{N}^{-1}\mathbf{1}_T\}$ with $\overrightarrow{\mathcal{Y}} = \text{Ker}(A)$ and $A \stackrel{\text{def}}{=} \sqrt{N}^{-1} J_{1,N} \otimes I_T$.
- Faces of $\mathcal{X}$ are subsets of the collection of affine subspaces indexed by $(\overline{\mathcal{T}}_n, \underline{\mathcal{T}}_n)_n \subset \mathcal{T}^N$ (with $\overline{\mathcal{T}} \cap \underline{\mathcal{T}} = \emptyset$):

$$\mathcal{A}_{(\overline{\mathcal{T}}_n, \underline{\mathcal{T}}_n)_n} \stackrel{\text{def}}{=} \left\{ (\boldsymbol{x})_{nt} \; | \forall n, \;\; \boldsymbol{x}_n^\top \mathbb{1}_T = E_n \text{ and } \forall t \in \overline{\mathcal{T}}_n, x_{n,t} = \underline{x}_{n,t}, \; \text{ and } \forall t \in \underline{\mathcal{T}}_n, x_{n,t} = \overline{x}_{n,t} \right\}.$$

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

# Convergence rate: sketch of proof - 2:

- $\mathcal{Y}$ is affine subspace $\mathcal{Y} = \{\boldsymbol{x} \in \mathbb{R}^{NT} \mid A\boldsymbol{x} = \sqrt{N}^{-1}\mathbf{1}_T\}$ with $\overrightarrow{\mathcal{Y}} = \mathrm{Ker}(A)$ and $A \stackrel{\text{def}}{=} \sqrt{N}^{-1} J_{1,N} \otimes I_T$.
- Faces of $\mathcal{X}$ are subsets of the collection of affine subspaces indexed by $(\overline{\mathcal{T}}_n, \underline{\mathcal{T}}_n)_n \subset \mathcal{T}^N$ (with $\overline{\mathcal{T}} \cap \underline{\mathcal{T}} = \emptyset$):

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

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

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

$$S := (AB^\top)(A^\top B) = \frac{1}{N} \left( \sum_n \frac{\mathbb{1}_{\{k,\ell\} \subset \mathcal{T}_n^c}}{T - K_n} \right)_{k,\ell} + \frac{1}{N} \sum_{1 \leq t \leq T} \left( \textstyle\sum_n \mathbb{1}_{t \in \mathcal{T}_n} \right) E_{t,t}.$$

# Convergence rate: sketch of proof - 2:

- $\mathcal{Y}$ is affine subspace $\mathcal{Y} = \{\boldsymbol{x} \in \mathbb{R}^{NT} \,|A\boldsymbol{x} = \sqrt{N}^{-1}\mathbf{1}_T\}$ with $\overrightarrow{\mathcal{Y}} = \mathrm{Ker}(A)$ and $A \stackrel{\text{def}}{=} \sqrt{N}^{-1} J_{1,N} \otimes I_T$.
- Faces of $\mathcal{X}$ are subsets of the collection of affine subspaces indexed by $(\overline{\mathcal{T}}_n, \underline{\mathcal{T}}_n)_n \subset \mathcal{T}^N$ (with $\overline{\mathcal{T}} \cap \underline{\mathcal{T}} = \emptyset$):

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

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

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

$$S := (AB^\top)(A^\top B) = \frac{1}{N}\left(\sum_n \frac{\mathbb{1}_{\{k,\ell\}\subset\mathcal{T}_n^c}}{T-K_n}\right)_{k,\ell} + \frac{1}{N}\sum_{1\leq t\leq T} \left(\textstyle\sum_n \mathbb{1}_{t\in\mathcal{T}_n}\right) E_{t,t}.$$

- Denote $\bar{\mathcal{T}} \stackrel{\text{def}}{=} \cup_n \mathcal{T}_n^c$ and $P \stackrel{\text{def}}{=} I_T - S$. Then $P = \mathrm{diag}(P_{\bar{\mathcal{T}}}, 0_{\bar{\mathcal{T}}^c})$
$\rightarrow$ restrict to $\mathrm{Vect}(e_t)_{t\in\bar{\mathcal{T}}}$ to find $\lambda_1(P)$ (least positive eigval)

# Convergence rate: sketch of proof - 3:

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

# Convergence rate: sketch of proof - 3:

- Consider graph $\mathcal{G} = (\bar{\mathcal{T}}, \mathcal{E})$ whose vertices set is $\bar{\mathcal{T}}$ and edge $(k, \ell)$ has weight $S_{k,\ell} = \frac{1}{N} \sum_n \frac{\mathbb{1}_{\{k,\ell\} \subset \mathcal{T}_n^c}}{T - K_n}$. 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, $\forall u \perp \mathbf{1}$:

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

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

# Convergence rate: sketch of proof - 3:

- Consider graph $\mathcal{G} = (\bar{\mathcal{T}}, \mathcal{E})$ whose vertices set is $\bar{\mathcal{T}}$ and edge $(k,\ell)$ has weight $S_{k,\ell} = \frac{1}{N}\sum_n \frac{\mathbb{1}_{\{k,\ell\}\subset \mathcal{T}_n^c}}{T-K_n}$. 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, $\forall u \perp \mathbf{1}$:

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

  where $u_{t^*} := \max_t u_t$, $u_{s^*} := \min_t u_t$ and $d_{s^*,t^*}$ distance in $\mathcal{G}$, and $(s^*\text{-}t^*)$ a path from $s^*$ to $t^*$.
- As $\mathbf{1}$ 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**

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

**Disaggregation Problem**

$$\text{Find } \boldsymbol{x} \in \mathcal{Y}_{\boldsymbol{p}} \cap \left(\textstyle\prod_n \mathcal{X}_n\right)$$

# back to the two subproblems...

Master Problem

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

$\boldsymbol{p}^{(s)}$ →

Disaggregation Problem

$$\text{Find } \boldsymbol{x} \in \mathcal{Y}_{\boldsymbol{p}} \cap \left(\prod_n \mathcal{X}_n\right)$$

# back to the two subproblems...

**Master Problem**

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

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

$\boldsymbol{p}^{(s)}$ →

← $\mathcal{P}^{(s+1)}$

**Disaggregation Problem**

$$\text{Find } \boldsymbol{x} \in \mathcal{Y}_{\boldsymbol{p}} \cap \left(\prod_n \mathcal{X}_n\right)$$

# back to the two subproblems...

**Master Problem**

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

$\boldsymbol{p}^{(s)}$ →

← $\mathcal{P}^{(s+1)}$

**Disaggregation Problem**

$$\text{Find } \boldsymbol{x} \in \mathcal{Y}_{\boldsymbol{p}} \cap \left(\textstyle\prod_n \mathcal{X}_n\right)$$

$$\mathcal{P}^{(s+1)} = \mathcal{P}^{(s)} \cap \{\boldsymbol{p} \mid \ \mathrm{A}_{\mathcal{T}_0} < \textstyle\sum_{t\in\mathcal{T}_0} p_t\}$$

# To non-intrusive projections...

- projections $\boldsymbol{x}_n = P_{\mathcal{X}_n}(\boldsymbol{y}_n)$ can be computed locally ;

# To non-intrusive projections...

- projections $\boldsymbol{x}_n = P_{\mathcal{X}_n}(\boldsymbol{y}_n)$ can be computed locally ;
- projection $\boldsymbol{y} = P_{\mathcal{Y}_{\boldsymbol{p}}}(\boldsymbol{x})$ on $\mathcal{Y}_{\boldsymbol{p}} = \{\boldsymbol{x} \in \mathbb{R}^{NT} \mid \sum_n \boldsymbol{x}_n = \boldsymbol{p}\}$ is explicit (affine space):

$$\forall n,\ \boldsymbol{y}_n = \boldsymbol{x}_n + \frac{1}{N}\Big(\boldsymbol{p} - (\textstyle\sum_m \boldsymbol{x}_m)\Big) \ ;$$

# To non-intrusive projections...

- projections $\boldsymbol{x}_n = P_{\mathcal{X}_n}(\boldsymbol{y}_n)$ can be computed locally ;
- projection $\boldsymbol{y} = P_{\mathcal{Y}_{\boldsymbol{p}}}(\boldsymbol{x})$ on $\mathcal{Y}_{\boldsymbol{p}} = \{\boldsymbol{x} \in \mathbb{R}^{NT} \mid \sum_n \boldsymbol{x}_n = \boldsymbol{p}\}$ is explicit (affine space):

$$\forall n, \ \boldsymbol{y}_n = \boldsymbol{x}_n + \frac{1}{N}\Big(\boldsymbol{p} - (\textstyle\sum_m \boldsymbol{x}_m)\Big) \ ;$$

- only requires the aggregate quantity $\sum_n \boldsymbol{x}_n$ , then local operations;

# To non-intrusive projections...

- projections $\boldsymbol{x}_n = P_{\mathcal{X}_n}(\boldsymbol{y}_n)$ can be computed locally ;
- projection $\boldsymbol{y} = P_{\mathcal{Y}_{\boldsymbol{p}}}(\boldsymbol{x})$ on $\mathcal{Y}_{\boldsymbol{p}} = \{\boldsymbol{x} \in \mathbb{R}^{NT} \mid \sum_n \boldsymbol{x}_n = \boldsymbol{p}\}$ is explicit (affine space):

$$\forall n, \; \boldsymbol{y}_n = \boldsymbol{x}_n + \frac{1}{N}\Big(\boldsymbol{p} - (\textstyle\sum_m \boldsymbol{x}_m)\Big) \;;$$

- only requires the aggregate quantity $\sum_n \boldsymbol{x}_n$ , 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 $(\boldsymbol{x}_n^{(k)})_n$
$\rightarrow$ **Secure Multiparty Computation** (SMC) principle

**Require:** Each agent has a profile $(\boldsymbol{x}_n)_{n \in \mathcal{N}}$
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 $\forall t, s_{n,t,N} \stackrel{\text{def}}{=} x_{n,t} - \sum_{m=1}^{N-1} s_{n,t,m}$
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 $\boldsymbol{S} = \sum_{n \in \mathcal{N}} \boldsymbol{\sigma}_n$

# Issues: transmission of profiles for projection

In APM, agents still have to provide profiles $(\boldsymbol{x}_n^{(k)})_n$
$\rightarrow$ **Secure Multiparty Computation** (SMC) principle

**Require:** Each agent has a profile $(\boldsymbol{x}_n)_{n\in\mathcal{N}}$
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 $\forall t, s_{n,t,N} \stackrel{\text{def}}{=} x_{n,t} - \sum_{m=1}^{N-1} s_{n,t,m}$
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 $\boldsymbol{S} = \sum_{n\in\mathcal{N}} \boldsymbol{\sigma}_n$

$x_1 = s_{1,1} + s_{1,2} + s_{1,3}$

$x_2 = s_{2,1} + s_{2,2} + s_{2,3}$

$x_3 = s_{3,1} + s_{3,2} + s_{3,3}$

# Issues: transmission of profiles for projection

In APM, agents still have to provide profiles $(\boldsymbol{x}_n^{(k)})_n$
$\rightarrow$ **Secure Multiparty Computation** (SMC) principle

**Require:** Each agent has a profile $(\boldsymbol{x}_n)_{n\in\mathcal{N}}$
**for** each agent $n \in \mathcal{N}$ **do**
    Draw $\forall t, (s_{n,t,m})_{m=1}^{N-1} \in \mathcal{U}([0,A]^{N-1})$
    and set $\forall t, s_{n,t,N} \stackrel{\text{def}}{=} x_{n,t} - \sum_{m=1}^{N-1} s_{n,t,m}$
    Send $(s_{n,t,m})_{t\in\mathcal{T}}$ to agent $m \in \mathcal{N}$
**done**
**for** each agent $n \in \mathcal{N}$ **do**
    Compute $\forall t, \sigma_{n,t} = \sum_{m\in\mathcal{N}} s_{m,t,n}$
    Send $(\sigma_{n,t})_{t\in\mathcal{T}}$ to operator
**done**
Operator computes $\boldsymbol{S} = \sum_{n\in\mathcal{N}} \boldsymbol{\sigma}_n$

$$x_1 = s_{1,1} + s_{1,2} + s_{1,3}$$
$$x_2 = s_{2,1} + s_{2,2} + s_{2,3}$$
$$x_3 = s_{3,1} + s_{3,2} + s_{3,3}$$
$$\sum_n x_n = \sigma_1 + \sigma_2 + \sigma_3$$

# Combining all elements...

**Require:** $s = 0$ , $\mathcal{P}^{(0)} = \mathcal{P}$ ; $\textsc{Disag} = \textsc{False}$

1: **while** Not $\textsc{Disag}$ **do**

# Combining all elements...

**Require:** $s = 0$ , $\mathcal{P}^{(0)} = \mathcal{P}$ ; $\textsc{Disag} = \textsc{False}$

1: **while** Not $\textsc{Disag}$ **do**

2: Compute $\boldsymbol{p}^{(s)} = \arg\min_{\boldsymbol{p} \in \mathcal{P}_{\text{cs}}^{(s)}} \boldsymbol{p}$

**Master Problem**

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

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

# Combining all elements...

**Require:** $s = 0$ , $\mathcal{P}^{(0)} = \mathcal{P}$ ; $\textsc{Disag} = \textsc{False}$

1: **while** Not $\textsc{Disag}$ **do**

2: Compute $\boldsymbol{p}^{(s)} = \arg\min_{\boldsymbol{p} \in \mathcal{P}_{\text{cs}}^{(s)}} \boldsymbol{p}$

3: $\textsc{Disag} \leftarrow \text{APM}(\boldsymbol{p}^{(s)})$

**Master Problem**

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

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

$\downarrow$ $\boldsymbol{p}^{(s)}$

**Disaggregation Pb**

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

# Combining all elements...

**Require:** $s = 0$ , $\mathcal{P}^{(0)} = \mathcal{P}$ ; $\textsc{Disag} = \textsc{False}$

**while** Not $\textsc{Disag}$ **do**

  Compute $\boldsymbol{p}^{(s)} = \arg\min_{\boldsymbol{p} \in \mathcal{P}_{\text{cs}}^{(s)}} \boldsymbol{p}$

  $\textsc{Disag} \leftarrow \text{APM}(\boldsymbol{p}^{(s)})$

  **if** $\textsc{Disag}$ **then**

    Operator adopts $\boldsymbol{p}^{(s)}$

**Master Problem**

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

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

$\boldsymbol{p}^{(s)}$ ↓

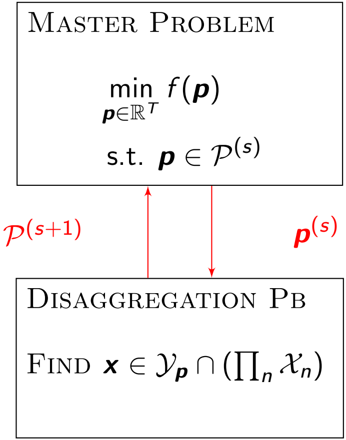
**Disaggregation Pb**

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

# Combining all elements...

**Require:** $s = 0$ , $\mathcal{P}^{(0)} = \mathcal{P}$ ; $\textsc{Disag} = \textsc{False}$

**while** Not $\textsc{Disag}$ **do**
    Compute $\boldsymbol{p}^{(s)} = \arg\min_{\boldsymbol{p} \in \mathcal{P}^{(s)}_{\text{cs}}} \boldsymbol{p}$
    $\textsc{Disag} \leftarrow \text{APM}(\boldsymbol{p}^{(s)})$
    **if** $\textsc{Disag}$ **then**
        Operator adopts $\boldsymbol{p}^{(s)}$
    **else**
        Obtain $\mathcal{T}_0^{(s)}, \mathrm{A}^{(s)}_{\mathcal{T}_0}$ from $\text{APM}(\boldsymbol{p}^{(s)})$
        $\mathcal{P}^{(s+1)} \leftarrow \mathcal{P}^{(s)} \cap \{\boldsymbol{p} | \sum_{t \in \mathcal{T}_0^{(s)}} p_t \leq \mathrm{A}^{(s)}_{\mathcal{T}_0}\}$
**end**



# Combining all elements...

**Require:** $s = 0$ , $\mathcal{P}^{(0)} = \mathcal{P}$ ; $\textsc{Disag} = \textsc{False}$

**while** Not $\textsc{Disag}$ **do**

  Compute $\boldsymbol{p}^{(s)} = \arg\min_{\boldsymbol{p} \in \mathcal{P}^{(s)}_{\text{cs}}} \boldsymbol{p}$

  $\textsc{Disag} \leftarrow \text{APM}(\boldsymbol{p}^{(s)})$

  **if** $\textsc{Disag}$ **then**

    Operator adopts $\boldsymbol{p}^{(s)}$

  **else**

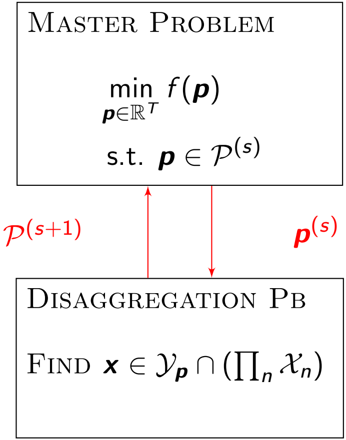
    Obtain $\mathcal{T}_0^{(s)}, \mathrm{A}_{\mathcal{T}_0}^{(s)}$ from $\text{APM}(\boldsymbol{p}^{(s)})$

    $\mathcal{P}^{(s+1)} \leftarrow \mathcal{P}^{(s)} \cap \{\boldsymbol{p} | \sum_{t \in \mathcal{T}_0^{(s)}} p_t \leq \mathrm{A}_{\mathcal{T}_0}^{(s)}\}$

  **end**

  $s \leftarrow s + 1$

**done**



# 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.

# 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 $\boldsymbol{x}^\infty$ of the APM sequence to obtain the cut..

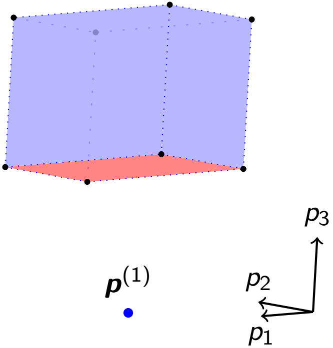
# Termination condition: number of cuts

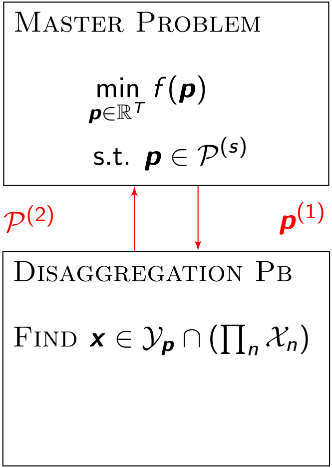
**Proposition**

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

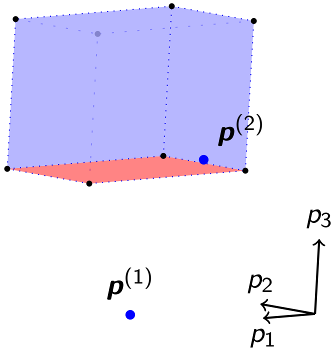
ISSUE: we need the limit $\boldsymbol{x}^\infty$ of the APM sequence to obtain the cut..
but in practice we can stop in finite time and obtain the ~~approximated~~ same cut!

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



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




MASTER PROBLEM

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

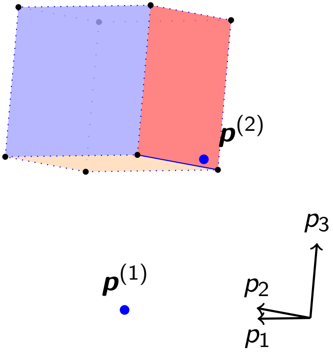
$$\text{s.t. } \boldsymbol{p} \in \mathcal{P}^{(s)}$$

$\boldsymbol{p}^{(2)}$

DISAGGREGATION PB

FIND $\boldsymbol{x} \in \mathcal{Y}_{\boldsymbol{p}} \cap (\prod_n \mathcal{X}_n)$

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




**Master Problem**

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

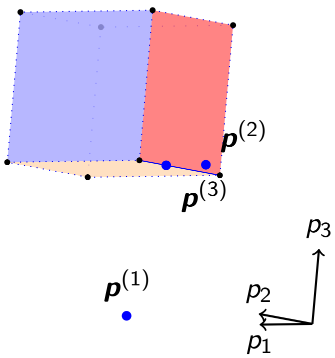
$$\text{s.t. } \boldsymbol{p} \in \mathcal{P}^{(s)}$$

$\mathcal{P}^{(3)}$

**Disaggregation Pb**

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

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




**Master Problem**

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

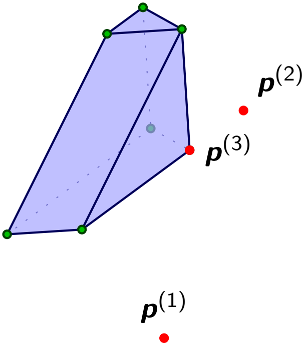
$\boldsymbol{p}^{(3)}$

**Disaggregation Pb**

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

Feasible!

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




**MASTER PROBLEM**

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

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

$\boldsymbol{p}^{(3)}$

**DISAGGREGATION PB**

FIND $\boldsymbol{x} \in \mathcal{Y}_{\boldsymbol{p}} \cap (\prod_n \mathcal{X}_n)$

FEASIBLE!

# Conclusion

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

# Conclusion

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

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

# Conclusion

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

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

# Conclusion

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

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

**Further work and extensions:**

# Conclusion

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

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

**Further work and extensions:**

- a variant of the algorithm can deal individual constaints $\mathcal{X}_n \rightarrow$ arbitrary (polyhedral) set, resolving local LPs to get a cut;

# Conclusion

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

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

**Further work and extensions:**

- a variant of the algorithm can deal individual constaints $\mathcal{X}_n \to$ 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 $\boldsymbol{p}$ and $N$ individual agents profiles $(\boldsymbol{x}_n)_n$, such that $(\boldsymbol{x}, \boldsymbol{p})$ solves the global (nonconvex) problem, while keeping private:

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

**Further work and extensions:**

- a variant of the algorithm can deal individual constaints $\mathcal{X}_n \rightarrow$ 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 !


Jacquot, Paulin, Olivier Beaude, Pascal Benchimol, Stéphane Gaubert, and Nadia Oudjane (2019a). "A Privacy-preserving Disaggregation Algorithm for Non-intrusive Management of Flexible Energy". In: IEEE 58th Conference on Decision and Control (CDC). IEEE. arXiv: 1903.03053.

– (2019b). "A Privacy-preserving Method to optimize distributed resource allocation". In: arXiv preprint. arXiv: 1908.03080.