A step toward decarbonization: the role of constraint coupled optimization in energy systems

Ruggero Carli

Department of Information Engineering, University of Padova

ECODREAM Workshop
Naples, 28 November 2025

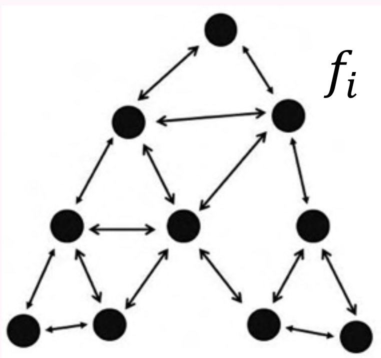
Outline

- Distributed coupled-constraint optimization
- Dynamic Average Consensus based on ADMM (Alternating Direction Method Multipliers)
- Application to energy systems (Energy communities)
- A step toward decarbonization…

Outline

- Distributed coupled-constraint optimization
- Dynamic Average Consensus based on ADMM (Alternating Direction Method Multipliers)
- Application to energy systems (Energy communities)
- A step toward decarbonization…

Distributed Constraint-Coupled Optimization



Undirected Graph $\mathcal{G} = (V, \mathcal{E})$

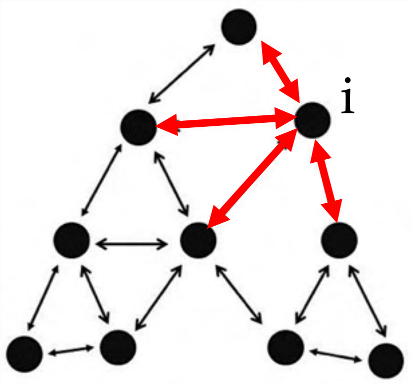
- V : set of nodes $V = \{1, \ldots, N\}$
- $\mathcal{E}$: set of edges
  (i, j) : i and j can communicate with each other
- f_i : local function known by node i

$$\min_{x_1,\ldots,x_N} \sum_{i=1}^{N} f_i(x_i)$$

$$\text{subject to:} \quad \sum_{i=1}^{N} A_i x_i = b$$

$$x_i \in X_i \qquad i = 1, \ldots, N$$

Distributed Constraint-Coupled Optimization

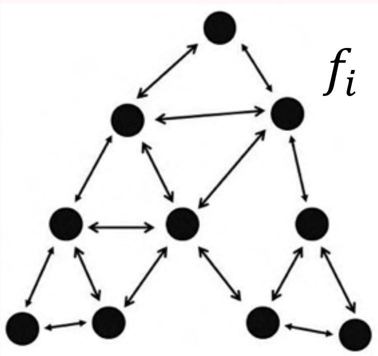


Undirected Graph $\mathcal{G} = (V, \mathcal{E})$

- V : set of nodes $V = \{1, \dots, N\}$
- $\mathcal{E}$: set of edges
  (i, j) : i and j can communicate with each other
- f_i : local function known by node i
- $\mathcal{N}_i$: set of neighbors of i

$$\mathcal{N}_i = \{j \in V \mid (i, j) \in \mathcal{E}\}$$

Distributed Constraint-Coupled Optimization



$$\min_{x_1, \dots, x_N} \sum_{i=1}^{N} f_i(x_i)$$

$$\text{subject to: } \sum_{i=1}^{N} A_i x_i = b$$

$$x_i \in X_i \qquad i = 1, \dots, N$$

Goal : designing *distributed* and *scalable* algorithms

(that is, only exchange of information between neighbors)

Optimization in Engineering : ADMM

$$\underset{x,y}{\text{minimize}} \quad f(x) + g(y)$$
$$\text{subject that} \quad Ax + By = c$$

Augmented Lagrangian (λ is vector of Lagrange multipliers, $\rho > 0$)

$$\mathcal{L}_\rho(x, y, \lambda) = f(x) + g(y) - \lambda^T (Ax + By - c) + \frac{\rho}{2}\|Ax + By - c\|^2$$

Optimization in Engineering : ADMM

$$\underset{x,y}{\text{minimize}} \quad f(x) + g(y)$$
$$\text{subject that} \quad Ax + By = c$$

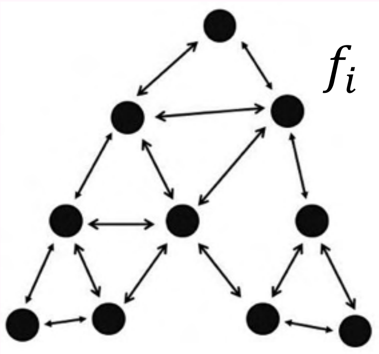
Augmented Lagrangian (λ is vector of Lagrange multipliers, $\rho > 0$)

$$\mathcal{L}_\rho(x, y, \lambda) = f(x) + g(y) - \lambda^T (Ax + By - c) + \frac{\rho}{2}\|Ax + By - c\|^2$$

ADMM – Alternating direction multipliers method

1. $$x(k+1) = \underset{x}{\text{argmin}}\, \mathcal{L}_\rho(x, y(k), \lambda(k))$$
2. $$y(k+1) = \underset{y}{\text{argmin}}\, \mathcal{L}_\rho(x(k+1), y, \lambda(k))$$
3. $$\lambda(k+1) = \lambda(k) + \rho\,(Ax(k+1) + By(k+1) - c)$$

Distributed Constraint-Coupled Optimization



$$\begin{aligned} \min_{x_1,\dots,x_N} \quad & \sum_{i=1}^{N} f_i(x_i) \\ \text{subject to:} \quad & \sum_{i=1}^{N} A_i x_i = b \\ & x_i \in X_i \qquad i = 1, \dots, N \end{aligned}$$

- Introduction of Augmented Lagrangian
- λ Lagrange multiplier vector
- Implementation of ADMM algorithm

Parallel Constraint-Coupled Optimization

Parallel ADMM (Bertsekas, Tsitsiklis, 1989)

$$x_{i,k+1} \in \operatorname*{argmin}_{x_i \in X_i} \left\{ f_i(x_i) + \lambda_k^\top A_i x_i + \frac{c}{2} \|A_i x_i - A_i x_{i,k} + d_k\|^2 \right\}$$

$$d_{k+1} = \frac{1}{N} \sum_{i=1}^{N} (A_i x_{i,k+1} - b_i)$$

$$\lambda_{k+1} = \lambda_k + c\, d_{k+1}$$

$\sum_{i=1}^{N} b_i = b$

Parallel Constraint-Coupled Optimization

Parallel ADMM (Bertsekas, Tsitsiklis, 1989)

$$x_{i,k+1} \in \underset{x_i \in X_i}{\operatorname{argmin}} \left\{ f_i(x_i) + \lambda_k^\top A_i x_i + \frac{c}{2} \| A_i x_i - A_i x_{i,k} + d_k \|^2 \right\}$$

$$d_{k+1} = \frac{1}{N} \sum_{i=1}^{N} (A_i x_{i,k+1} - b_i)$$

$$\lambda_{k+1} = \lambda_k + c\, d_{k+1}$$

it measures the feasibility of x_i *(or violation of constraint…)*

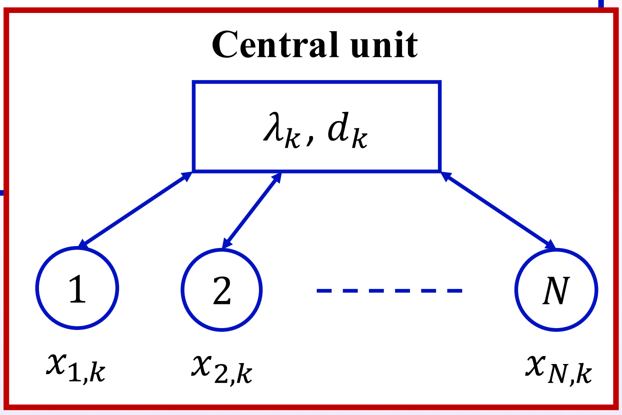
Parallel Constraint-Coupled Optimization

Parallel ADMM (Bertsekas, Tsitsiklis, 1989)

$$x_{i,k+1} \in \underset{x_i \in X_i}{\operatorname{argmin}} \left\{ f_i(x_i) + \lambda_k^\top A_i x_i + \frac{c}{2} \| A_i x_i - A_i x_{i,k} + d_k \|^2 \right\}$$

$$d_{k+1} = \frac{1}{N} \sum_{i=1}^{N} (A_i x_{i,k+1} - b_i)$$

$$\lambda_{k+1} = \lambda_k + c\, d_{k+1}$$



Distributed Constraint-Coupled Optimization

Local copies $\lambda_k \longrightarrow \lambda_{i,k}$ $\quad d_k \longrightarrow d_{i,k}$

are updated iteratively according to a **consensus-based schemes** to force **agreement** of the local copies

$$x_{i,k+1} = \sum_{j \in \mathcal{N}_i} w_{ij}\ x_{j,k}$$

W = $[w_{ij}]$ consensus matrix

A. Falsone, I. Notarnicola, G.Notarstefano, M. Prandini. *Tracking ADMM for distributed constraint coupled optimization.* In Automatica 2020

Distributed Constraint-Coupled Optimization

Local copies $\lambda_k \longrightarrow \lambda_{i,k}$ $\quad d_k \longrightarrow d_{i,k}$

$$x_{i,k+1} \in \underset{x_i \in X_i}{\text{argmin}} \left\{ f_i(x_i) + \lambda_k^\top A_i x_i + \frac{c}{2}\|A_i x_i - A_i x_{i,k} + d_k\|^2 \right\}$$

$$d_{k+1} = \frac{1}{N}\sum_{i=1}^{N}(A_i x_{i,k+1} - b_i)$$

$$\lambda_{k+1} = \lambda_k + c\, d_{k+1}$$

W = $[w_{ij}]$ consensus matrix

Algorithm 1 Tracking-ADMM

1: **Initialization**
2: $x_{i,0} \in X_i$
3: $d_{i,0} = A_i x_{i,0} - b_i$
4: $\lambda_{i,0} \in \mathbb{R}^p$
5: **Repeat until convergence**
6: $\delta_{i,k} = \sum_{j\in\mathcal{N}_i} w_{ij}\, d_{j,k}$
7: $\ell_{i,k} = \sum_{j\in\mathcal{N}_i} w_{ij}\, \lambda_{j,k}$
8: $x_{i,k+1} \in \underset{x_i \in X_i}{\text{argmin}} \left\{ f_i(x_i) + \ell_{i,k}^\top A_i x_i + \frac{c}{2}\|A_i x_i - A_i x_{i,k} + \delta_{i,k}\|^2 \right\}$
9: $d_{i,k+1} = \delta_{i,k} + A_i x_{i,k+1} - A_i x_{i,k}$
10: $\lambda_{i,k+1} = \ell_{i,k} + c\, d_{i,k+1}$

Distributed Constraint-Coupled Optimization

Local copies $\lambda_k \longrightarrow \lambda_{i,k}$ $\quad d_k \longrightarrow d_{i,k}$

$$x_{i,k+1} \in \operatorname*{argmin}_{x_i \in X_i} \left\{ f_i(x_i) + \lambda_k^\top A_i x_i + \frac{c}{2} \| A_i x_i - A_i x_{i,k} + d_k \|^2 \right\}$$

$$d_{k+1} = \frac{1}{N} \sum_{i=1}^{N} (A_i x_{i,k+1} - b_i)$$

$$\lambda_{k+1} = \lambda_k + c\, d_{k+1}$$

$W = [w_{ij}]$ consensus matrix

Algorithm 1 Tracking-ADMM

Initialization

$x_{i,0} \in X_i$

$d_{i,0} = A_i x_{i,0} - b_i$

$\lambda_{i,0} \in \mathbb{R}^p$

Repeat until convergence

$\delta_{i,k} = \sum_{j \in \mathcal{N}_i} w_{ij}\, d_{j,k}$ ⟷ *consensus*

$\ell_{i,k} = \sum_{j \in \mathcal{N}_i} w_{ij}\, \lambda_{j,k}$

$$x_{i,k+1} \in \operatorname*{argmin}_{x_i \in X_i} \left\{ f_i(x_i) + \ell_{i,k}^\top A_i x_i + \frac{c}{2} \| A_i x_i - A_i x_{i,k} + \delta_{i,k} \|^2 \right\}$$

$d_{i,k+1} = \delta_{i,k} + A_i x_{i,k+1} - A_i x_{i,k}$

$\lambda_{i,k+1} = \ell_{i,k} + c\, d_{i,k+1}$

Distributed Constraint-Coupled Optimization

Local copies $\lambda_k \longrightarrow \lambda_{i,k}$ $\quad d_k \longrightarrow d_{i,k}$

$$x_{i,k+1} \in \operatorname*{argmin}_{x_i \in X_i} \left\{ f_i(x_i) + \lambda_k^\top A_i x_i + \frac{c}{2} \| A_i x_i - A_i x_{i,k} + d_k \|^2 \right\}$$

$$d_{k+1} = \frac{1}{N} \sum_{i=1}^{N} (A_i x_{i,k+1} - b_i)$$

$$\lambda_{k+1} = \lambda_k + c\, d_{k+1}$$

$W = [w_{ij}]$ consensus matrix

Algorithm 1 Tracking-ADMM

Initialization

$x_{i,0} \in X_i$

$d_{i,0} = A_i x_{i,0} - b_i$

$\lambda_{i,0} \in \mathbb{R}^p$

Repeat until convergence

$\delta_{i,k} = \sum_{j \in \mathcal{N}_i} w_{ij}\, d_{j,k}$

$\ell_{i,k} = \sum_{j \in \mathcal{N}_i} w_{ij}\, \lambda_{j,k}$ ⟷ *consensus*

$$x_{i,k+1} \in \operatorname*{argmin}_{x_i \in X_i} \left\{ f_i(x_i) + \ell_{i,k}^\top A_i x_i + \frac{c}{2} \| A_i x_i - A_i x_{i,k} + \delta_{i,k} \|^2 \right\}$$

$d_{i,k+1} = \delta_{i,k} + A_i x_{i,k+1} - A_i x_{i,k}$

$\lambda_{i,k+1} = \ell_{i,k} + c\, d_{i,k+1}$

Distributed Constraint-Coupled Optimization

Local copies $\lambda_k \longrightarrow \lambda_{i,k}$ $\quad d_k \longrightarrow d_{i,k}$

$$x_{i,k+1} \in \operatorname*{argmin}_{x_i \in X_i} \left\{ f_i(x_i) + \lambda_k^\top A_i x_i + \frac{c}{2} \|A_i x_i - A_i x_{i,k} + d_k\|^2 \right\}$$

$$d_{k+1} = \frac{1}{N} \sum_{i=1}^{N} (A_i x_{i,k+1} - b_i)$$

$$\lambda_{k+1} = \lambda_k + c\, d_{k+1}$$

$W = [w_{ij}]$ consensus matrix

Algorithm 1 Tracking-ADMM

Initialization

$x_{i,0} \in X_i$

$d_{i,0} = A_i x_{i,0} - b_i$

$\lambda_{i,0} \in \mathbb{R}^p$

Repeat until convergence

$\delta_{i,k} = \sum_{j \in \mathcal{N}_i} w_{ij}\, d_{j,k}$

$\ell_{i,k} = \sum_{j \in \mathcal{N}_i} w_{ij}\, \lambda_{j,k}$

$$x_{i,k+1} \in \operatorname*{argmin}_{x_i \in X_i} \left\{ f_i(x_i) + \ell_{i,k}^\top A_i x_i + \frac{c}{2} \|A_i x_i - A_i x_{i,k} + \delta_{i,k}\|^2 \right\}$$

$d_{i,k+1} = \delta_{i,k} + A_i x_{i,k+1} - A_i x_{i,k}$

$\lambda_{i,k+1} = \ell_{i,k} + c\, d_{i,k+1}$

Distributed Constraint-Coupled Optimization

Local copies $\lambda_k \longrightarrow \lambda_{i,k}$ $\quad d_k \longrightarrow d_{i,k}$

$$x_{i,k+1} \in \operatorname*{argmin}_{x_i \in X_i} \left\{ f_i(x_i) + \lambda_k^\top A_i x_i + \frac{c}{2} \|A_i x_i - A_i x_{i,k} + d_k\|^2 \right\}$$

$$d_{k+1} = \frac{1}{N} \sum_{i=1}^{N} (A_i x_{i,k+1} - b_i)$$

$$\lambda_{k+1} = \lambda_k + c\, d_{k+1}$$

$W = [w_{ij}]$ consensus matrix

Algorithm 1 Tracking-ADMM

Initialization

$x_{i,0} \in X_i$

$d_{i,0} = A_i x_{i,0} - b_i$

$\lambda_{i,0} \in \mathbb{R}^p$

Repeat until convergence

$\delta_{i,k} = \sum_{j \in \mathcal{N}_i} w_{ij}\, d_{j,k}$

$\ell_{i,k} = \sum_{j \in \mathcal{N}_i} w_{ij}\, \lambda_{j,k}$

$$x_{i,k+1} \in \operatorname*{argmin}_{x_i \in X_i} \left\{ f_i(x_i) + \ell_{i,k}^\top A_i x_i + \frac{c}{2} \|A_i x_i - A_i x_{i,k} + \delta_{i,k}\|^2 \right\}$$

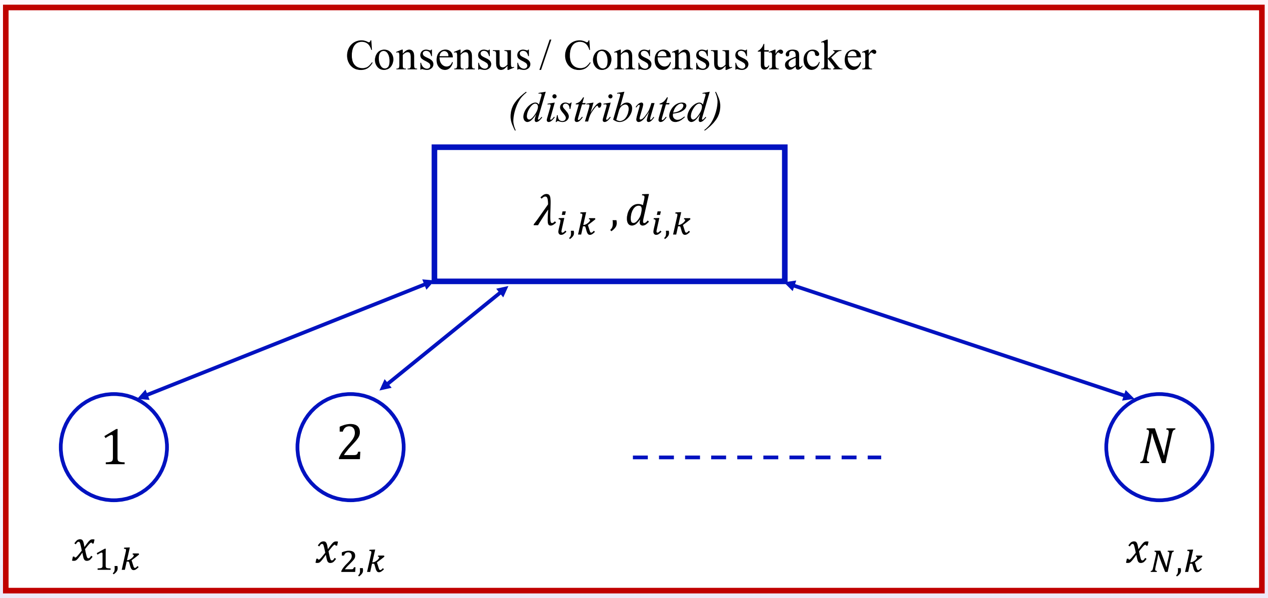
$d_{i,k+1} = \delta_{i,k} + A_i x_{i,k+1} - A_i x_{i,k}$

$\lambda_{i,k+1} = \ell_{i,k} + c\, d_{i,k+1}$

Distributed Constraint-Coupled Optimization

$$d_{i,k+1} = \sum_{j \in \mathcal{N}_i} w_{ij}\, d_{j,k} + (A_i x_{i,k+1} - b_i) - (A_i x_{i,k} - b_i)$$

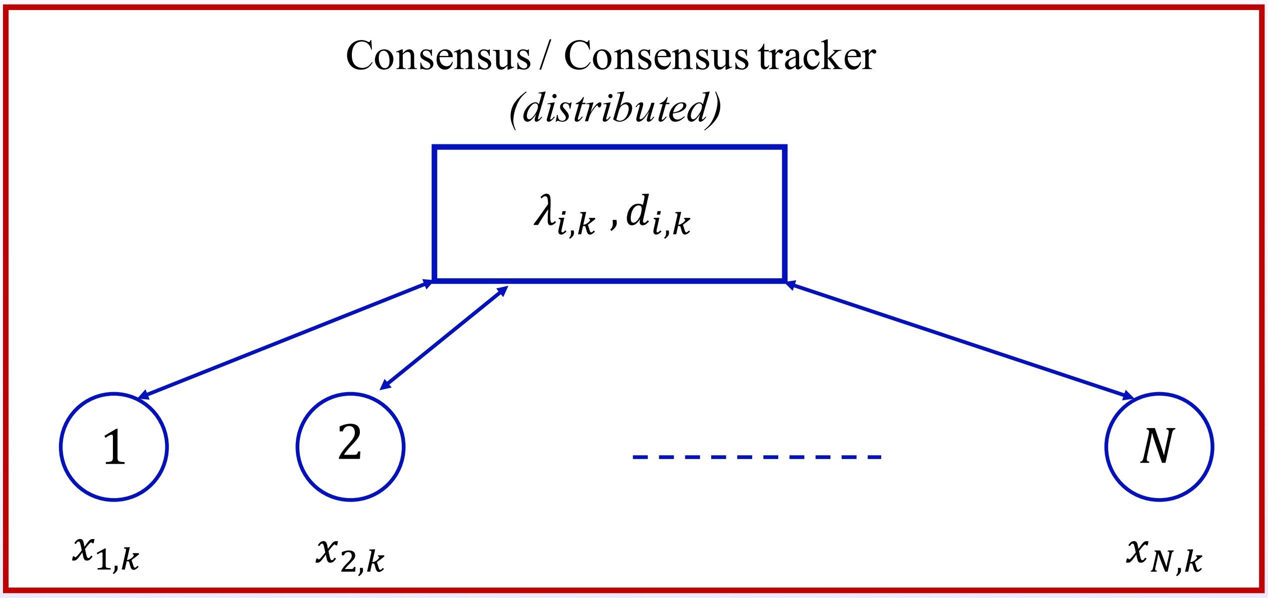
$d_{i,k}$ acts as a ***distributed tracker*** of the signal $(1/N)\sum_{i=1}^{N} (A_i x_{i,k} - b_i)$



Distributed Constraint-Coupled Optimization

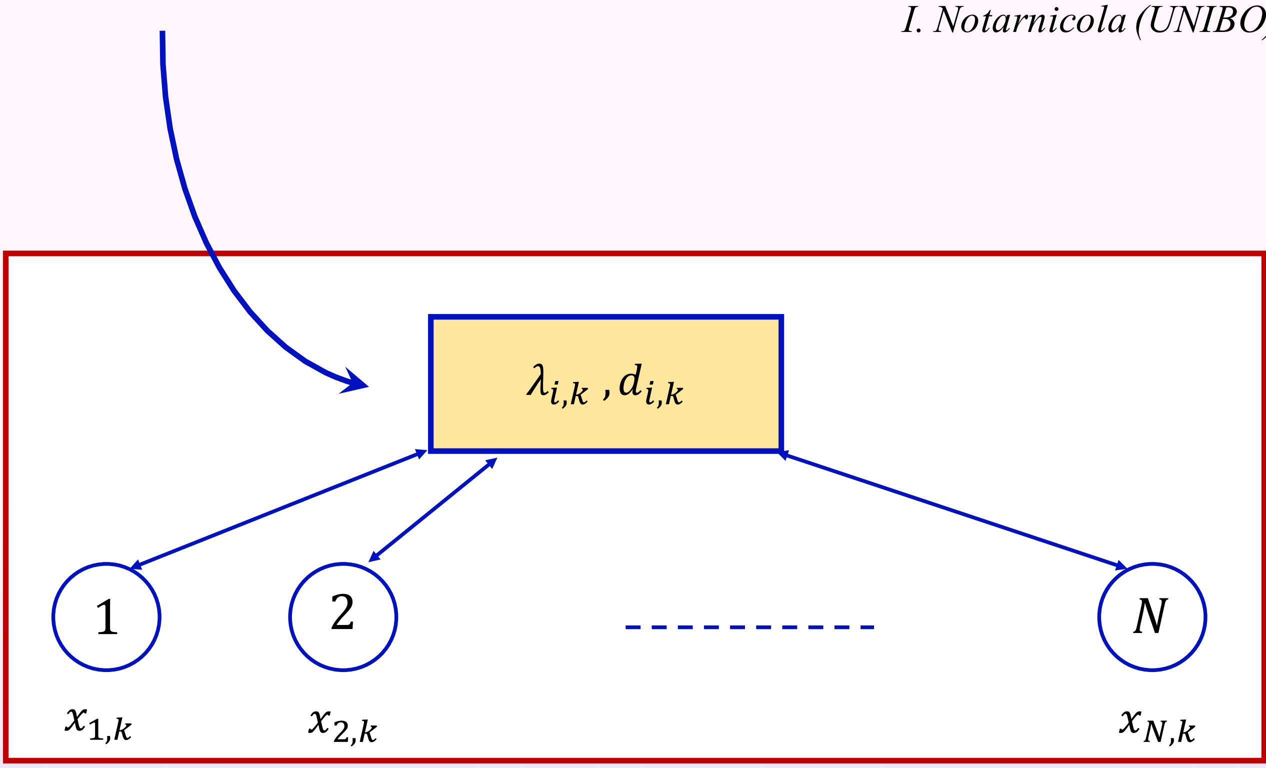
Drawbacks :

- Standard consensus versus *asynchrony/packet losses* ?
- Distributed tracker requires specific initiatization



Distributed Constraint-Coupled Optimization

Alternative idea : ADMM-consensus *(ongoing work with G. Carnevale, I. Notarnicola (UNIBO))*



Dynamic average consensus - ADMM

- Distributed coupled-constraint optimization
- Dynamic Average Consensus based on ADMM (Alternating Direction Method Multipliers)
- Application to energy systems (Energy communities)
- A step toward decarbonization…

N. Bastianello, R. Carli. *ADMM for dynamic average consensus over imperfect networks.* In Necsys 2022

Dynamic Consensus - ADMM

$\{v_{i,k} \in \mathbb{R}^n\}_{k \in \mathbb{N}},\ i = 1, \ldots, N$ *signals observed by each of the nodes*

$\{x_i\}_{i=1}^N$ *local states of the nodes*

Goal : to track time-varying average $\left\{ \bar{v}_k := \frac{1}{N} \sum_{i=1}^{N} v_{i,k} \right\}_{k \in \mathbb{N}}$

Dynamic Consensus - ADMM

$\{v_{i,k} \in \mathbb{R}^n\}_{k \in \mathbb{N}},\ i = 1, \ldots, N$ *signals observed by each of the nodes*

$\{x_i\}_{i=1}^N$ *local states of the nodes*

Goal : to track time-varying average $\left\{ \bar{v}_k := \frac{1}{N} \sum_{i=1}^{N} v_{i,k} \right\}_{k \in \mathbb{N}}$

Optimization Problem

$$\boldsymbol{x}_k^* = \underset{x_i \in \mathbb{R}^n, i=1,\ldots,N}{\arg\min} \sum_{i=1}^{N} \frac{1}{2} \|x_i - v_{i,k}\|^2$$

$$\text{s.t. } x_i = x_j \text{ if } (i,j) \in \mathcal{E}$$

→ $\boldsymbol{x}_k^* = \mathbf{1}_N \otimes \bar{v}_k$

Dynamic Consensus - ADMM

$\{v_{i,k} \in \mathbb{R}^n\}_{k\in\mathbb{N}},\ i = 1,\dots,N$ *signals observed by each of the nodes*

$A_i x_{i,k} - b_i$

$\{x_i\}_{i=1}^N$ *local states of the nodes*

Goal : to track time-varying average $\left\{ \bar{v}_k := \frac{1}{N}\sum_{i=1}^N v_{i,k} \right\}_{k\in\mathbb{N}}$

$(1/N)\sum_{i=1}^N (A_i x_{i,k} - b_i)$

Optimization Problem

$$\boldsymbol{x}_k^* = \underset{x_i\in\mathbb{R}^n, i=1,\dots,N}{\arg\min} \sum_{i=1}^N \frac{1}{2}\|x_i - v_{i,k}\|^2$$
$$\text{s.t. } x_i = x_j \text{ if } (i,j)\in\mathcal{E}$$

$\rightarrow$ $\boldsymbol{x}_k^* = \mathbf{1}_N \otimes \bar{v}_k$

Distributed optimization over networks

Optimization Problem

$$\boldsymbol{x}_k^* = \underset{x_i\in\mathbb{R}^n, i=1,\dots,N}{\arg\min} \sum_{i=1}^N \frac{1}{2}\|x_i - v_{i,k}\|^2$$
$$\text{s.t. } x_i = x_j \text{ if } (i,j)\in\mathcal{E}$$

$\rightarrow$ $\boldsymbol{x}_k^* = \mathbf{1}_N \otimes \bar{v}_k$

Solution based on distributed **relaxed ADMM algorithm**

- Asynchronous implementation/ robustness to packet losses
- No specific initialization is required
- Interesting convergence rate

Optimization in Engineering : relaxed-ADMM

$$\begin{array}{ll} \underset{x,y}{\text{minimize}} & f(x)+g(y) \\ \text{subject that} & Ax+By=c \end{array}$$

Augmented Lagrangian (λ is vector of Lagrange multipliers, $\rho > 0$)

$$\mathcal{L}_\rho(x,y,\lambda) = f(x)+g(y)-\lambda^T(Ax+By-c)+\frac{\rho}{2}\|Ax+By-c\|^2$$

Relaxed ADMM ***(Classical ADMM $\alpha = 1/2$)***

$$y(k+1) = \underset{y}{\text{argmin}}\{\mathcal{L}_\rho(x(k),y;\lambda(k)) + \rho(2\alpha-1)\langle By,(Ax(k)+By(k)-c)\rangle\}$$

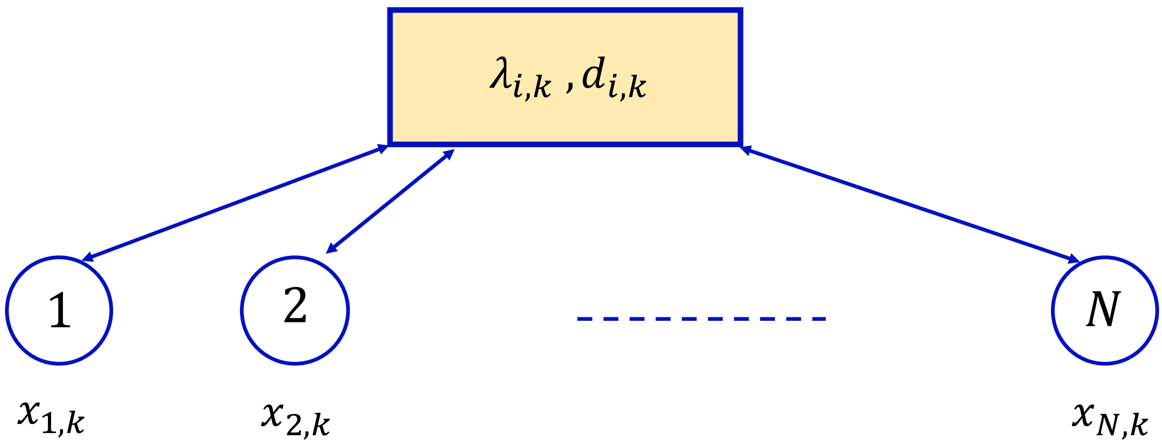
$$\lambda(k+1) = \lambda(k) - \rho(Ax(k)+By(k+1)-c) - \rho(2\alpha-1)(Ax(k)+By(k)-c)$$

$$x(k+1) = \underset{x}{\text{argmin}}\,\mathcal{L}_\rho(x,y(k+1);\lambda(k+1))$$

If $0 < \alpha < 1,\ \rho > 0$ *relaxed ADMM converges to the optimal solution*

Distributed Constraint-Coupled Optimization

Dynamic consensus ADMM (*distributed relaxed-ADMM*)



Features :

- distributed algorithm robust to packet losses, asynchrony
- no need for specific initialization
- interesting convergence rate

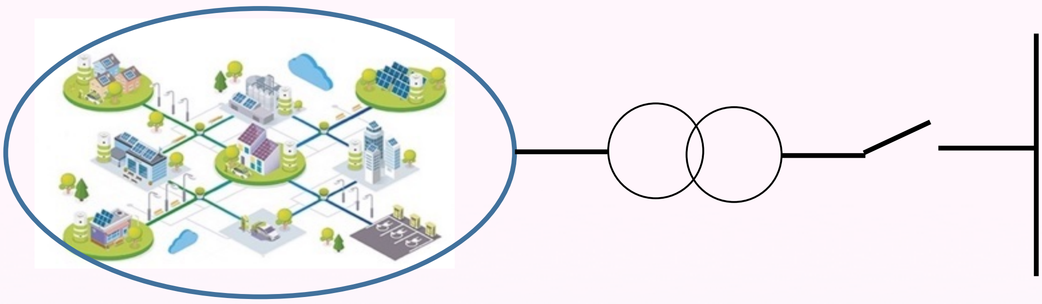
Outline

- Distributed coupled-constraint optimization
- Dynamic Average Consensus based on ADMM (Alternating Direction Method Multipliers)
- Application to energy systems (Energy communities)
- A step toward decarbonization…

Energy communities

Energy communities : legal entities that empowers *"citizens, small businesses and local authorities to* ***produce, manage and consume their own energy****."*

(*thousands projects in Europe* → *reduction CO_2 emissions)*



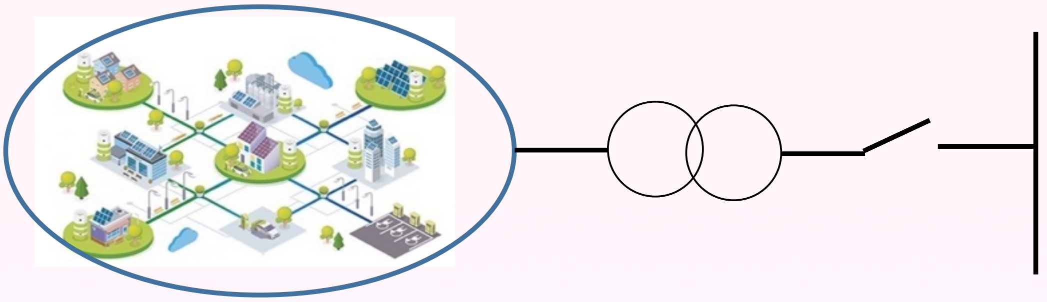
Energy community *Main distribution grid*

The concept of ***energy community*** has been introduced by the European Community in the Clean Energy for all Europeans Package (CEP) in 2019

Energy communities – Italian case

Energy communities : legal entities that empowers *"citizens, small businesses and local authorities to **produce, manage and consume their own energy**."*

(5000+ projects in Europe → reduction CO_2 emissions)



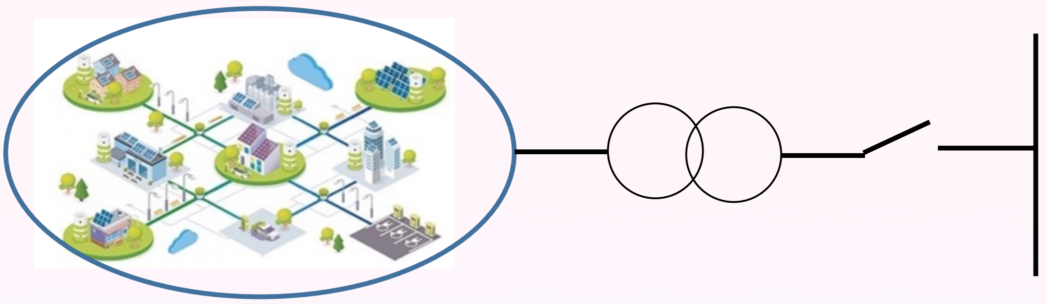
Energy community *Main distribution grid*

Incentives to build energy communities : PV panels, batteries, heat pumps…

Energy communities – Italian case

Energy communities : legal entities that empowers *"citizens, small businesses and local authorities to **produce, manage and consume their own energy**."*

(5000+ projects in Europe → reduction CO_2 emissions)



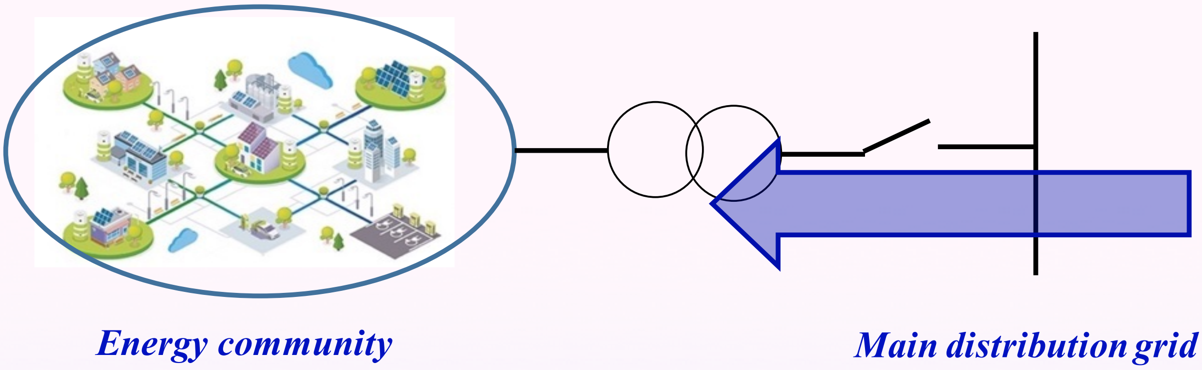
Energy community *Main distribution grid*

Incentives to maximize ***shared energy*** = *energy produced and used within the community*

Energy communities – Italian case

Energy communities : legal entities that empowers *"citizens, small businesses and local authorities to **produce, manage and consume their own energy**."*

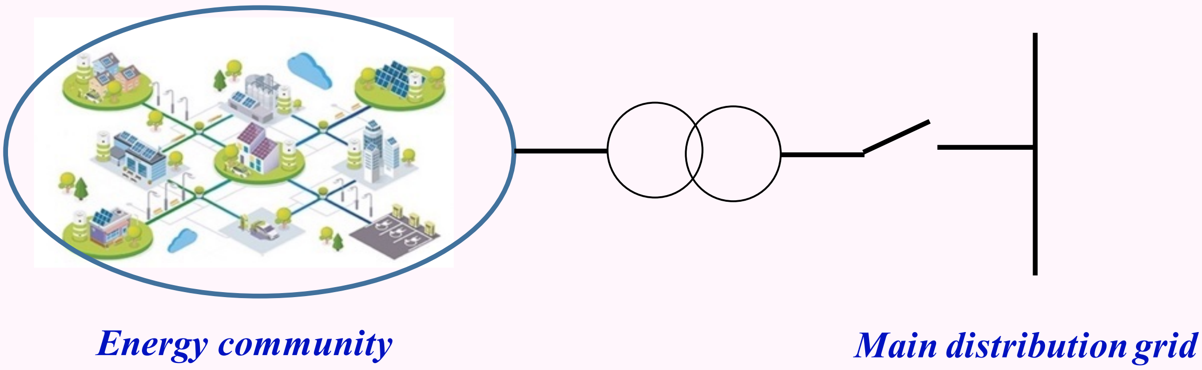
(5000+ projects in Europe → reduction CO_2 emissions)



Incentives to maximize ***shared energy*** *= energy produced and used within the community*

→ *minimization of energy required from the main grid*

Design of an Energy community

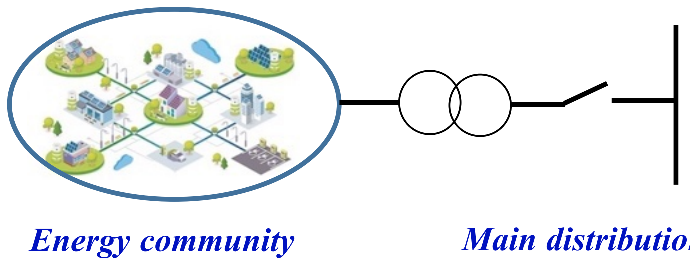


- selection of the users within an EC based on their load profiles
- selection of the energy units (photovoltaics, electrical and thermal storage, heat pumps, charging stations for electric vehicles, etc.)

↓

optimization problem (MILP) including information about users, forecast, etc.

Real-time management of ECs (active power)



$$\min_{x_1,\ldots,x_N} \quad \sum_{i=1}^{N} f_i(x_i)$$

$$\text{subject to:} \quad \sum_{i=1}^{N} A_i x_i = b$$

$$x_i \in X_i \qquad i = 1, \ldots, N$$

MAXIMIZATION
(storage, shiftable loads, charging station, etc.)

SHARED ENERGY *(incentives)*

subject to →

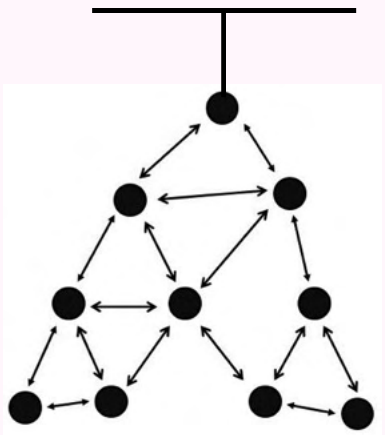
- ***local constraints*** (voltage, current, power constraints)
- ***coupled constraints*** (power balance)

including →

- prediction on the production (forecast)
- load profiles

Real-time management of ECs (active power)

Main distribution grid

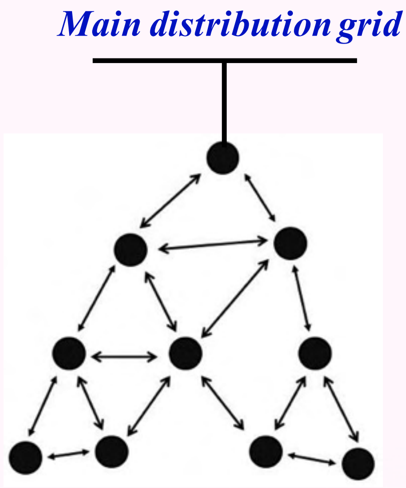


EC as a directed graph

$$\mathcal{G} = (V, \mathcal{E})$$

Each node is a **prosumer**
(= producer + consumer)

Real-time management of ECs (active power)



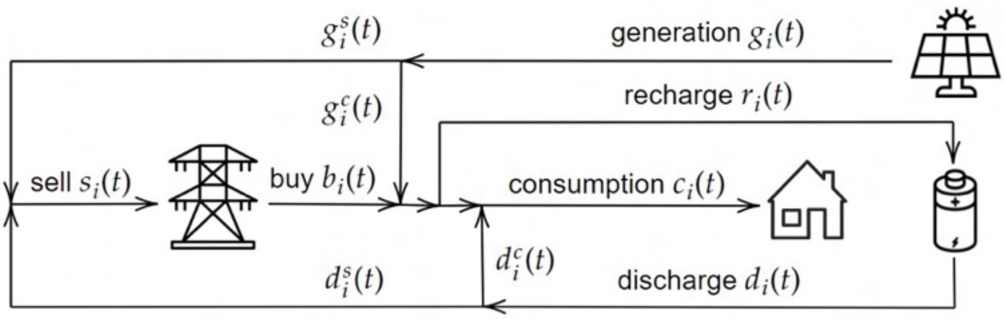
EC as a directed graph

$$\mathcal{G} = (V, \mathcal{E})$$

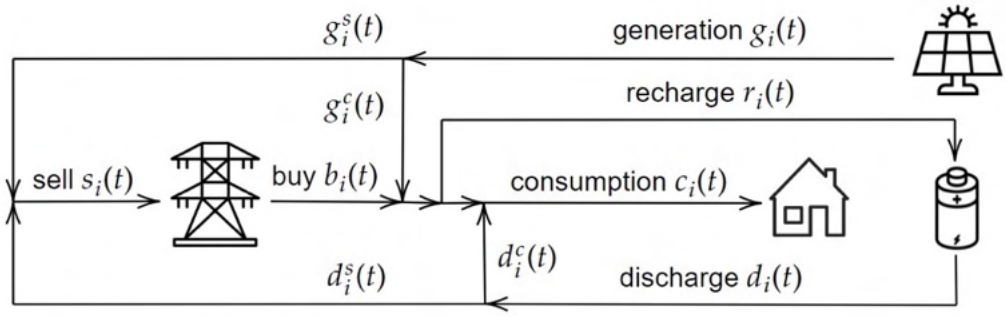
Each node is a **prosumer**
(= producer + consumer)

We consider *residential Energy community* : each prosumer has renewable generation (PV panel), battery, load profile

Modeling a prosumer (renewable production)



Modeling a prosumer (renewable production)



Renewable generation $g_i(t)$

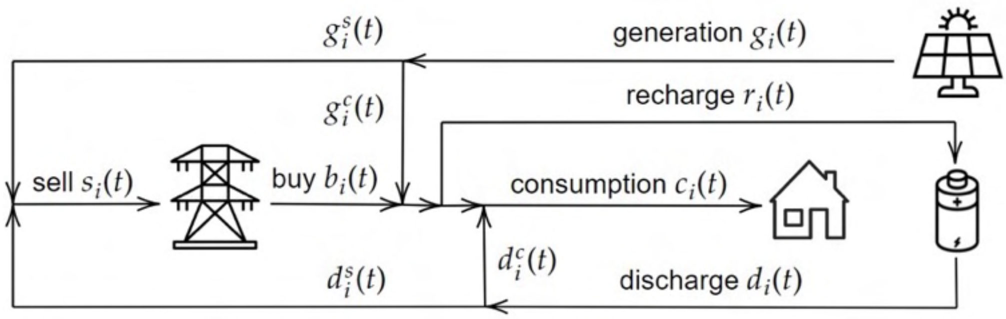
$$g_i(t) = g_i^c(t) + g_i^{sc}(t) + g_i^{sg}(t)$$

consumed by the user

sold to the grid but consumed by the community

sold to the grid but not consumed by the community

Modeling a prosumer (battery)



Battery recharging energy $r_i(t)$

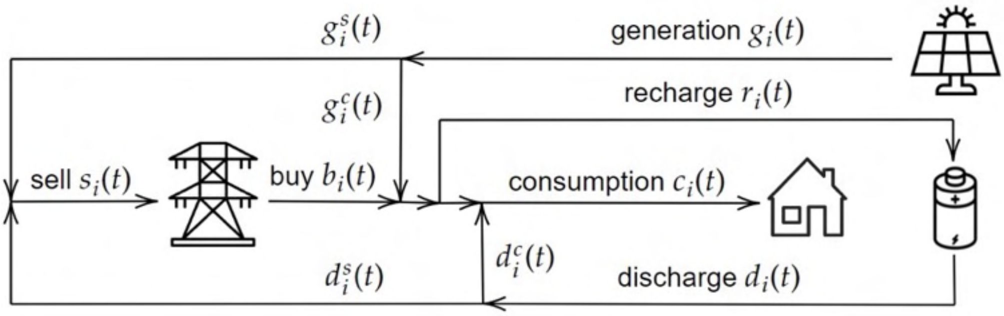
Battery discharging energy $d_i(t)$

$$d_i(t) = d_i^c(t) + d_i^s(t)$$

consumed by the user

sold to the grid

Modeling a prosumer (load)



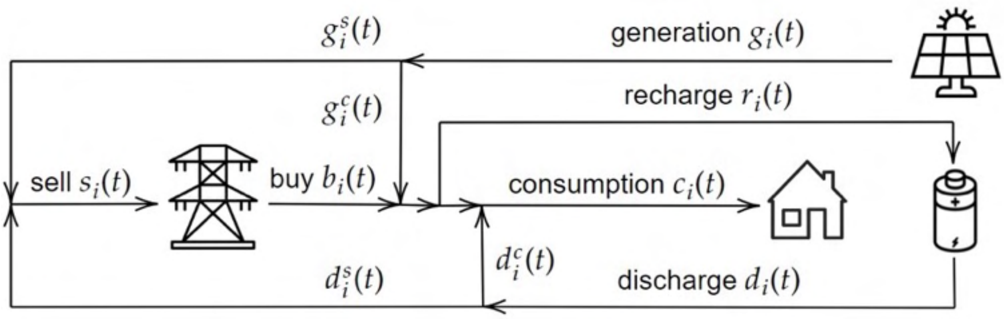
Load $c_i(t)$

$$c_i(t) = c_{F,i}(t) + c_{SL,i}(t - T_{in,i})$$

fixed load (strict requirement)

shiftable load

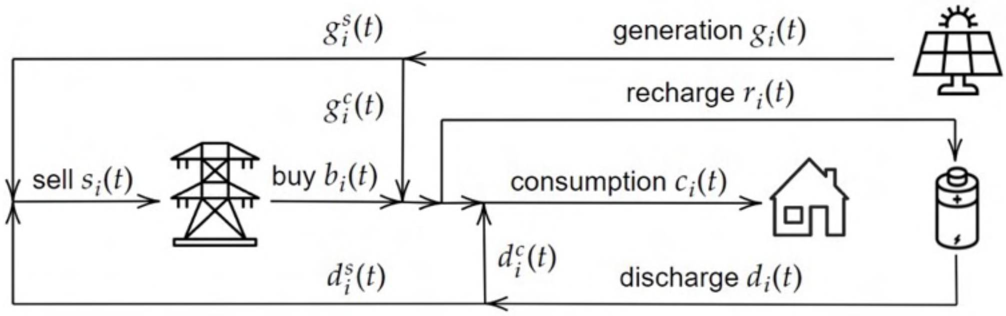
Modeling a prosumer



Bought energy $b_i(t) = r_i(t) + c_i(t) - d_i^c(t) - g_i^c(t),$

Sold energy $s_i(t) = d_i^s(t) + g_i^{sg}(t) + g_i^{sc}(t).$

Shared Energy



Shared Energy : *Is the minimum over a window between the energy injected into the grid and energy withdrawn from the grid by the users within that window.*

Incentives for shared energy!

Real-time management of ECs (active power)

Objective Function:

Minimize total energy cost → Energy Bought − Energy Shared

Linear cost

Real-time management of ECs (active power)

Objective Function:

Minimize total energy cost → Energy Bought − Energy Shared

Linear cost

Decision Variables:

- Binary variable for battery mode (charging/discharging)
- Amount of charging/discharging power for battery
- Binary variable for shiftable load (ON/OFF)

Real-time management of ECs (active power)

Objective Function:

Minimize total energy cost → Energy Bought − Energy Shared

Linear cost

Decision Variables:

- Binary variable for battery mode (charging/discharging)
- Amount of charging/discharging power for battery
- Binary variable for shiftable load (ON/OFF)

Local constraints

- Battery capacity limit
- Charging/Discharging power limit
- Load satisfaction
- Binary logic constraints

Coupled constraints

- Energy balance

Real-time management of ECs (active power)

Objective Function:

Minimize total energy cost → Energy Bought − Energy Shared

Linear cost

Decision Variables:

- Binary variable for battery mode (charging/discharging)
- Amount of charging/discharging power for battery
- Binary variable for shiftable load (ON/OFF)

Local constraints

- Battery capacity limit
- Charging/Discharging power limit
- Load satisfaction
- Binary logic constraints

Coupled constraints

- Energy balance

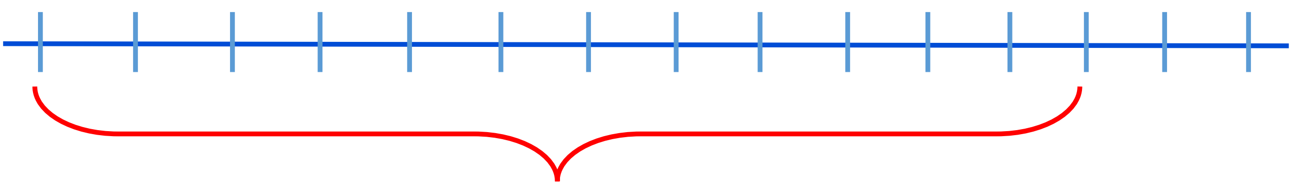
Including : weather forecast and load profiles (predictive optimization)

Real-time management of ECs (active power)

Objective Function:

Minimize total energy cost → Energy Bought − Energy Shared

Predictive optimization



Time Window

Real-time management of ECs (active power)

MILP (mixed-integer linear programming) problem

Objective Function:

Minimize total energy cost → Energy Bought − Energy Shared

Decision Variables:

- *Binary variable for battery mode (charging/discharging)*
- Amount of charging/discharging power for battery
- *Binary variable for shiftable load (ON/OFF)*

Local constraints

- Battery capacity limit
- Charging/Discharging power limit
- Load satisfaction
- Binary logic constraints

Coupled constraints

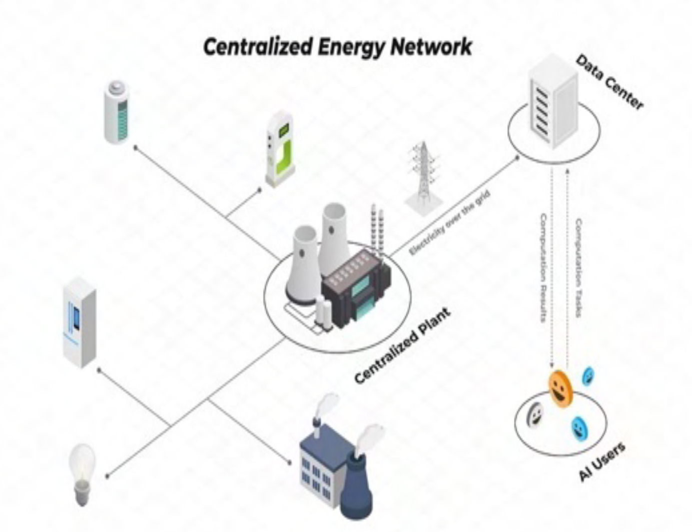
- Energy balance

Including : weather forecast and load profiles (predictive optimization)

Solution approach : centralized solution

Centralized Optimization

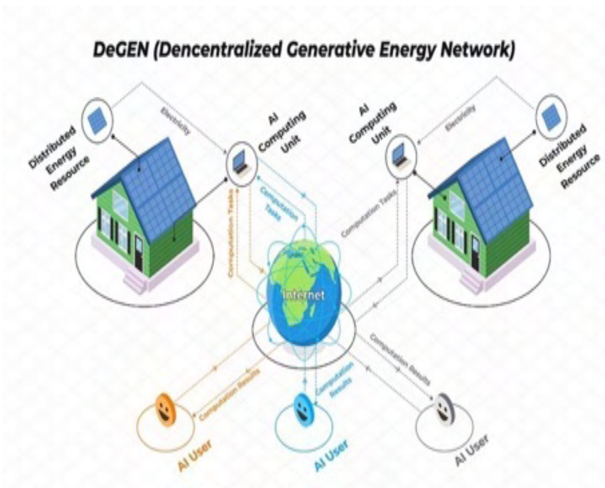
- All data collected at a central controller.
- Problem solved as a single large-scale
- Provides global optimum.
- Not scalable and raises privacy concerns.



Solution approach : decentralized solution

Distributed Optimization

- Problem decomposed into subproblems for each prosumer.
- Each subproblem solved locally by each prosumer (exchange of information only with neighbors)
- Ad-hoc strategies to guarantee coupling constraints are satisfied
- Preserves privacy and enables scalability in large energy communities.



Real-time management of ECs (active power)

MILP (mixed-integer linear programming) problem

Objective Function:

Minimize total energy cost → Energy Bought − Energy Shared

Decision Variables:

- *Binary variable for battery mode (charging/discharging)*
- Amount of charging/discharging power for battery
- *Binary variable for shiftable load (ON/OFF)*

Local constraints

- Battery capacity limit
- Charging/Discharging power limit
- Load [illegible]
- Binary logic constraints

Coupled constraints

- Energy balance

$$\min_{x_1,\dots,x_N} \quad \sum_{i=1}^{N} f_i(x_i)$$

$$\text{subject to:} \quad \sum_{i=1}^{N} A_i x_i = b$$

$$x_i \in X_i \qquad i = 1, \dots, N$$

Including : weather forecast and load profiles (predictive optimization)

Real-time management of ECs (active power)

MILP (mixed-integer linear programming) problem

$$\min_{x_1,\dots,x_N} \sum_{i=1}^{N} f_i(x_i)$$

$$\text{subject to: } \sum_{i=1}^{N} A_i x_i = b$$

$$x_i \in X_i \qquad i = 1,\dots,N$$

Objective Function:

Minimize total energy cost

Energy Bought − Energy Shared

Decision Variables:

- Binary variable for battery mode (charging/discharging)
- Amount of charging/discharging power for battery
- Binary variable for shiftable load (ON/OFF)

Local constraints

- Battery capacity limit
- Charging/Discharging power limit
- Load satisfaction
- Binary logic constraints

Coupled constraints

- Energy balance

Including : weather forecast and load profiles (predictive optimization)

Real-time management of ECs (active power)

MILP (mixed-integer linear programming) problem

$$\min_{x_1,\dots,x_N} \sum_{i=1}^{N} f_i(x_i)$$

$$\text{subject to: } \sum_{i=1}^{N} A_i x_i = b$$

$$x_i \in X_i \qquad i = 1,\dots,N$$

Energy Bought − Energy Shared

Local constraints

- Battery capacity limit
- Charging/Discharging power limit
- Load satisfaction
- Binary logic constraints

Coupled constraints

- Energy balance

- Binary variable for battery mode (charging/discharging)
- Amount of charging/discharging power for battery
- Binary variable for shiftable load (ON/OFF)

Including : weather forecast and load profiles (predictive optimization)

Real-time management of ECs (active power)

MILP (mixed integer linear programming) problem

Objective Function:

Minimize total energy cost

$$\min_{x_1,\ldots,x_N} \sum_{i=1}^{N} f_i(x_i)$$

$$\text{subject to:} \quad \sum_{i=1}^{N} A_i x_i = b$$

$$x_i \in X_i \qquad i = 1, \ldots, N$$

Energy Bought − Energy Shared

Local constraints

- Battery capacity limit
- Charging/Discharging power limit
- Load satisfaction
- Binary logic constraints

Coupled constraints

- Energy balance

X_i is not convex!

Real-time management of ECs (active power)

$$\min_{x_1,\ldots,x_N} \sum_{i=1}^{N} f_i(x_i)$$

$$\text{subject to:} \quad \sum_{i=1}^{N} A_i x_i = b$$

$$x_i \in \operatorname{conv}(X_i) \qquad i = 1, \ldots, N$$

Real-time management of ECs (active power)

$$\min_{x_1,\dots,x_N} \quad \sum_{i=1}^{N} f_i(x_i)$$

$$\text{subject to:} \quad \sum_{i=1}^{N} A_i x_i = b$$

$$x_i \in \text{conv}(X_i) \qquad i$$

$\bar{x}_i,\ \bar{\delta}_i,\ \bar{\ell}_i$

Algorithm 1 Tracking-ADMM

1: **Initialization**

2: $\quad \boldsymbol{x_{i,0}} \in \text{conv}(X_i)$

3: $\quad d_{i,0} = A_i x_{i,0} - b_i$

4: $\quad \lambda_{i,0} \in \mathbb{R}^p$

5: **Repeat until convergence**

6: $\quad \delta_{i,k} = \sum_{j \in \mathcal{N}_i} w_{ij}\, d_{j,k}$

7: $\quad \ell_{i,k} = \sum_{j \in \mathcal{N}_i} w_{ij}\, \lambda_{j,k}$

8: $\quad x_{i,k+1} \in \underset{x_i \,\in\, \text{conv}(X_i)}{\text{argmin}} \Big\{ f_i(x_i) + \ell_{i,k}^{\top} A_i x_i + \frac{c}{2} \|A_i x_i - A_i x_{i,k} + \delta_{i,k}\|^2 \Big\}$

9: $\quad d_{i,k+1} = \delta_{i,k} + A_i x_{i,k+1} - A_i x_{i,k}$

10: $\quad \lambda_{i,k+1} = \ell_{i,k} + c\, d_{i,k+1}$

Real-time management of ECs (active power)

$$\min_{x_1,\dots,x_N} \quad \sum_{i=1}^{N} f_i(x_i)$$

$$\text{subject to:} \quad \sum_{i=1}^{N} A_i x_i = b$$

$$x_i \in \text{conv}(X_i) \qquad i$$

$\bar{x}_i,\ \bar{\delta}_i,\ \bar{\ell}_i$

$$x_i^* = \underset{x_i \in X_i}{\text{argmin}} \Big\{ f_i(x_i) + \bar{\ell}_i^T A_i x_i + \frac{c}{2\rho} \|A_i x_i - A_i \bar{x}_i + \bar{\ell}_i\|^2 \Big\}$$

Algorithm 1 Tracking-ADMM

1: **Initialization**

2: $\quad \boldsymbol{x_{i,0}} \in \text{conv}(X_i)$

3: $\quad d_{i,0} = A_i x_{i,0} - b_i$

4: $\quad \lambda_{i,0} \in \mathbb{R}^p$

5: **Repeat until convergence**

6: $\quad \delta_{i,k} = \sum_{j \in \mathcal{N}_i} w_{ij}\, d_{j,k}$

7: $\quad \ell_{i,k} = \sum_{j \in \mathcal{N}_i} w_{ij}\, \lambda_{j,k}$

8: $\quad x_{i,k+1} \in \underset{x_i \,\in\, \text{conv}(X_i)}{\text{argmin}} \Big\{ f_i(x_i) + \ell_{i,k}^{\top} A_i x_i + \frac{c}{2} \|A_i x_i - A_i x_{i,k} + \delta_{i,k}\|^2 \Big\}$

9: $\quad d_{i,k+1} = \delta_{i,k} + A_i x_{i,k+1} - A_i x_{i,k}$

10: $\quad \lambda_{i,k+1} = \ell_{i,k} + c\, d_{i,k+1}$

Real-time management of ECs (active power)

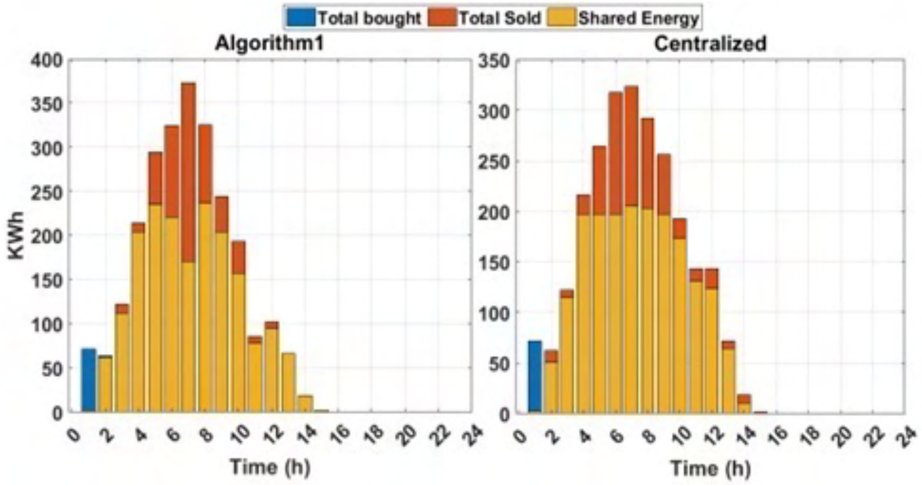


Fig. 2. The total energy bought and sold by agents and the corresponding shared energy: Centralized vs. distributed Algorithm [1]

M. Messilem, R. Brumali, G. Carnevale, G.Notarstefano, R. Carli. *A distributed MILP-ADMM framework for Italian Energy Communities: Shared Energy Incentives. Renewables and Shiftable Loads* In Sensys 2025

Real-time management of ECs (active power)

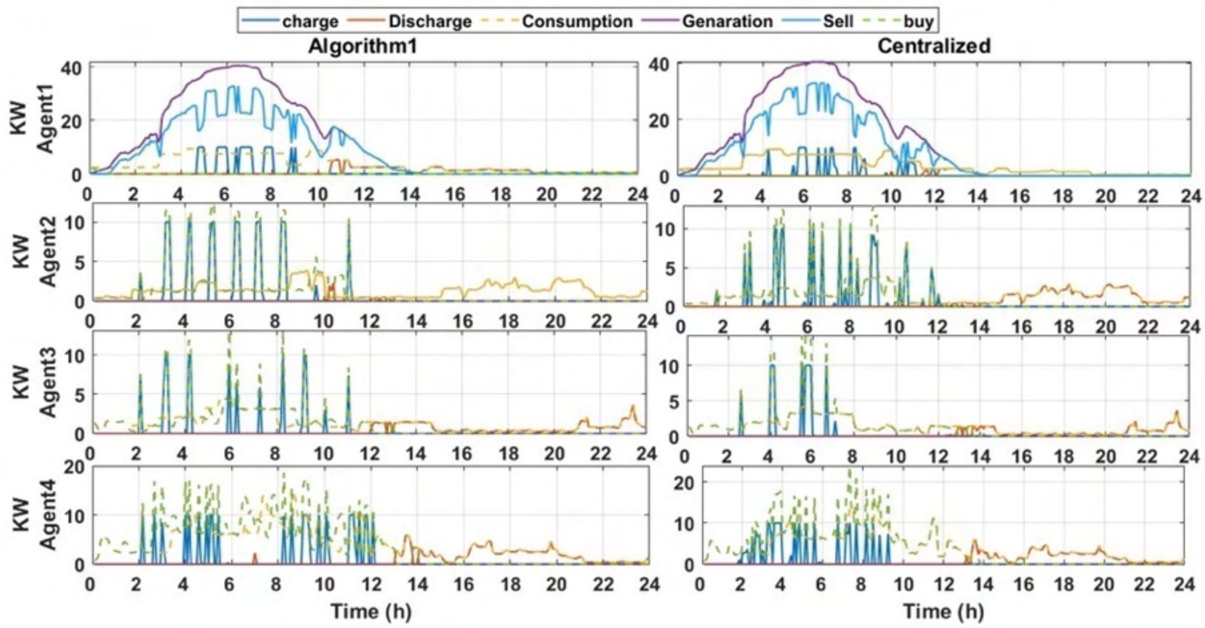
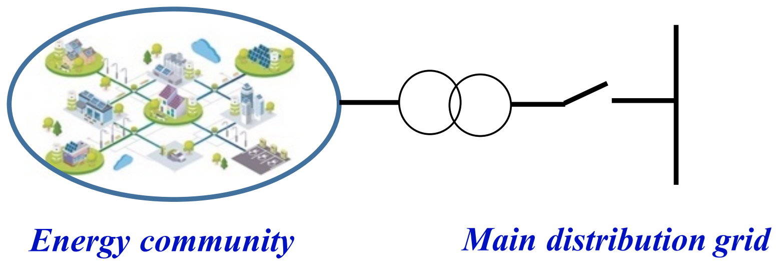


Fig. 3. Evolution of power exchange in the network: Centralized vs. distributed Algorithm [1]

Real-time management of ECs (ancillary services)

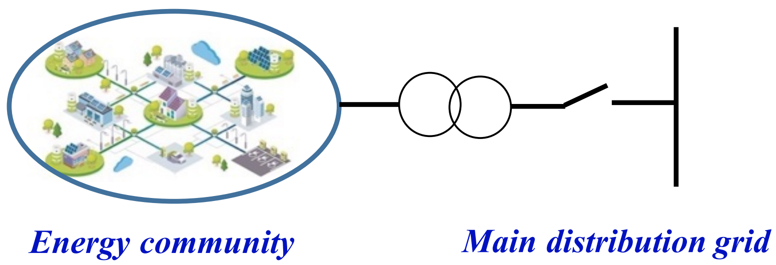


$$\min_{x_1,\dots,x_N} \quad \sum_{i=1}^{N} f_i(x_i)$$
$$\text{subject to:} \quad \sum_{i=1}^{N} A_i x_i = b$$
$$x_i \in X_i \qquad i = 1,\dots,N$$

Electronic power converters are extensively applied in microgrids to interface distributed energy resources to the grid. They provide :

- primary active-power control *(to maximize shared energy!)*
- other *ancillary services* – exploiting additional degrees of freedom
  - *minimization of power losses,*
  - *regulation of reactive power,*
  - *unbalance compesantion at PCC*

Real-time management of ECs



$$\min_{x_1,\dots,x_N} \quad \sum_{i=1}^{N} f_i(x_i)$$
$$\text{subject to:} \quad \sum_{i=1}^{N} A_i x_i = b$$
$$x_i \in X_i \qquad i = 1,\dots,N$$

Control architecture for the management of Energy Communities

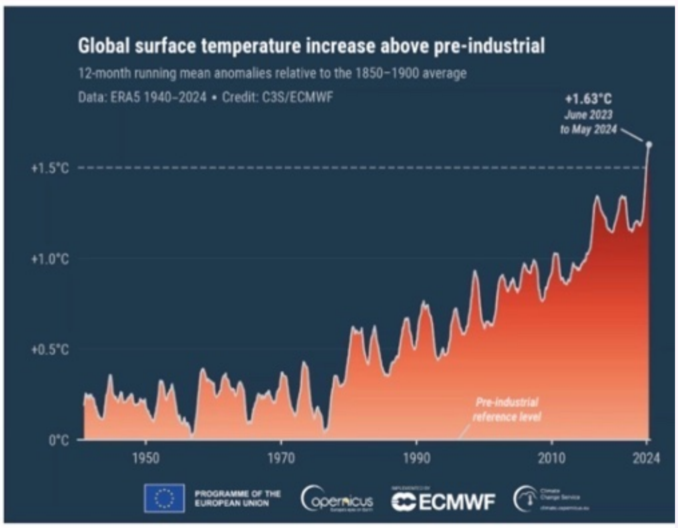
- *Energy management in Energy Communities* (time-scale $\sim$ 15min)
- Ancillary services improving *power quality* (shorter time-scale $\sim$ 1min)
- Optimization over *networks of energy communities* (longer time-scale $\sim$ hours)

Outline

- Distributed coupled-constraint optimization
- Dynamic Average Consensus based on ADMM (Alternating Direction Method Multipliers)
- Application to energy systems (Energy communities)
- A step toward decarbonization…

Decarbonization to mitigate global warming

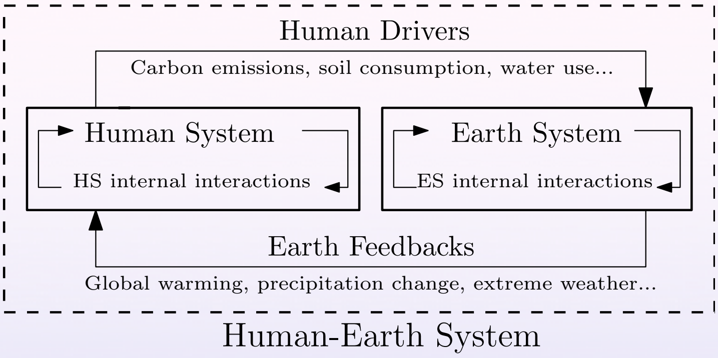
Development of Energy communities can be a promising action to decarbonization



Steps toward decarbonization

How can control theory contribute? Just few suggestions:

- modeling and optimization of systems for capturing CO_2 - for instance microalgae cultivation in photobioreactors
- hydrogen batteries – modeling and control challenges
- Data-driven models for the evolution of animal species due to climate changes (risk of extiction)
- Analysis of Human-Earth systems



Collaborators

- Ivano Notarnicola, Guido Carnevale, R. Brumali, Giuseppe Notarstefano (Unibo)
- Mohamed Messilem (Unipd)
- Tommaso Caldognetto, Andrea Lauri (Unipd)

Questions?