



**HAL**  
open science

# A distributed scheduling method for demand response in energy communities with distributed generation and storage

Mariam Sangaré, Eric Bourreau, Michael Poss

## ► To cite this version:

Mariam Sangaré, Eric Bourreau, Michael Poss. A distributed scheduling method for demand response in energy communities with distributed generation and storage. 2023. <hal-04188878v4>

**HAL Id: hal-04188878**

**<https://hal.science/hal-04188878v4>**

Preprint submitted on 14 Nov 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# A distributed scheduling method for demand response in energy communities with distributed generation and storage

Mariam Sangaré<sup>1</sup>, Eric Bourreau<sup>1</sup>, and Michael Poss<sup>1</sup>

<sup>1</sup> LIRMM, University of Montpellier, Montpellier, France  
✉{eric.bourreau,michael.poss,mariam.sangare}@lirmm.fr

This paper presents a distributed optimization method for energy communities having distributed renewable generation and storage units. We explain how the resulting optimization problem can be cast as a bi-level optimization problem where the followers solve mixed-integer linear programs. Given the difficulty of these problems, we develop a heuristic where incentives are sent to the followers. Specifically, to maximize the collective-self consumption rate, we exploit the notion of allocation key, which traditionally used a posteriori for economic gain sharing to build an a priori incentive to demand response. The proposed distributed management method returns high-quality solutions to the optimal centralized real-world benchmark instances constructed over a month with accurate historical data from our demonstrator in South France.

**Keywords**— Distributed management, Demand response, Heuristic algorithms, Mixed-Integer Programming.

## 1 Introduction

Global economic and demographic growth is driving a significant increase in energy demand. Against this backdrop, governments and communities strive to increase the production and consumption of renewable energies. States are developing regulations to promote access to production and storage tools on the one hand [Clean energy for all Europeans package](#) and [Directive \(EU\) 2018/2001](#), and surplus trading on the other hands: [Code de l'énergie](#), [Décret n° 2017-676](#) and [PUCA](#). Thus, collective self-consumption appeared in France in 2015, enabling renewable energy generators to exchange their surpluses with specific individuals.

The French Energy Code defines collective self-consumption as an operation linking one or more producers to one or more consumers using the public electricity distribution network for their energy exchanges. In addition to the social benefits, the advantages of collective self-consumption include participating in the energy transition, saving money on energy bills, reducing fuel poverty, and minimizing energy generation surpluses. These energy and economic challenges require optimization tools throughout the energy community lifecycle from the planning point of view.

Thus, the authors of [17, 20] address long-term planning. In [20], the authors study a power system extension model; the goal is to determine the investment to expand a multi-commodity power system. To do so, the authors consider Demand Response a capacity reserve. The resulting problem is uncertain since DR is uncertain even over short-term planning. While considering uncertainty in energy systems' management, many kinds of uncertain parameters make robust models untenable given the systems' size; [17] presents a robust optimization framework to address this issue. Turning to medium-term planning, the authors of [26] present a single leader and several followers bi-level problem. The leader interacts with each follower via a contract detailing the parties' exchange prices. The leader's problem is choosing the contract to propose to each follower to maximize his profit, given that the members may or may not accept them. If a follower refuses the leader's contract, she can exchange with one of the leader's competitors. Each follower's objective is to choose the least expensive contract. Finally, the authors of [24] present a short-term planning model; the authors deal with the decentralized coordination of flexible loads and uncertain renewable generation. The authors rely on demand-side incentives to attract low-cost flexibility in a decentralized and uncertain context to balance supply and demand in each period.

Two optimization approaches to energy system planning exist in the literature: centralized and distributed optimization. Centralized optimization methods, which consist of a central manager collecting information and preferences from all members, solving an optimization problem then sending signals to each community member, are widely used in the literature [1, 19, 25] as well as the aforementioned [20, 26]. While they can return optimal solutions, they have some drawbacks:

- The large computational capacity requirement for the centralized manager.
- The information bottlenecks that lead to long transfer times.
- The question of social acceptance, since people may be reluctant to accept instructions from a third party regarding using their devices [21].

Therefore, distributed optimization is a relevant alternative for large electrical systems; it refers to each member locally determining their loads. Thus, the coordinator does not bear the computational effort since each member locally solves a local optimization problem. The local optimization problems are often some linear program (LP). Some of the most used distributed optimization methods are the alternating direction method of multipliers (ADMM) [4, 11, 12, 13, 23], dual decomposition [15] and Consensus + innovations [7, 9].

Regarding ADMM, [12] proposes a method for day-ahead scheduling and compares the centralized and distributed approaches. Reference [11] presents a distributed model for multi-microgrids that cooperate to reduce the aggregated operational cost. The authors of [23] present a community that cannot inject the surplus in the distribution network and an ADMM-based method to maximize the community’s utility. The authors use game theory to model members’ preferences. Each member solves an optimization problem to maximize the local utility. Thus, global problems are solved under global constraints, such as the battery storage system (BSS) constraints. On a ring network composed of several agents, the next agent  $n$  starts to calculate and update only after receiving information from the previous one, which is problematic for complex problems. Thus, [4] presents a parallel and distributed method for energy management to increase ADMM’s convergence rate.

ADMM works in a cascade, making it ineffective for large systems. Reference [27] presents a distributed Lagrangian dual method that rapidly converges based on Nesterov accelerated gradient and dynamic set sizes. In [10], the author generalizes the Lagrangians to a broader class of functions that satisfies the strong duality between primal and dual problems and then derives the generalized Karush–Kuhn–Tucker conditions for this generalized Lagrange multiplier method, which is helpful to provide consensus protocols.

Turning to Consensus + innovation, [7] presents an approach to coordinate the loads, energy generation, and storage for a set of auto-sufficient agents of a microgrid. Each agent interacts with a defined neighborhood to find a consensus on the power supply price while ensuring the total generation and load are equal. The authors of [9] provide a relaxed consensus + innovation-based management to coordinate the power exchanges and prices among the microgrid and the smart grid and study the uncertainty impact on such system’s performance.

In this paper, we introduce a bilevel formulation to manage a community where the members do not directly interact. Thus, there are no peer-to-peer exchanges and no interaction

between community members. The problem’s generic form is:

$$\min_{x^L \in X^L, x^F \in X^F} \quad \Pi^L(x^L, x^F) \quad (1a)$$

$$\text{s.t.} \quad x^F \in \arg \max_{x^F \in X^F} \sum_{n \in \mathcal{N}} \Pi_n^F(x^L, x^F), \quad (1b)$$

where  $x^L$  and  $x^F$  are the leader’s and followers’ decision variables, and  $\Pi^L$  is the leader’s objective function, which in our case is the energy coming from the primary grid. Each follower  $n$ ’s objective function  $\Pi_n^F$  is the gain, which is negative if  $n$ ’s global need is greater than the energy availability; it is positive otherwise. The bilevel formulation is hardly solvable due to integer variables in the followers’ problem and the problem’s size. Thus, we introduce a heuristic where CC provides the surplus allocation rule called allocation key as an incentive, which is the main methodological contribution of this paper. The notion of allocation key is a well-known concept used (a posterior) to share the economic gain between the members who share their generation surplus [18]. In this paper, we use the concept of allocation key to share the energy injected by prosumers in the community for day-ahead scheduling. Thus, we define the allocation key as an algorithm that returns the maximum amount of energy a given member can collect from the community at each period. After receiving the incentive, each member determines their load, need, withdraws, and/or storage and injection schedules. Then, she sends her energy needs and availability information to CC, which updates its incentives. Since there are no direct information links between each pair of members, prosumer  $n$  does not know which part of the injection is consumed in the community. If the total periodic injection exceeds the total needed, the remaining energy is injected into the grid; otherwise, the remaining required energy is collected from the primary grid. The main aim is to consume local generation as much as possible locally.

We detail next the differences between this work and [19], who share some authors with this paper and study a problem that might look similar to the one studied in this paper at first glance. On the one hand, the operational constraints regarding the management of batteries and the scheduling of tasks are similar in both works, justifying the use of the same nomenclature. On the other hand, the crucial difference between the two works lies in how the electrical exchanges among community members are modeled. In [19], we assumed that all decisions are taken by a centralized entity, omniscient. This led to solving a single Mixed-Integer Linear Program, for which a column-generation-based heuristic was also proposed. However, this assumption is not realistic in most situations which is why we propose herein a distributed approach. This leads to the existence of a coordinator and the use of a bilevel formulation. Because of the bi-level nature of the problem, the heuristic solution

we propose herein, based on the aforementioned allocation keys, fundamentally differs from the centralized approach proposed in [19]. Therefore, the algorithmic developments and numerical experiments proposed in the two papers are completely different.

The rest of the paper is structured as follows: Section 2 describes the problem we study. Section 3 details each level of formulation (11). Section 4 presents an iterative heuristic, returning the efficient solution. Section 5 presents the experimental results on a realistic instance, while Section 6 reports more results based on real-world data. The paper concludes in Section 7.

## 2 Problem description

We consider a set of  $N$  individuals involved in a collective self-consumption project, each having different asset possession characteristics as schematized in Figure 1. Figure 1 is presented as an indication that there are various profiles of individuals in the community. In this work, an energy community refers to individuals connected to the same distribution network that produces, consumes, and exchanges energy. We specifically consider renewable energy communities as defined by the package [Clean energy for all Europeans](#). The members who generate energy can store their surpluses and exchange them within the community or the main grid via an aggregator. We assume the local exchange prices are more attractive than those with the main grid. Thus, local producers prioritize local exchange over main grid injection, maximizing their profit. However, in such a community, no additional links exist between members apart from being in the same community, i.e., they do not directly interact. A community coordinator (CC) is the intermediary between members to ensure a given management objective. For example, CC ensures the community’s functioning in terms of collective self-consumption. She interacts with each member, collecting specific information and sharing incentives to demand response (DR) to ensure the whole community’s management. DR is a load profile modulation granted by a member in return for a certain amount of consideration. Each member remains directly connected to the primary grid and collects energy when needed. Our partners have a margin of each kWh of energy locally exchanged in the community. An aggregator collects each producer’s surpluses; for simplicity, we assume there is a unique aggregator for all community members.

The coordinator fixes the periodic local energy exchange prices. However, to preserve trust in the community, these prices are not differentiated, i.e., the prices are the same for each member. Thus, more than the price-based incentive is required in such a system. We do not address the pricing in this paper. CC must build some energy-based incentives to

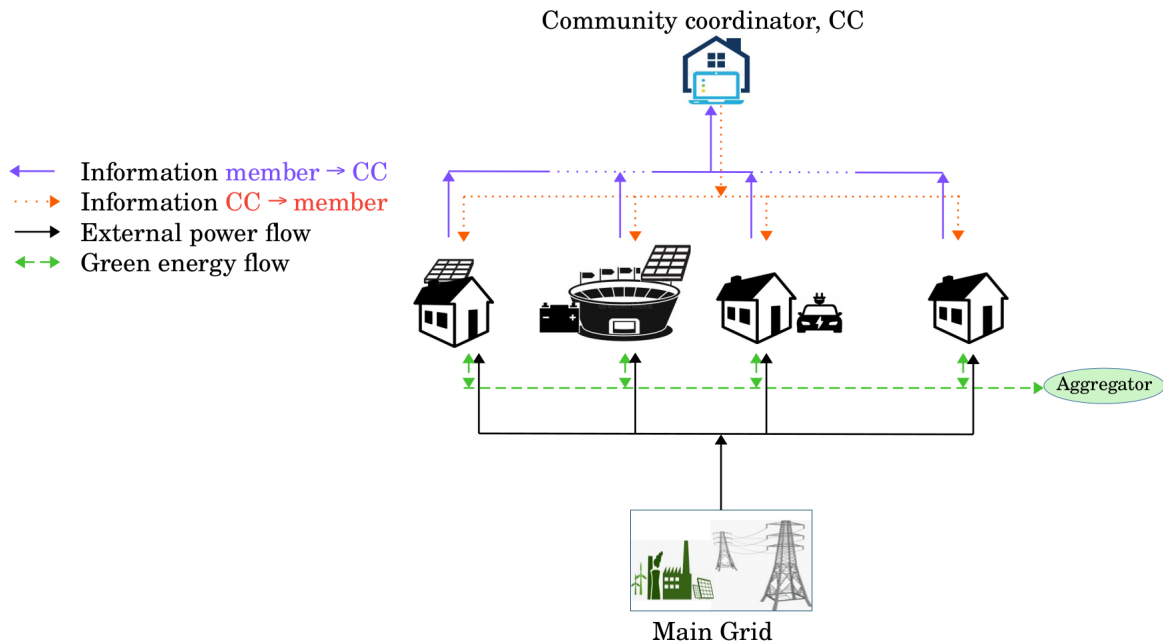


Figure 1: Community scheme.

maximize the collective self-consumption rate since there is no price differentiation. Thus, CC receives each member’s hourly energy demand and prosumers’ energy availability to compute the energy-based incentives. These incentives aim to limit the periodic energy exchanges with the aggregator and the main grid, which is equivalent to maximizing collective self-consumption. They also incentivize prosumers having batteries to store and inject when needed during planning horizon to serve the community or the primary grid. Notice the more green energy is exchanged in the community, the more the coordinator realizes the economic gain as they are correlated; this allows CC to help such a community unleash its potential fully ([5]).

As shown in Figure 1, a member can be a traditional consumer, a prosumer with a battery storage system (BSS), or a consumer owing BSS. Following [25], we assume each member has the required control and exchange equipment for this study. Each member must perform some tasks that we classify into three classes according to their characteristics. The first class contains tasks that allow modulation of specific environments’ temperatures (house room, water heater). We call this type A class. Each member must decide the periodic power consumption to reach the desired temperature at the expected time. We use function  $\theta_j$  to evaluate the temperature in member  $n$ ’s room where task  $j \in J^A$  is performed.

- The temperature variation function derived from the [Newton cooling law](#) is:

$$\theta_1(p_h) = \theta_1(p_{h-1}) + \frac{\Delta}{C_r}(p_h - U(\theta_1(p_{h-1}) - T_h^{\text{out}})),$$

for house rooms. Where  $C_r$  is heat capacity (J/K), parameter  $U$  designates the heat loss coefficient of a room in (W/K), and  $\Delta$  is the heating time in seconds.  $T_h^{\text{out}}$  is the external temperature at period  $h$ . Notice that  $\theta_1$  works for both heating and cooling.

- For a water heater, using the method from [22], the temperature variation function  $\theta_2$  is given by

$$\theta_2(p_h) = \frac{\Delta_h}{Mc_p} \left( -\frac{S}{R}[\theta_2(p_{h-1}) - T_h^{\text{room}}] + \nu p_h - 1000M_w c_p[\theta_2(p_{h-1}) - T_{in}] \right) + \theta_2(p_{h-1}),$$

where  $c_p$  is the isobaric specific heat capacity of water (kcal/kg·°C),  $\nu$  is the efficiency of the electricity-to-heat transformation,  $M$  is the weight of the water (kg). The data  $S$  designates the exchange surface of the water container with the external area,  $R$  is the thermal resistance of the tank insulation in  $m^2 \cdot ^\circ\text{C}/\text{W}$  and  $M_w$  is the average hot water demand rate during the time interval, which we assume to be equal to zero since we don't have that information. Parameter  $T_{in}$  is the supply domestic cold water temperature, and  $T^{\text{room}}$  is the ambient environment temperature. Finally,  $\Delta_h$  heating time in seconds.

The schedules start at period 1. We have the initial period  $h = 0$  where no decision is required, i.e.,  $p_0 = 0$ . In addition, each task  $j \in J^A$  that is performed in room  $k$  of member  $n$  has a power  $\bar{p}_{njk} > 0$ , expressed in kW. In this work, we discretize  $[0, \bar{p}_{njk}]$  by considering only the integer values in  $[1, \bar{p}_{njk}]$  (0 is not considered as in that case, the device is turned off). Our numerical experiments consider devices with  $\bar{p}_{njk} = 2$ .

The second class, called type B, contains the tasks that must be performed during defined periods under some energy consumption level constraint (charging the electric vehicle or washer). For example, each member  $n$  must decide a schedule in a set of feasible schedules  $S_{nj}$  to perform task  $j$ . The last class, called type C class, contains uncontrollable tasks. The periodic energy consumption of type C tasks is estimated for each member during the planning horizon. The question we address in this paper is: How can we efficiently perform tasks in a community to maximize the collective self-consumption rate?

### 3 Mathematical formulation

In this section, we detail the bilevel formulation (11). CC is the leader, and each community member is a follower. Let us first detail the input data and followers' variables, provided in Tables 1 and 2. In addition, the leader's variables are the maximum amount of energy a given member can take from the community. We detail next the main components of (11).

Table 1: Data

Input data	
$\mathcal{N}$	set of community's members, we denote $ \mathcal{N} $ as $N$ .
$\mathcal{H}$	set of the planning horizon's periods, we denote $ \mathcal{H} $ as $H$ .
$J^A$	set of type A tasks.
$J_n^B$	set of member $n$ 's type B tasks.
$K_{nj}^A$	set of rooms of member $n$ where task $j \in J^A$ can be performed.
$S_{nj}$	set of schedules given by $n \in \mathcal{N}$ to perform task $j \in J^B$ .
$P_{nj}^A$	set of the power levels available to perform task $j \in J^A$ in room $k \in K_{nj}^A$ of member $n \in \mathcal{N}$ .
$\Delta_h$	length of period $h$ ( <i>hour</i> ).
$\pi_n$	member $n$ 's subscribed power level.
$\Gamma_n$	member's $n$ 's BSS's capacity.
$\xi_n$	initial state of charge of member $n$ 's BSS $n \in \mathcal{N}$ .
$\eta_n$	member $n$ 's BSS's automatic discharge rate of battery $n \in \mathcal{N}$ .
$\phi_n$	member $n$ 's BSS's maximum cycles during horizon $H$ $n \in \mathcal{N}$ .
$\beta$	maximum spending degradation threshold allowed (%).
$\Omega_n$	equal to 1 if member $n$ is allowed to exchange electricity, 0 otherwise.
$d_n, c_n$	discharge and charging efficiencies of $n$ 's BSS $n \in \mathcal{N}^{\text{BSS}}$ (%).
$\nu_{nj}$	equal 1 if task $j \in J^A$ is executed by member $n$ in room $k \in K_{nj}^A$ , 0 otherwise.
$\theta_j(p)$	temperature variation function when performing task $j \in J^A$ according to power $p$ .

Continued on next page

Table 1: Data (Continued)

Input data	
$[t_{nj k}^{\text{low}}, t_{nj k}^{\text{up}}]$	temperature targeted by member $n$ when performing task $j \in J^A$ in room $k$ ( $^{\circ}\text{C}$ ).
$[h_{nj k}^{\text{low}}, h_{nj k}^{\text{up}}]$	time where member $n$ 's comfort zone must be reached when performing task $j \in J^A$ .
$\bar{T}_{n1 k}^{\text{room}}$	initial temperature of room $k \in K_{n1}^A$ where member $n$ wants to perform task $j = 1$ .
$\bar{T}_{n2 k}^{\text{water}}$	initial water's temperature in heater $k \in K_{n2}^A$ where $n$ wants to perform task $j = 2$ .
$\bar{T}_{nj k}^{\text{all}}$	$\bar{T}_{nj k}^{\text{all}} = \bar{T}_{n1 k}^{\text{room}}$ if $j = 1$ , $\bar{T}_{nj k}^{\text{all}} = \bar{T}_{n2 k}^{\text{water}}$ if $j = 2$ .
$T_h^{\text{out}}$	outside temperature at time period $h \in \mathcal{H}$ ( $^{\circ}\text{C}$ ).
$T_{nkh}^{\text{room}}$	ambient temperature in the room containing water heater $k \in K_{n2}^A$ at period $h$ ( $^{\circ}\text{C}$ ).
$P_{nh}^{\text{PV}}$	energy production of member $n$ at period $h$ (kW).
$P_{njhs}^{\text{B}}$	consumption of task $j \in J_n^{\text{B}}$ of member $n$ at period $h$ in the schedule $s \in S_{nj}$ (kW).
$c_n, d_n$	charging and discharging power of member $n$ 's BSS (kW).
$p_{nh}^{\text{C}}$	cumulative power consumption of type C tasks at time period $h \in \mathcal{H}$ .
$v_h^{\text{MG}}$	unit purchase price of electricity from the primary grid during period $h \in \mathcal{H}$ ( $\text{€}/\text{kWh}$ ).
$v_h^{\text{Com}}$	unit purchase price of electricity in the community during period $h \in \mathcal{H}$ ( $\text{€}/\text{kWh}$ ).
$\tilde{v}_h^{\text{Com}}$	unit sale price of electricity in the community during period $h \in \mathcal{H}$ ( $\text{€}/\text{kWh}$ ).

### 3.1 Leader's problem

CC determines the periodic incentive to maximize the collective self-consumption rate. This incentive that we note  $F_{nh}^{\text{RCom}}$  is the maximum amount of energy member  $n$  can draw from the community at period  $h$ . The leader shares the energy between members via  $F^{\text{RCom}}$ , and

$x_{njkh}^A \in \{0, 1\}$	is equal to 1 if and only if task $j \in J^A$ is in progress at period $h$ in room $k \in K_{nj}^A$ of member $n$ , and the device is on power level $p$ .
$x_{nj}^s \in \{0, 1\}$	is equal to 1 if and only if schedule $s \in S_{nj}$ is chosen for the execution of task $j \in J_n^B$ of community's member $n$ .
$z_{nh} \in \{0, 1\}$	is equal to 1 if and only if member $n$ 's BSS is charging at period $h$ .
$w_{nh} \in \{0, 1\}$	is equal to 1 if and only if the operation of member $n$ 's BSS changes from discharging or inactive to charging at period $h$ .
$T_{njkh} \geq 0$	represents the temperature in room $k$ of member $n \in \mathcal{N}$ reached by performing task $j \in J^A$ in period $h$ .
$q_{nh} \in \mathbb{R}$	is the amount of energy injected into/out member $n$ 's BSS at period $h$ , with $q_{nh} \leq 0$ if it discharges, $q_{nh} \geq 0$ if it is charging, and $q_{nh} = 0$ if inactive.
$E_{nh} \geq 0$	is the amount of electricity available in member $n$ 's BSS at period $h$ .
$F_{nh}^{\text{out}} \geq 0$	is the amount of energy injected by member $n$ at period $h$ .
$F_{nh}^{\text{in}} \geq 0$	is the amount of energy drawn from the community by $n$ at period $h$ .
$C_{nh} \geq 0$	is the amount of energy drawn from the grid by member $n$ at period $h$ .

Table 2: Each member's variables.

the remaining energy at period  $h$  is denoted  $I_h$ , which is injected into the main grid via the aggregator. Finally, the leader has variable  $\tau_{nh}$ , which designates member  $n$ 's contribution to  $I_h$  at period  $h$ .

Recall  $\Pi^L$  is the leader's objective function, which defines the total energy collected from the main grid. Minimizing function  $\Pi^L(x^L, x^F)$  is equivalent to maximizing the collective self-consumption rate. The leader's problem is:

$$\min_{x^L \in X^L(x^F)} \Pi^L(x^L, x^F), \quad (2)$$

where  $x^F$  represent followers' response to leader's decision, and  $x^L = (F^{\text{RCom}}, I, \tau)$ . More explicitly, the leader's problem is:

$$\min \sum_{n \in \mathcal{N}} \sum_{h \in \mathcal{H}} C_{nh} \quad (3a)$$

$$\text{s.t.} \quad \sum_{n \in \mathcal{N}} F_{nh}^{\text{RCom}} + I_h = \sum_{n \in \mathcal{N}} F_{nh}^{\text{out}} \quad h \in \mathcal{H} \quad (3b)$$

$$x^L \in \arg \max_{x^F \in X^F(x^L)} \sum_{n \in \mathcal{N}} \Pi_n^F(x^L, x^F) \quad (3c)$$

$$\tau_{nh} = g(I_h) \quad n \in \mathcal{N}, h \in \mathcal{H} \quad (3d)$$

$$F_{nh}^{\text{RCCom}}, I_h \geq 0, \tau_{nh} \in \mathbb{R} \quad n \in \mathcal{N}, h \in \mathcal{H}. \quad (3e)$$

where  $x^L$  and  $x^F$  are the leaders and followers' variables, and function  $\Pi_n^F(x^L, x^F)$  is member  $n$ 's objective function.

The leader shares the surplus between members via  $F^{\text{RCCom}}$  and sends the remaining energy  $I_h$  for  $h \in \mathcal{H}$  to the grid in constraint (3b) taking into account the followers' best response as shown in constraint (3c). As shown in Figure 1, the coordinator does not physically manage the energy. Indeed, the leader only ensures the coordination of the members' loads, sending an energy-based incentive. After sharing energy in the community, the leader determines the amount  $I$  that is exchanged with the aggregator and then the necessity of sharing the revenue associated with this injection among the prosumers. We model this revenue sharing by the leader's variable  $\tau_{nh}$ , which designates member  $n$ 's contribution to  $I_h$  at period  $h$ . Thus, variable  $\tau_{nh}$  depends on  $I_h$  for  $n \in \mathcal{N}, h \in \mathcal{H}$  as shown in constraint (3d).

### 3.2 The followers

Each follower,  $n$ 's objective is to determine her consumption, injection, and storage schedules to maximize her gain  $\Pi_n^F$ . Let  $x^F$  be the feasible region of the lower level's decision variable  $x^F$ . Thus, the lower-level problem is schematized by:

$$\max_{x^F \in X^F(x^L)} \Pi^F(x^L, x^F). \quad (4)$$

The detailed lower-level problem is presented in what follows. Each follower  $n$  has individual constraints related to the tasks to perform. These are listed below and commented on subsequently.

$$\sum_{s \in S_{nj}} x_{nj}^s = 1 \quad j \in J_n^B \quad (5a)$$

$$\forall j \in J^A, k \in K_{nj}^A, \quad \sum_{p \in P_{nj}^A} x_{njkh}^A \leq 1 \quad h \in \mathcal{H} \quad (5b)$$

$$T_{njkh} = \theta_j \left( \sum_{p \in P_{nj}^A} p \cdot x_{njkh}^A \right) \quad h \in \mathcal{H} \quad (5c)$$

$$t_{nj}^{\text{low}} \leq T_{njkh} \leq t_{nj}^{\text{up}} \quad h \in [h_{nj}^{\text{low}}, h_{nj}^{\text{up}}] \text{ if } \nu_{nj} = 1 \quad (5d)$$

$$T_{nj0} = \bar{T}_{nj}^{\text{all}} \quad \text{if } \nu_{nj} = 1. \quad (5e)$$

Recall that a type A task is performed to regulate the temperature in certain rooms (room or water heating), and a type B task must be performed under time window and consumption level constraints. Constraint (5a) refers to type B tasks. Indeed, exactly one schedule must be selected for each type B task. Constraints (5b)–(5e) are related to type A tasks. At each period, for each type A task, at most, one power level is selected (5b). Constraint (5c) ensures the temperature calculation at each period. Constraint (5d) imposes the temperature in the desired comfort zone, and (5e) sets the initial rooms' temperature. Another individual constraint concerns the BSS management constraint for  $n \in \mathcal{N}$ :

$$\Delta_h d_n \varrho_n (z_{nh} - 1) \leq q_{nh} \quad h \in \mathcal{H} \quad (6a)$$

$$q_{nh} \leq \Delta_h c_n c_n z_{nh} \quad h \in \mathcal{H} \quad (6b)$$

$$z_{nh} - z_{n(h-1)} \leq w_{n(h-1)} \quad h \in \mathcal{H} \setminus \{1\} \quad (6c)$$

$$\frac{E_{nh} - q_{n(h-1)}}{E_{n(h-1)}} = \eta_n \quad h \in \mathcal{H} \quad (6d)$$

$$\sum_{h \in \mathcal{H}} w_{nh} \leq \phi_n \quad (6e)$$

$$E_{nh} \leq \Gamma_n \quad h \in \mathcal{H} \quad (6f)$$

$$E_{n0} = \xi_n. \quad (6g)$$

Constraints (6a) and (6b) avoid the simultaneous BSS's charge and discharge. Variable  $q_{nh}$  is the amount injected if positive. Otherwise, the amount discharged from BSS. (6d) updates the BSS's state at each period, and (6e) limits the number of cycles during the planning horizon. Constraint (6f) enforces the amount of energy in the BSS to be less than its capacity, and (6g) sets the BSS's initial state.

Each member  $n$  maximizes her gain denoted by  $\Pi_n^F(x^L, x^F)$ . Constraint (7a) calculates member  $n$ 's gain. Notice that prosumers' gain depends on the amount injected in the grid; we denote  $\tau_{nh}$  as a parameter that designates member  $n$ 's contribution to this amount at period  $h$ .

$$\Pi_n^F(x^L, x^F) = \sum_{h \in \mathcal{H}} \left[ v_h^{\text{Com}} F_{nh}^{\text{out}} - \tilde{v}_h^{\text{Com}} F_{nh}^{\text{in}} + \tau_{nh} v_h^{\text{GES}} I_h - \tilde{v}_h^{\text{MG}} C_{nh} \right] \Delta_h \quad (7a)$$

In addition, each member  $n$  has some global operational constraints at  $h \in \mathcal{H}$  which are:

$$\sum_{j \in J^A} \sum_{k \in K_{nj}^A} \sum_{p \in P_{nj}^A} p \cdot x_{njkh}^A + \sum_{j \in J^B} \sum_{s \in S_{nj}} P_{njhs}^B x_{nj}^s + P_{nh}^C + \frac{q_{nh}}{\Delta_h} = P_{nh}^{\text{PV}} + C_{nh} + F_{nh}^{\text{in}} - F_{nh}^{\text{out}} \quad (8a)$$

$$F_{nh}^{\text{out}} \leq P_{nh}^{\text{PV}} + \frac{E_{nh}}{\Delta_h} \Omega_n \quad (8b)$$

$$F_{nh}^{\text{in}} + C_{nh} \leq \pi_n \quad (8c)$$

$$F_{nh}^{\text{in}} \leq F_{nh}^{\text{RCom}} \quad (8d)$$

$$F_{nh}^{\text{out}} \leq \pi_n. \quad (8e)$$

Constraint (8a) ensures energy conservation. A member can inject if she is a prosumer or a prosumer owning BSS (8b). Parameter  $\Omega_n$  equals one if member  $n$  can inject energy, i.e., if  $n$  is a prosumer or holds a BSS, 0 otherwise. Constraints (8c), and (8e) enforce each member's energy flows to be less than her subscribed power. Constraint (8d) imposes  $F_{nh}^{\text{RCom}}$  as the upper bound on  $F_{nh}^{\text{in}}$  for  $n \in \mathcal{N}, h \in \mathcal{H}$ .

Finally, we have the variables domains defined as follows: for  $n \in \mathcal{N}, j \in J^A, h \in \mathcal{H}, p \in P_{nj}^A, k \in K_{nj}^A, j' \in J^B, s \in S_{nj}$ ,

$$x_{njkh}^A \in \{0, 1\} \quad (9a)$$

$$x_{nj'}^s \in \{0, 1\} \quad (9b)$$

$$z_{nh}, w_{nh} \in \{0, 1\} \quad (9c)$$

$$F_{nh}^{\text{out}}, F_{nh}^{\text{in}}, C_{nh}, E_{nh} \geq 0, q_{nh} \in \mathbb{R}. \quad (9d)$$

Thus, each follower  $n$ 's problem is:

$$\begin{aligned} \max \quad & \Pi_n^F(x^L, x^F) \\ \text{s.t.} \quad & (5), (6), (7), (8), (9). \end{aligned} \quad (10a)$$

### 3.3 The full bilevel problem

Let us specify again the overall bilevel problem we are facing as:

$$\min_{x^L \in X^L(x^F), x^F \in X^F(x^L)} \Pi^L(x^L, x^F) \quad (11a)$$

$$\text{s.t.} \quad x^F \in \arg \max_{\tilde{x}^F \in X^F(x^L)} \Pi^F(x^L, \tilde{x}^F), \quad (11b)$$

where  $X^L(x^F) = \{x^L \in \mathbb{R}^{n^\ell} : g^L(x^L, x^F) \leq 0\}$  and  $X^F(x^L) = \{x^F \in \mathbb{R}^{n^1} \times \{0, 1\}^{n^2} : g^F(x^L, x^F) \leq 0\}$ ,  $n^\ell, n^1, n^2 \in \mathbb{N}$  and  $n^2 > 0$ . The follower's problem contains integer variables. Thus, the bi-level problem in this paper belongs to the class of mixed-integer linear bi-level programs (MILBPs), and classical approaches like KKT reformulation cannot be

used.

There are not so many algorithms available in the literature that can handle bilevel problems with integer variables in the follower problems. Table 3 presents those that we are aware of, where the column “largest instance size” provides the numbers of variables/constraints for the largest instances considered in each of these papers. We further present in Table 4 the dimensions of our problem in the more straightforward case where there are no type A and B tasks; notice that Smart Lou Quila contains to date  $N = 17$  members, including three BSS owners if the time horizon is sliced into periods of 30 minutes,  $H = 48$  for one day. Comparing the two tables, we see that only the algorithm of [6] has been tested on instances of similar dimensions. Unfortunately, a strong assumption of that algorithm is that there be no leader’s continuous variables in followers’ problem, as mentioned in the column “Assumption” of Table 3. Thus, their algorithm does not apply to our problem, and we further see in the table that each other algorithm also requires an assumption that our model violates.

Approach	Paper	Larger instance’s size	Assumptions	Conflict
Branch & bound	[16]	40/16	No coupling constraint in the upper level.	$\times$ (3b)
	[28]	920/184	All leader’s variables are required to be integral.	$\times$ (3e)
Branch & cut	[3]	14/16	No continuous variables in the leader’s problem.	$\times$ (3e)
	[2]	- <sup>1</sup>	Followers’ variables do not affect the leader’s constraints.	$\times$ (3b)
	[6]	80000/5000	No leader’s continuous variables in followers’ problem.	$\times$ (8d)
SBEA <sup>2</sup>	[14]	920/368	The leader variables are integer valued.	$\times$ (3e)

<sup>1</sup> No computational experiments.  
<sup>2</sup> a sampling-based exact algorithm

Table 3: Brief state of art.

Type	Leader	Followers	Total
Continuous variables	864	3456	4128
Binary variables	0	288	288
Constraints	144	5091	5235

Table 4: Problem’s size when no type A and B tasks exist.

In addition to this, we should note that even solving the followers’ problem can be difficult on its own, as we sometimes have issues reaching convergence. Therefore, it is not reasonable to hope for an exact algorithm for the overall bi-level problem. Instead, we present in the next section an ad-hoc math-heuristic algorithm adapted to the problem we face in this paper.

## 4 Iterative heuristic

Since the bilevel program (11) can hardly be solved exactly for the large instances, we introduce a heuristic algorithm. Each member determines local schedules and sends information about power requirements and availability to CC, the intermediary between members. Before presenting our distributed framework, we present the local scheduling model solved by each member. The local scheduling model for each member  $n \in \mathcal{N}$  is (10).

The members do not have information on the community's state in the distributed scheme. As a result, consumers do not know when to consume it or who sends the energy they receive. Thus, at each iteration, CC will provide the information  $F^{\text{RCom}}$  to incentivize the members to shift their consumption to period  $h \in \mathcal{H}$  where  $F_{nh}^{\text{RCom}}$  exceeds zero.

### 4.1 Allocation keys

Allocation keys are traditionally used as an ex-post means of distributing financial gains among EC members. When using ex-post, the allocation key is not an incentive since members cannot decide. In this paper, we use allocation keys for ex-ant scheduling. We use this notion to build an energy-based incentive to demand response. Indeed, in this paper, an allocation key is an algorithm that returns values that indicate to each member the maximum amount of energy she can collect from the community.

**Definition 1.** *An allocation key is an algorithm  $\mathbf{K}$  that, given  $C_{nh}$  and  $F^{\text{out}}$ , computes  $F_{nh}^{\text{RCom}}$  for  $n \in \mathcal{N}, h \in \mathcal{H}$ .*

The key allows CC to indicate the state of energy availability to each member without sharing private information with the members. Several ext-post keys exist in the literature [18], some currently used in the French electrical power system. Among the keys in the literature, we have the allocation in prorate to consumption, production, investment, etc. The ext-post key used in [Smart Lou Quila](#) inspires **K1**. We introduce a second key, denoted **K2**, that fits well with day-ahead scheduling.

- **Key K1.** Presented in Algorithm 3 for  $l = 1$ , this key shares the energy according to members' demand. Lines 7–10 in Algorithm 3 address the case where the total injection  $\sum_{n \in \mathcal{N}} F_{nh}^{\text{out}}$  exceeds the demand  $\sum_{n \in \mathcal{N}} C_{nh}$  for  $h \in \mathcal{H}$ . Each member  $n$  receives  $C_{nh}$  at period  $h$ . In lines 14–33, we consider the case where each member can receive the minimum energy required in the community  $C^{\text{min}}$  at a given period  $h$ , i.e., if  $N \cdot C^{\text{min}} \leq \text{residual}$ . In that case, we allocate  $C^{\text{min}}$  to members. We remove the satisfied members and share the *residual* energy among the remaining members. In

lines 31–33, the residual is insufficient to serve each member, i.e.,  $u \cdot C^{\min} \geq residual$ , where  $u$  is the number of unsatisfied members, so we share the *residual* uniformly among the remaining members. In lines 36–37, we handle cases where the energy available is insufficient to satisfy the total need and not sufficient to allocate  $C^{\min}$  to each member, so we share the energy available uniformly between members. Finally, in lines 40–42, value  $F_{nh}^{\text{RCom}}$  equals zero for each member  $n$  if the total injection equals zero at period  $h$ .

- **Key K2.** This key consists in determining the candidates for the energy reception and then sharing (uniformly) the power between these members. Member  $n$  is a candidate at period  $h$  if she does not inject energy, i.e., if  $F_{nh}^{\text{out}} = 0$ . After receiving all members' information, CC determines at each period who are the individuals likely to receive energy and shares the energy available uniformly among these members as detailed in Algorithm 1 where  $\sigma_{nh}$  is equal to 1 if member  $n \in \mathcal{N}$  is a candidate to receive energy from other members at period  $h \in \mathcal{H}$ , in other words, if  $F_{nh}^{\text{out}} = 0$ . Value  $\sigma_h^{\text{Tot}}$  is the number of candidates to receive energy from the community at period  $h \in \mathcal{H}$ .

---

**Algorithm 1: Key K2.**


---

**Input** :  $F^{\text{out}}$   
**Output:** initial  $F^{\text{RCom}}$   
 $\sigma_h^{\text{Tot}} = N$ ;  $\sigma_{nh} \leftarrow 1$  for  
 $n \in \mathcal{N}, h \in \mathcal{H}$   
**for**  $h \in H$  **do**  
    **for**  $n \in \mathcal{N}$  **do**  
        **if**  $F_{nh}^{\text{out}} > 0$  **then**  
             $\sigma_{nh} \leftarrow 0$   
             $\sigma_h^{\text{Tot}} \leftarrow \sigma_h^{\text{Tot}} - 1$   
        **for**  $n \in \mathcal{N}$  **do**  
             $F_{nh}^{\text{RCom}} \leftarrow \sigma_{nh} \sum_{m \in \mathcal{N}} F_{mh}^{\text{out}} / \sigma_h^{\text{Tot}}$

---



---

**Algorithm 2: Update K2.**


---

**Input** :  $F^{\text{in}}, F^{\text{out}}$  and  $F^{\text{RCom}}$   
**Output:** updated  $F^{\text{RCom}}$   
 $\sigma_h^{\text{Tot}} = N$ ;  $\sigma_{nh} \leftarrow 1$  for  
 $n \in \mathcal{N}, h \in \mathcal{H}$   
**for**  $h \in H$  **do**  
    **for**  $n \in \mathcal{N}$  **do**  
        **if**  $F_{nh}^{\text{RCom}} - F_{nh}^{\text{in}} > 0$  **then**  
             $\sigma_{nh} \leftarrow 0$ ;  $F_{nh}^{\text{RCom}} \leftarrow F_{nh}^{\text{in}}$   
             $\sigma_h^{\text{Tot}} \leftarrow \sigma_h^{\text{Tot}} - 1$   
    **for**  $n \in \mathcal{N}$  **do**  
         $F_{nh}^{\text{RCom}} \leftarrow$   
         $F_{nh}^{\text{RCom}} + \sigma_{nh} (F_h^{\text{out}} - F_h^{\text{in}}) / \sigma_h^{\text{Tot}}$

---

We set  $F_h^{\text{in}} = \sum_{n \in \mathcal{N}} F_{nh}^{\text{in}}$  and  $F_h^{\text{out}} = \sum_{n \in \mathcal{N}} F_{nh}^{\text{out}}$  for  $h \in \mathcal{H}$  for simplicity in Algorithm 2.

- **Key K3.** Key **K1** allocates energy even to members who inject energy since it is based on consumption, and there is no test on prosumers' injections. Thus, a prosumer can simultaneously receive and inject energy at certain periods. To avoid simultaneous injection and extraction, we combine **K1** and **K2** in a key called **K3**. Key **K3** is

presented in Algorithm 3 for input  $l = 3$ . We add some lines into **K1** that allow determining the candidates as defined previously. Thus, **K3** allocates energy according to consumption to members who need energy.

- Key **K4**. This key consists of sharing the surplus to promote consumption between needy members. Thus,

$$F_{nh}^{\text{RCom}} = \frac{\sigma_{nh} C_{nh}}{\sum_{m \in \mathcal{N}: \sigma_{mh} > 0} C_{mh}} \sum_{m \in \mathcal{N}} F_{mh}^{\text{out}},$$

where  $\sigma_{nh}$  equals one if member  $n$  is candidate to receive energy as defined in **K2**.

- Key **K5**. This key consists in sharing the surplus between the members who need energy in prorate to their investment in storage and generation tools. Thus, let us note  $\Sigma_n$  as member  $n$ 's total investment to acquire some tools,

$$F_{nh}^{\text{RCom}} = \frac{\sigma_{nh} \Sigma_n}{\sum_{m \in \mathcal{N}: \sigma_{mh} > 0} \Sigma_m} \sum_{m \in \mathcal{N}} F_{mh}^{\text{out}}.$$

Notice that **K1** and **K3** are fair in the sense of [8], i.e., the member who consumes the less is favored to the one who consumes the most while sharing the surplus. Further, by using **K2**, some members needing energy might receive more than required at certain times since the surplus is allocated uniformly, hence the need to update values  $F_{nh}^{\text{RCom}}$  at each iteration to ensure more efficient management using Algorithm 2. Indeed, CC must reallocate  $F_h^{\text{rest}}$ .

We update **K2** as presented in Algorithm 2. Member  $n$  will not receive more energy than  $F_{nh}^{\text{in}}$  at period  $h$  if  $F_{nh}^{\text{RCom}} - F_{nh}^{\text{in}} > 0$ . Let us denote  $F_h^{\text{out}} = \sum_{n \in \mathcal{N}} F_{nh}^{\text{out}}$   $h \in \mathcal{H}$  and  $F_h^{\text{in}} = \sum_{n \in \mathcal{N}} F_{nh}^{\text{in}}$   $h \in \mathcal{H}$  for each period  $h \in \mathcal{H}$ . Thus,  $F_h^{\text{rest}} = F_h^{\text{out}} - F_h^{\text{in}}$ , is the amount of remaining energy after each iteration using **K2**.

## 4.2 The overall algorithm

Each member initially solves the load scheduling MILP with  $F_{nh}^{\text{RCom}} = 0$  for  $n \in \mathcal{N}, h \in \mathcal{H}$ . That allows for determining the initial periodic energy needs and availability. Thus, the members send their surplus and need information to CC, who shares the surplus using the allocation key, sends values  $F_n^{\text{RCom}}$  to each member who updates the local schedules, and so on until the community reaches an acceptable level for the day's operations. Algorithm 4 presents the distributed management framework.

At the end of the second step, if there exists a period  $h$  where  $F_h^{\text{rest}} \neq 0$ , the amount

---

**Algorithm 3: key K1 and K3.**


---

```

Input      :  $C, F^{\text{out}}$  and  $l \in \{1, 3\}$ 
Output    :  $F^{\text{RCom}}$ 
1 if  $l = 3$  then  $\sigma_h^{\text{Tot}} = N$  for  $h \in \mathcal{H}$ ;    $\sigma_{nh} = 1$    for  $n \in \mathcal{N}, h \in \mathcal{H}$ 
2 for  $h \in \mathcal{H}$  do
3   if  $\sum_{n \in \mathcal{N}} F_{nh}^{\text{out}} > 0$  then
4     for  $n \in \mathcal{N}$  do
5       if  $F_{nh}^{\text{out}} > 0$  and  $l = 3$  then
6          $\sigma_{nh} \leftarrow 0$ ;    $\sigma_h^{\text{Tot}} \leftarrow \sigma_h^{\text{Tot}} - 1$ ;    $C_{nh} \leftarrow +\infty$ 
7     if  $\sum_{n \in \mathcal{N}} F_{nh}^{\text{out}} \geq \sum_{n \in \mathcal{N}} C_{nh}$  then
8       for  $n \in \mathcal{N}$  do
9         if  $l = 1$  then
10           $F_{nh}^{\text{RCom}} \leftarrow C_{nh}$ 
11        else
12           $F_{nh}^{\text{RCom}} \leftarrow C_{nh} \cdot \sigma_{nh}$ 
13      else
14         $C^{\min} \leftarrow \min_{n \in \mathcal{N}}(C_{nh})$ 
15        if  $l = 1$  then  $u \leftarrow N$  else  $u \leftarrow \sigma_h^{\text{Tot}}$ 
16        if  $u \cdot C^{\min} \leq \sum_{n \in \mathcal{N}} F_{nh}^{\text{out}}$  then
17          for  $n \in \mathcal{N}$  do
18            if  $l = 1$  then  $idx_n \leftarrow n$  else  $idx_n \leftarrow n \cdot \sigma_{nh}$ 
19           $residual \leftarrow \sum_{n \in \mathcal{N}} F_{nh}^{\text{out}}$ 
20          while  $u \cdot C^{\min} \leq residual$  do
21             $m \leftarrow 0$ 
22            for  $n \in \mathcal{N}$  do
23              if  $idx_n \neq 0$  then
24                 $F_{nh}^{\text{RCom}} \leftarrow F_{nh}^{\text{RCom}} + C^{\min}$ 
25              if  $C_{nh} = C^{\min}$  then
26                 $m \leftarrow m + 1$ ;    $idx_n \leftarrow 0$ ;    $C_{nh} \leftarrow +\infty$ 
27              else
28                 $C_{nh} \leftarrow +C_{nh} - C^{\min}$ 
29             $residual \leftarrow residual - u \cdot C^{\min}$ 
30             $u \leftarrow u - m$ ;    $C^{\min} \leftarrow \min_{n \in \mathcal{N}}(C_{nh})$ 
31          for  $n \in \mathcal{N}$  do
32            if  $idx_n \neq 0$  then
33               $F_{nh}^{\text{RCom}} \leftarrow F_{nh}^{\text{RCom}} + (residual/u)$ 
34          else
35            for  $n \in \mathcal{N}$  do
36              if  $l = 1$  then
37                 $F_{nh}^{\text{RCom}} \leftarrow (\sum_{n \in \mathcal{N}} F_{nh}^{\text{out}} / u)$ 
38              else
39                 $F_{nh}^{\text{RCom}} \leftarrow (\sum_{n \in \mathcal{N}} F_{nh}^{\text{out}} / u) \cdot \sigma_{nh}$ 
40      else
41        for  $n \in \mathcal{N}$  do
42           $F_{nh}^{\text{RCom}} \leftarrow 0$ 

```

---

---

**Algorithm 4:** Distributed management model.

---

**Input** :  $\mathbf{K} \in \{\mathbf{K1}, \mathbf{K2}, \mathbf{K3}, \mathbf{K4}, \mathbf{K5}\}$

**Output:**  $F^{\text{RCom}}$  and the local best schedules

**Step 0:**  $F_{nh}^{\text{RCom}} \leftarrow 0$  for  $n \in \mathcal{N}, h \in \mathcal{H}, \text{iter} \leftarrow 1$

Each member  $n$  solves (10) (in parallel) and sends  $(F_n^{\text{out}}, C)$  to CC;  $F_h^{\text{rest}} \leftarrow F_h^{\text{out}}$

**Step 1:** CC determines  $F^{\text{RCom}}$  using key  $\mathbf{K}$  and sends  $F_n^{\text{RCom}}$  to member  $n$  who solves (10) for  $n \in \mathcal{N}$

**if**  $\mathbf{K} = \mathbf{K2}$  **then**

**while**  $\text{iter} \leq \text{maxIter}$  **do**

        CC updates  $F^{\text{RCom}}$  using Algorithm 2 and sends  $F_n^{\text{RCom}}$  to member  $n \in \mathcal{N}$

        Each member  $n$  solves (10) and sends  $(F_n^{\text{out}}, F^{\text{in}})$  to CC,  $\text{iter} \leftarrow \text{iter} + 1$

**Step 2:** Manage  $F^{\text{rest}}$  corresponding to variable  $I$ , determine  $\tau$  for gain sharing.

---

$F_h^{\text{rest}}$  is injected into the grid, and the economic gain associated with  $F^{\text{rest}}$  is shared between the prosumers, **Step 2** refers to that in Algorithm 4. In this paper, we do not address the gain-sharing aspect.

Observe that, for  $\mathbf{K} = \mathbf{K2}$ , the members' local solutions do not change after at most  $\text{maxIter} = N + 1$  if each local problem is optimally solved at each iteration. Indeed, at each iteration, a group of  $l \in \{1, \dots, N\}$  members are satisfied and removed from the set of candidates to receive energy all over the horizon. These members will not receive more energy over the horizon: their solutions stay the same, and  $F_h^{\text{rest}}$  is allocated to the candidates  $h \in \mathcal{H}$  who optimally solve their problem. The worst case occurs if  $l = 1$  at each iteration; hence,  $\text{maxIter} = N + 1$  iterations with the initial iteration.

**Remark 1.** *As each member solves the optimization problem (10) after receiving the leader's incentive, the heuristic's solution is feasible for the bilevel formulation (11).*

### 4.3 The remaining energy and storage management

BSS owners should be encouraged to store energy at certain periods. Indeed, as prosumers maximize their profit, they can inject more energy than is required to satisfy the global energy need. In the distributed setting, the prosumers cannot foresee the portion of their surplus that is locally consumed. However, CC can provide storage incentives to encourage BSS owners to store. The storage incentive must prioritize local consumption; hence, there is interest in setting limits and an upper bound on BSS owners' storage.

Let us introduce  $\rho_{nh}$  as the value that limits member  $n \in \mathcal{N}$  injection at period  $h \in \mathcal{H}$ . We assume that  $\rho_{nh}$  is proportional to quantity depending on member  $n \in \mathcal{N}$ 's physical

characteristic in terms of asset owing plus a variable a quantity depending on the member  $n$ 's participation in the global production, which multiplies the global need.

$$\rho_{nh} = \delta_n + \frac{P_{nh}^{\text{PV}}}{\sum_{m \in \mathcal{N}} P_{mh}^{\text{PV}}} C_h \quad h \in \mathcal{H}. \quad (12)$$

We set  $\rho_{nh}$  for each  $n \in \mathcal{N}, h \in \mathcal{H}$  such that it is proportional to the energy need and inversely proportional to the community's generation as presented in constraint (12), where  $\delta_n$  is an ad-hoc constant determined by trial and error. This makes sense since we assume that the self-discharge rate of the batteries is low. We add constraints (13) to each member  $n$ 's model (10).

$$F_{nh}^{\text{out}} \leq \rho_{nh} \zeta_n + \pi_n (1 - \zeta_n) \quad h \in \mathcal{H}, \quad (13)$$

where  $\zeta_n$  is equal to 1 if member  $n \in \mathcal{N}$  owns a storage unit, 0 otherwise and  $\delta_n$  a physical parameter related to member  $n$ 's generation and storage systems' size. Figure 5 presents the periodic  $F^{\text{rest}}$  and  $Need$  before and after adding constraints (13).

## 5 Numerical experiments

We use a realistic instance based on [Smart Lou Quila](#)'s demonstrator. The planning horizon consists of a day sliced into 24 equal-length periods. The first six members' real asset-owing characteristics are presented in Table 5. The last member does not have any assets. Members subscribed power are respectively  $\pi = (36, 9, 6, 9, 9, 6, 9)$  kVa.

We use the energy generation data collected on July 23<sup>th</sup> 2022. Figure 2 presents each prosumer's generation and the community's total periodic production. The members' loads are artificially generated. In other words, we generate loads of the different types: A, B, and C. The following array  $\nu$  shows each member's requested type A tasks. We consider two different type A tasks, i.e.,  $|J^{\text{A}}| = 2$  for  $n \in \mathcal{N}; j \in J^{\text{A}}, j = 1$  means the corresponding task is room temperature regulation and  $j = 2$  means that the requested task is water heating. Finally,  $\nu_{njk}$  is equal to 1 if task  $j \in J^{\text{A}}$  is executed by member  $n$  in room  $k$ , 0 otherwise. We refer to [19] for more details:

Member 1		Member 2	
Photovoltaic (PV)	Yes	PV	Yes
PV Capacity	6.12kWp	PV Capacity	3.2kWp
Battery	Yes	Battery	Yes
Number	1 unit	Number	1 unit
Capacity	9.8kWh	Capacity	9.8kWh
Initial state of charge	4.5kWh	Initial state of charge	4.5kWh
Efficiency	97.5%	Efficiency	97.5%
Power	5kW	Power	3.7kW
Periodic discharge rate	1%	Periodic discharge rate	1%
Member 3		Member 4	
PV	No	PV	Yes
PV Capacity	0kWp	PV Capacity	3.2kWp
Battery	yes	Battery	No
Number	1 unit	Number	0 unit
Capacity	9.8kWh	Capacity	0kWh
Initial state of charge	4.5kWh	Initial state of charge	0kWh
Efficiency	97.5%	Efficiency	0%
Power	3.7kW	Power	0kW
Periodic discharge rate	1%	Periodic discharge rate	0%
Member 5		Member 6	
PV	Yes	PV	Yes
PV Capacity	3.2kWp	PV Capacity	3.2kWp
Battery	No	Battery	No

Table 5: Production and storage assets description in the community.

$$\nu_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \quad \nu_2 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix} \quad \nu_3 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix} .$$

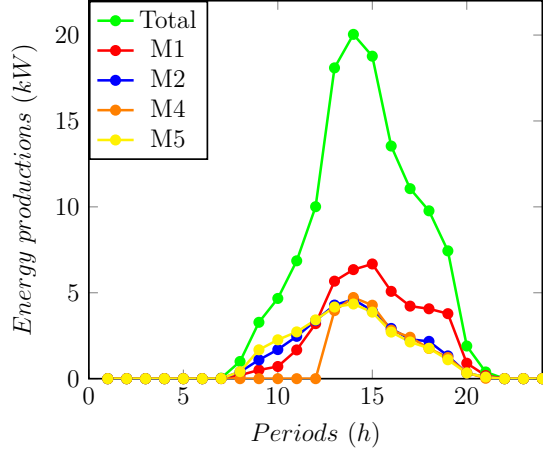


Figure 2: Generation on July 23<sup>th</sup> 2022.

Table 6 presents members 2 to 7 houseroom characteristics. For simplicity, we assume that each room  $k$  of each member has an initial temperature equal to  $22^{\circ}\text{C}$  and that the temperature of these rooms must lie in the interval  $[t_{nk}^{\text{low}}, t_{nk}^{\text{up}}]$  during the time interval  $[34, 48]$ .

$$t^{\text{low}} = \begin{pmatrix} 20 & 19 & 20 & 20 & 21 & 20 & 19 \\ 20 & 19 & 20 & 20 & 20 & 20 & 19 \\ 20 & 19 & 20 & 20 & 20 & 20 & 19 \end{pmatrix} t^{\text{up}} = \begin{pmatrix} 25 & 25 & 25 & 25 & 25 & 25 & 25 \\ 25 & 25 & 25 & 25 & 25 & 25 & 25 \\ 25 & 25 & 25 & 25 & 25 & 25 & 25 \end{pmatrix}$$

Each member has a single water heater, and Table 7 presents their physical characteristics. Parameter  $c_p = 1$  is the specific heat capacity of water. Column  $M$  shows the weight of the water (kg), column  $S$  designates the exchange surface of the water container with the external area,  $K$  is the exchange coefficient ( $\text{kcal}/\text{h}\cdot\text{m}^2\cdot^{\circ}\text{C}$ ). Finally, column  $\bar{T}^{\text{water}}$  presents the initial temperature of each member's water heater. The members' water must reach  $60^{\circ}\text{C}$  at period 16. The efficiency of the electricity-to-heat transformation  $v = Mc_p / (Mc_p r + \Delta SK r)$ . Also, for simplicity, we assume that  $p_{nh}^c$  which is member  $n \in \mathcal{N}$  non-controllable consumption at period  $h$ , is equal to one for each member at each period.

Notice we do not address the pricing in this work, i.e., we do not consider how prices  $\tilde{v}_h^{\text{com}}$  and  $v_h^{\text{com}}$  for  $h \in \mathcal{H}$  are set. CC determines these prices for the Smart Lou Quila scheme. The aim is to maximize local consumption of locally produced energy correlated to CC's margin. Therefore, the exchange prices in the community must be more attractive, i.e., the selling in the community must be higher than the price at which the aggregator takes the excess, and the buying prices in the community must be lower than the buying price from MG. Without

Member	room	Surface	$C_r$	$U$
Member 2	room 1	$9m^2$	297	12
	room 2	$15m^2$	495	20
	room 3	$18m^2$	594	24
Member 3	room 1	$18m^2$	594	24
	room 2	$9m^2$	297	12
	room 3	$9m^2$	297	12
Member 4	room 1	$12m^2$	396	16
	room 2	$20m^2$	660	26.6
	room 3	$12m^2$	396	16
Member 5	room 1	$25m^2$	825	33.3
	room 2	$10m^2$	330	13.3
	room 3	$12m^2$	396	16
Member 6	room 1	$18.5m^2$	610.5	24.6
	room 2	$9m^2$	297	12
	room 3	$10m^2$	330	13.3
Member 7	room 1	$15m^2$	495	20
	room 2	$10m^2$	330	13.3
	room 3	$14m^2$	462	18.6

Table 6: Rooms physical characteristics.

Member	S	M	$\bar{T}^{\text{water}}$	K	r
1	7	350	15	1	1.2
2	1.5	75	17	1	1.2
3	2	100	18	1	1.2
4	3.75	200	19	1	1.2
5	2.4	150	10	1	1.2
6	2	100	10	1	1.2
7	2.6	150	8	1	1.2

Table 7: Water heaters characteristics.

loss of generality, we use the following prices in euros per kWh given by Smart Lou Quila's

coordinator for each  $h \in \mathcal{H}$ :

$$\begin{aligned}
 v_h^{\text{MG}} = v_h^{\text{GES}} &= 0.1685 \\
 \tilde{v}_h^{\text{GES}} &= 0.065 \\
 v_h^{\text{Com}} &= 0.1400 \\
 \tilde{v}_h^{\text{Com}} &= 0.12,
 \end{aligned}$$

where  $\tilde{v}^{\text{GES}}$  is the price of selling to the aggregator. These prices must also be sufficient to amortize the marginal cost of using the BSS if we limit the injections.

## 5.1 Keys comparison

This section presents the results on a realistic instance to illustrate the demand response. First, we compare the allocation mechanisms and show the need for CC to control storage. Then, we show the impact of these controls on the distributed solution. Table 8 presents the distributed solution for each allocation key.

We do not update **K2**. In this table, Column **Avail** is the amount of photovoltaic energy that can be shared in the community or injected into the grid, and column **Loss** is part of **Avail** that needs to be injected into the grid.

Key	Distributed solutions			
	obj kWh	CPU	Available kWh	Loss kWh
K1	125.62	0.20	27.97	13.78
K2	115.50	0.20	27.97	3.65
K3	117.96	0.20	27.97	6.11
K4	115.50	0.25	27.97	3.65
K5	119.30	0.25	27.97	7.44

Table 8: Key comparison.

Table 8 shows that the distributed management with keys **K2** and **K4** returns the best solutions, i.e., ensures more local consumption of green energy than the other keys. Since initial consumption is determined without information, consumers consume randomly to meet their initial optimum. Therefore, a consumption-based sharing mechanism will not incentivize these members to shift the load. Furthermore, sharing based on consumption leads to allocating energy to members who do not need it. Figure 3 shows the different keys'

consumption shifts during the planning horizon after  $N + 1$  iterations for key **K2** and two iterations for **K1** and **K3**.

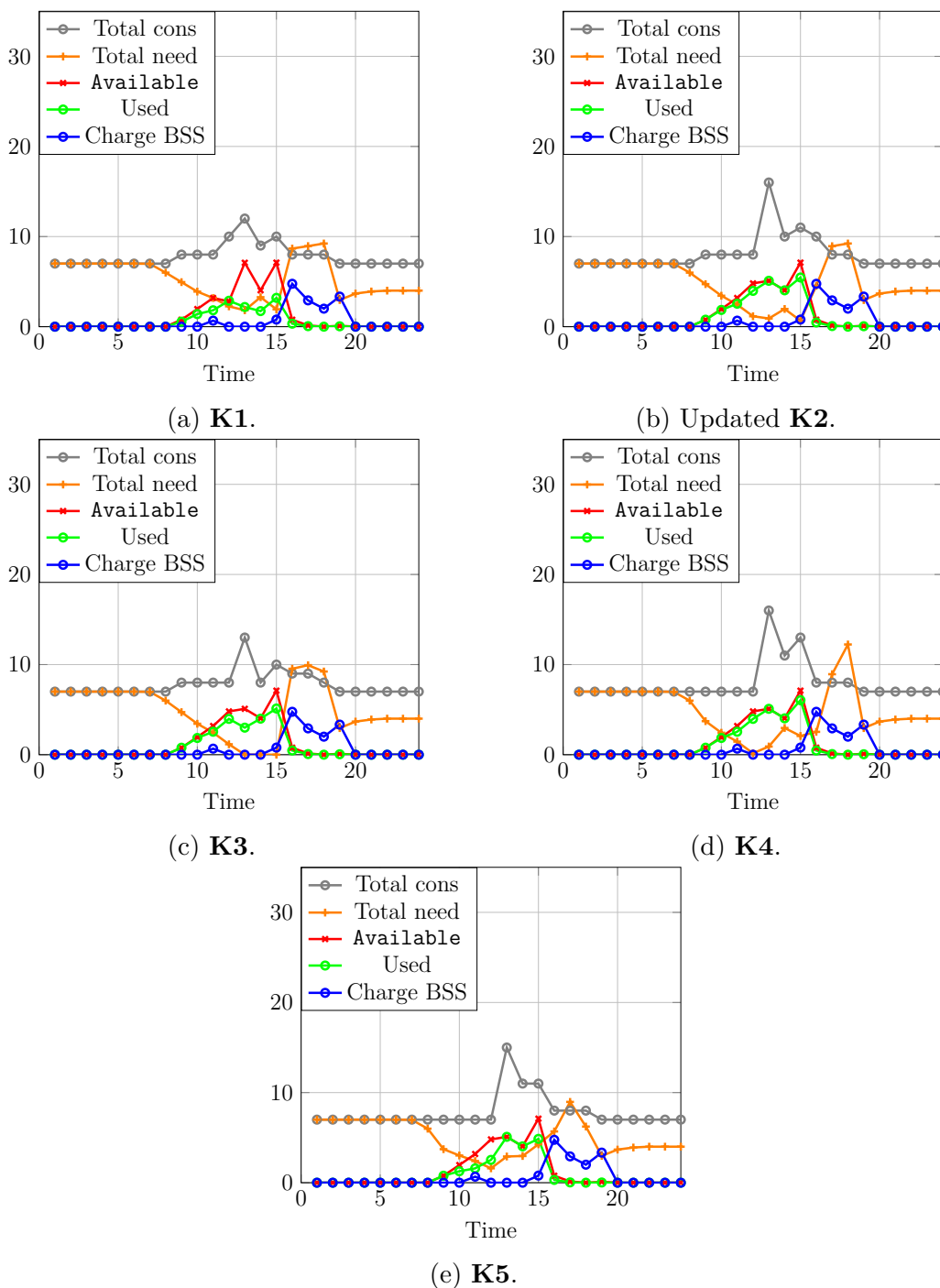


Figure 3: Demand response illustration.

Curves Available and Used represent respectively the amount of green energy injected

by prosumers and the amount of green energy available that is locally consumed. **Total need** represents the total energy needed in the community for direct consumption or to be stored for ulterior consumption. The area between curves Available and Used is the energy injected into the main grid via the aggregator. It corresponds to column **Loss** in Table 8. Allocating energy to members who need it reduces the area between available and Used curves, leading to a good distributed solution.

Table 9 presents the solution instance using updated **K2**. Iteration 1 in column 1 corresponds to the case where  $F^{\text{RCom}} = 0$ . Each column shows each member’s total energy collected from the grid during the planning horizon, one row per member. Row **Sum** presents the energy collection for the whole community during the planning horizon.

	1	2	3	4	5	6	7	8
1	8.71	<b>8.13</b>	8.13	8.13	8.13	8.13	8.13	8.13
2	<b>7.6</b>	7.6	7.6	7.6	7.6	7.6	7.6	7.6
3	12	9.86	9.69	9.64	9.64	<b>9.63</b>	9.63	9.63
4	13.52	<b>12.39</b>	12.39	12.39	12.39	12.39	12.39	12.39
5	34	26.84	26.21	25.75	25.66	25.58	25.56	<b>25.54</b>
6	31	23.84	23.21	22.75	22.66	22.58	22.56	<b>22.54</b>
7	34	26.84	26.27	26.09	26.06	26.04	<b>26.03</b>	26.03
<b>Sum</b>	139.83	115.5	113.5	112.35	112.14	111.94	111.9	<b>111.86</b>

Table 9: Distributed management solution using updated **K2**.

Updating keys leads to better allocation. Consequently, consumers shift more loads at available energy periods. Consequently, we obtain a high-quality distributed solution since the objective value of the centralized optimum solution is 111.16 kWh. Furthermore, the solution is bi-level feasible since the community’s problem is optimally solved at each iteration.

However, there is a need to incentivize storage if the energy to be shared is important. Indeed, since the prosumers maximize their profit, they inject more at periods where the prices lead to a maximum profit. This results in too much injection some times.

For example, if we increase the total production by 50%, the distributed solution leads to a solution with 79.34 kWh collected from the main grid versus 64.02 kWh, the optimal centralized solution’s objective value, which is an increase of 24%. The remaining energy  $F^{\text{rest}}$  is presented by Figure 4. We notice a lot of green energy available in periods 10 to 17 while no energy is needed in the whole community (the opposite situation occurs after period

17). That is due to the lack of global perception of the community’s state for prosumers. CC could have incentive members owning batteries to store energy during periods 10 to 17 for ulterior use.

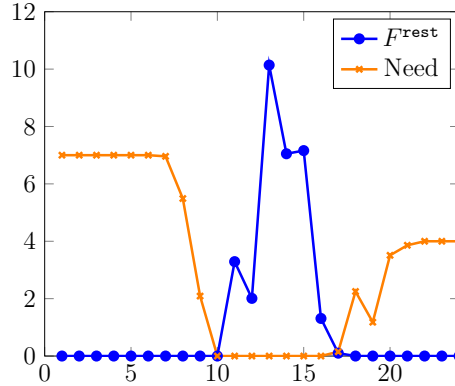


Figure 4: Storage requirement illustration.

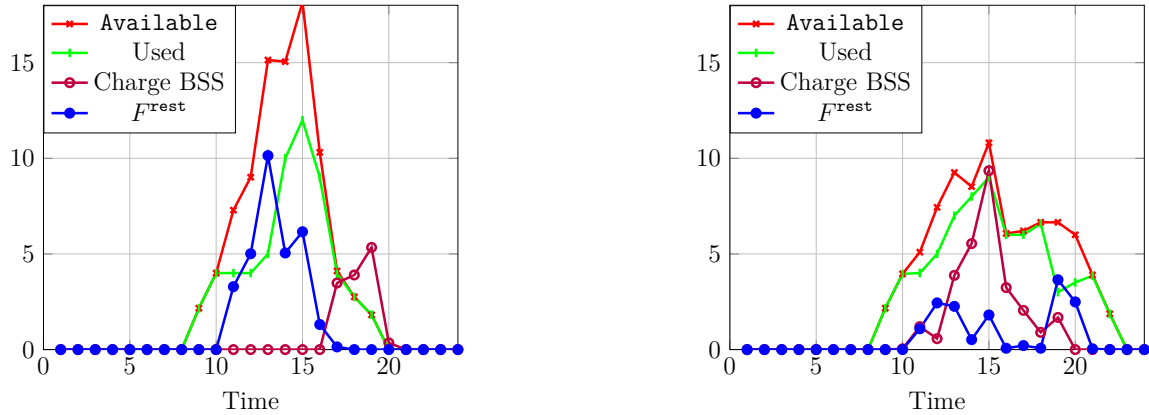
## 5.2 Storage control

As shown in Figure 4, there is a need to incentivize BSS owners to store at certain periods. We express this incentive as an upper bound on these members’ periodic injections. We introduce constraint (13) to achieve it. Thus, CC calculates  $\rho_{nh}$  for  $n \in \mathcal{N}, h \in \mathcal{H}$  at each iteration. In this paper, we set  $\delta_n$  proportional to the size of member  $n \in \mathcal{N}$  compared to the other generating members who own the BSS. Each member  $n$  owing a BSS has an initial  $\delta_n = 1$  kWh. Suppose member  $m$  produces twice as much as the other members,  $\delta_m = 2$  kWh. Three members own the BSS, namely members 1, 2, and 3, according to the previous affirmation  $\delta = (2, 1, 1, 0, 0, 0)$ . We set each member’s local scheduling time limit to 50 seconds at each iteration.

Limiting prosumers owing BSS makes these members stock to reinject at other times, allowing better solutions, as displayed in Figure 5b.

## 6 Case study

In this section, we experiment with accurate data and show how our model can be used for long- and medium-term decision-making and operational community management. Indeed, in section 6.1, we present the solutions of instances built over a whole month, while in



(a) Without storage control: objective is 79.37 kWh, energy injected into MG is 31.10 kWh.

(b) With storage control: objective 68.13 kWh, energy injected into MG is 14.65 kWh.

Figure 5: Storage control’s impact.

section 6.2, we assume that forecast data is available on day  $d + 1$ . In both cases, we present and analyze the solutions.

## 6.1 Analysis based on historical data

This section presents the results of accurate historical data from Smart Lou Quila, illustrating the benefit of storing energy for further use in the community. Since we do not know the actual load classification, we assume each member’s loads belong to the type C, i.e., no consumption modulation is expected.

We compare two types of management approaches. The first one corresponds to the actual management used at Smart Lou Quila, which we call “NoOpt” to reflect that this approach does not optimize the use of the member’s storage units. NoOpt involves solving the community’s problem with Algorithm 4 in which each follower solves problem (10) with all scheduling variables fixed except  $C_{nh}$ ,  $F_{nh}^{out}$ , and  $F_{nh}^{in}$  for  $n \in \mathcal{N}, h \in \mathcal{H}$ . In addition, the leader’s allocation follows key **K1**. The second approach, called “Opt” in what follows, allows storage planning among the different members. It involves solving the management problem with Algorithm 4 in which each follower solves (10) with no scheduling variables fixed. In Opt, we compare the allocation rules **K1** and **K2**.

We collect the data on a monthly basis for each instance. Specifically, we build an instance for each month from January to September 2023. The data are collected every 30 minutes over the month. Figure 8a illustrates the loads over May 2023. In addition, we have three BSS owners, as shown in Table 5. This paper assumes a storage unit’s state can change at

least twice daily. Consequently, the parameter  $\phi$  representing the maximum cycle allowed by the storage unit is set to  $2 \cdot (H/48)$  for each BSS owner in Opt;  $H/48$  is the number of days for a given month.

The data and models presented in the paper are available at <https://github.com/MsangL/distributedManagement>.

We cannot compare our solutions to bi-level solutions as we do not solve the bilevel problem. However, given the leader’s objective, which is to minimize energy collection from the main grid, we know that a bi-level solution cannot be better than a centralized solution. Thus, we compare our distributed solutions to the optimal centralized solutions.

**Remark 2.** *A centralized solution is where the coordinator solves a single MILP as presented in [19] with all members’ variables and constraints, and peer-to-peer exchanges are allowed.*

Figure 6 presents the objective function for distributed and centralized management approaches. We compute the centralized solution by solving MILP with `time_limit = tl = 3600s`. The distributed management heuristic returns high-quality solutions compared to the centralized solutions. Table 10 presents more detail on the solutions; in this table, columns Month and  $N$  present the time horizon and the community’s size. Notice that Smart Lou Quila contains five members who generate energy in 2023. In the same line, column  $P^{PV}$  and Demand present respectively the total renewable generation and energy consumption over each horizon; Figure 8a presents the generation and consumption profiles over May 2023. Values Obj give the total amount of energy collected from the grid over the horizon. Finally, column GAP presents the gap between the best-distributed value and the centralized solution with storage units.

				Opt						NoOpt	
				Centralized solutions			Distributed solutions			Distributed solution	
Month	$N$	$P^{PV}$ (kW)	Demand(kW)	Obj(kWh)	gap(%)	CPU	K1	K2	GAP%	K1	GAP%
Jan	8	397	33016	16310	0	2	16315	<b>16310</b>	0	16310	0
Feb	8	752	26989	13118	0	2	13150	<b>13122</b>	0.03	13155	0.28
Mar	8	1493	33016	15762	0	2	15792	<b>15772</b>	0.06	15811	0.31
Avr	8	2354	11522	4626	0	2	4836	<b>4722</b>	2.10	5029	8.01
May	8	2363	8795	3274	0	2	3510	<b>3415</b>	4.30	3741	14.26
Jun	13	1944	12997	5529	0	71	5613	<b>5578</b>	0.89	5660	2.37
Jul	17	2017	16027	7008	0	10	7079	<b>7047</b>	0.56	7117	1.56
Aug	17	1781	16094	7160	0	96	7224	<b>7197</b>	0.52	7273	1.58
Sep	17	1581	15143	6782	0	11	6836	<b>6815</b>	0.49	6864	1.19

Table 10: Solutions comparison (period length: 0.5h).

Our solutions are feasible in the sense of bi-level programming, as all followers' problems are optimally solved at each iteration for each key. Indeed, the CPU times to obtain the solutions while using the key **K2** are respectively: 5, 4, 5, 23, 28, 66, 44, 22, and 58. Also, the solving process can be parallelized per member at each iteration since the members' problems are separated, boosting the solving process.

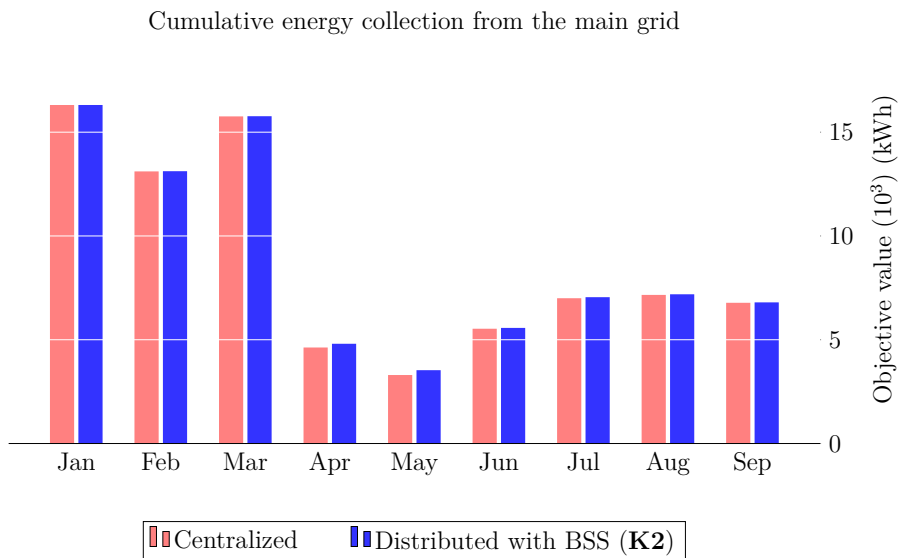


Figure 6: Distributed management solution's quality for Opt.

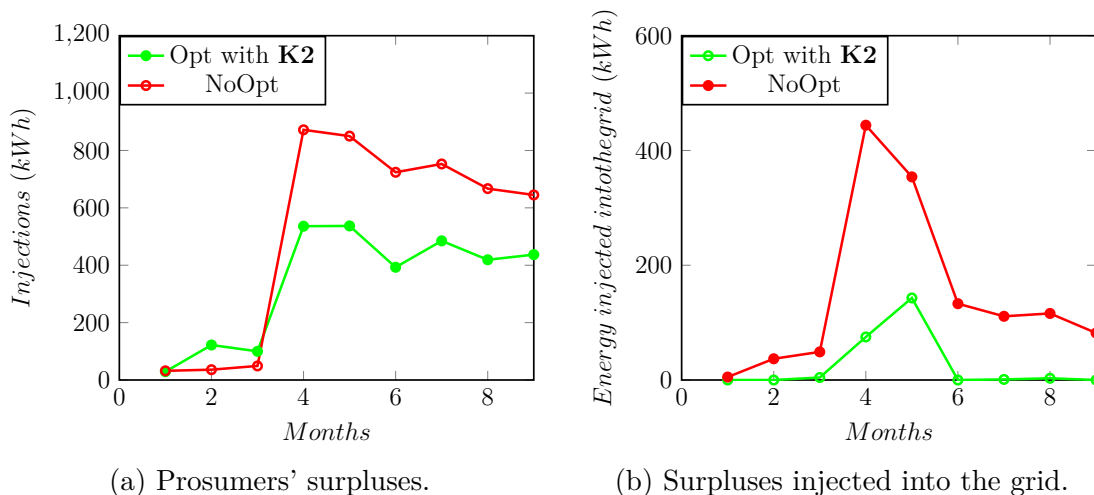
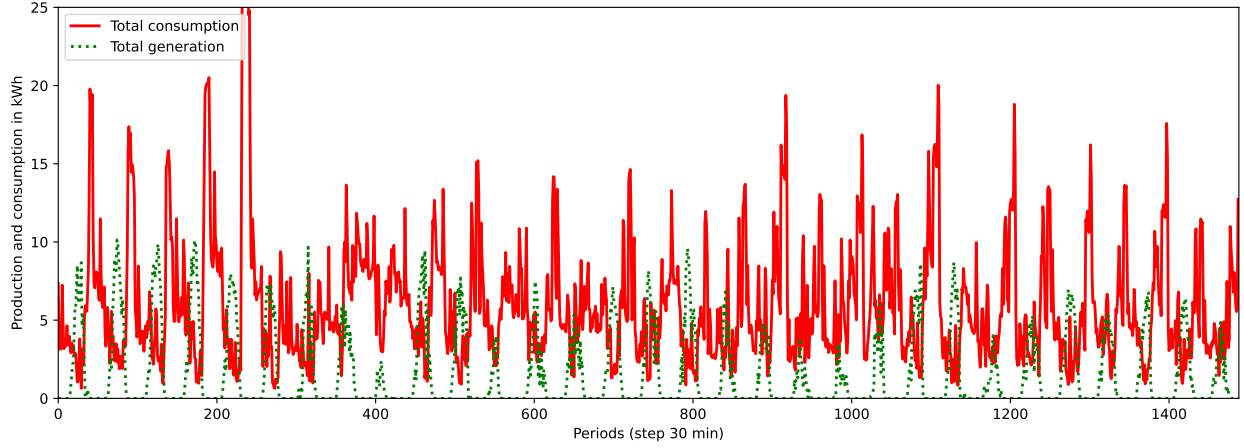
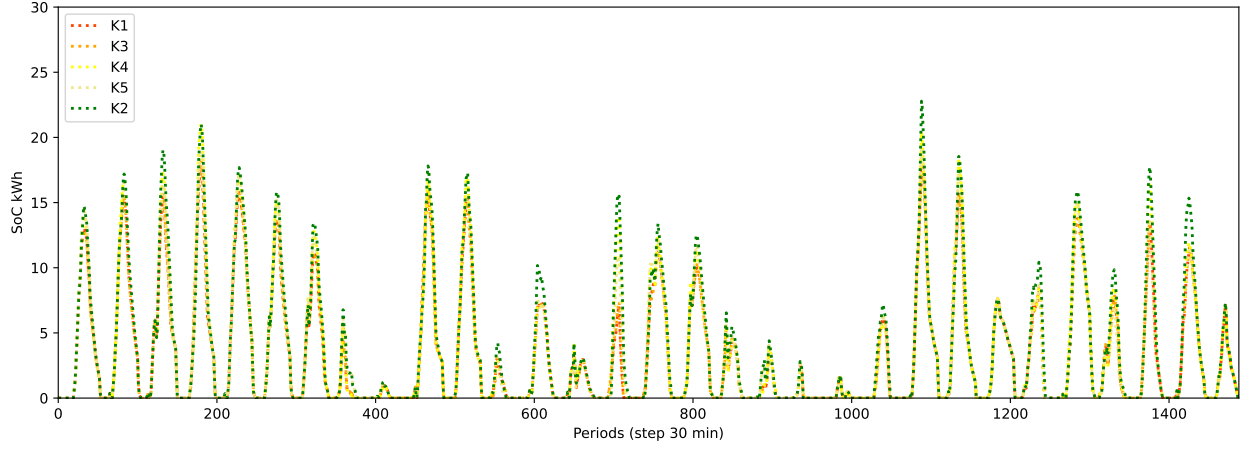


Figure 7: Opt vs NoOpt.

From Table 10, **K2** returns the best solutions in Opt because it allows storage for ulterior usage. Indeed, **K2** shares uniformly, if a member  $n$  owning a BSS receives at period  $h$  an



(a) Inputs.



(b) Total BSSs' level.

Figure 8: Instance for may 2023.

amount  $F_{nh}^{\text{RCom}}$  greater than her consumption, i.e.,  $F_{nh}^{\text{RCom}} \geq C_{nh}$ , she can store the excess for her ulterior usage; the sum of the periodic state of charge over May 2023 per key is: (4874, 5589, 4894, 5229, 5150) kWh.

Experience has shown that, unlike the conventional approach (NoOpt), Opt can effectively establish DR in a community. It also helps in maximizing the collective self-consumption rate. Indeed, 7.46% of prosumers' surpluses are injected into the grid in Opt against 29% in 2023 using NoOpt (see Figure 7). Further, the collective self-consumption rate goes from 80% in NoOpt to 97% in Opt.

## 6.2 Day ahead planning with forecast data

In this section, we assume the data correspond to reliable generation and loads for members on day  $d$ , and the coordinator performs day-ahead planning for day  $d + 1$ .

In this case, our model can be used for the day ahead planning. To assess the solutions in this operational case, we solve the community’s problem for each day from January to September 2023 using key **K2**. Regarding the storage units, we set the maximum number of cycles per storage unit to  $\phi = 2$ .

Table 12 presents the energy collected from the main grid for each day in May 2023; in this table, columns “Per month Obj” and “Per day total Obj” present, respectively, the solution while solving the instance for one month and the total energy collected from the grid obtained by solving the problem for each day. Column “Total CPU” presents the total CPU time to solve the problem for each day. Finally, column “gap” presents the gap in percentage between both solution approaches.

Table 11 presents the solutions for the other months. The average gap between the best monthly distributed solutions and the day-ahead solutions is 0.054%, and the average CPU time to solve a member’s problem at each iteration is 0.21s.

Month	N	Per month Obj (kWh)	Per day total Obj (kWh)	Total CPU	gap(%)
Jan	8	16310	16310	3.13	0
Feb	8	13122	13123	2.21	0.008
Mar	8	15772	15278	1.44	0.04
Avr	8	4722	4709	1.50	-0.28
May	8	3415	3440	1.50	0.73
Jun	13	5578	5579	3.94	0.018
Jul	17	7047	7050	3.91	0.043
Aug	17	7197	7195	3.75	-0.028
Sep	17	6815	6812	2.50	-0.044

Table 11: Day ahead solutions per month.

The following section discusses how the member’s privacy is preserved in the model and the pricing in the considered community.

## 6.3 Privacy and security

In this paper, we introduce an optimization model that includes a coordinator. The coordinator directs the loads to achieve a specific objective: maximizing the collective self-consumption rate. Our partner, [SWEEN](#), is the coordinator of Smart Lou Quila.

Day	Obj (kWh)	CPU	Day	Obj (kWh)	CPU
1	104.43	0.15	17	78.06	0.23
2	116.49	0.18	18	83.39	0.63
3	97.83	0.59	19	123.72	0.46
4	127.94	0.21	20	138.63	0.48
5	193.16	0.20	21	130.95	0.24
6	107.37	0.21	22	153.13	0.21
7	63.33	0.15	23	106.37	0.17
8	125.36	0.22	24	114.85	0.22
9	170.23	0.24	25	107.9	0.22
10	95.48	0.22	26	110.28	0.54
11	87.41	0.26	27	68.45	0.26
12	134.1	0.21	28	109.34	0.22
13	95.32	0.34	29	107.95	0.20
14	117.92	0.22	30	100.01	0.28
15	84.68	0.18	31	108.33	0.22
16	77.58	0.26			

Table 12: Day ahead solutions for May 2023.

The prices  $v^{\text{com}}$  and  $\tilde{v}^{\text{com}}$  used in the community are fixed and known to all members. Therefore, maximizing the collective self-consumption rate involves the coordinator encouraging members to consume when excess energy is available. This way, the coordinator promotes demand response by offering energy-based incentives. These incentives are determined solely based on members’ net consumption and generation data; no other information is shared with the coordinator. Additionally, no information is exchanged with any other member. Further, each member is 100% free to follow the coordinator’s incentive or not.

Regarding reliability, the system works if communication between the coordinator and a group of members is interrupted. The worst-case scenario is where the coordinator is 100% faulty. In this case, we can do a posteriori management without storage and Demand response (NoOpt) on data collected from the distribution system operator (Enedis in our case); this method ensures a collective self-consumption rate of 80% for SLQ in 2023.

## 6.4 Fainess

There are several conceptions of fairness in the literature; in Smart Lou Quila, fairness prioritizes the member who consumes the least while sharing the surplus; see Algorithm 3. Key **K2** does not explicitly prioritize the members who consume less. However, as **K2** shares uniformly, the one needing the least is necessarily satisfied first if there is enough energy to

share, as shown in Figure 9. In Figure 9, at iterations 1 and 2, both members receive the same amount, which satisfies member 1; the remaining energy is allocated to member 2 at iteration 3. The local energy available is insufficient to satisfy Member 2. Thus, the gap is collected from the main grid (red area). The members receive the same amount if they have the same need.

**Lemme 1.** *The process is secure and respects privacy.*

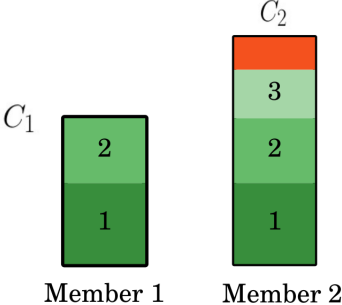


Figure 9: Fairness.

Fairness at Sween consists in prioritizing the member who consumes the least when sharing the surpluses. Thus, the keys **K2**, **K4**, **K5** are not fair.

## 7 Conclusion

In this paper, we present a single-leader, multi-follower, bi-level program. However, the lower-level problem is a MILP. Thus, we present a distributed management heuristic method to solve the bilevel problem for large problem instances. To do so, we propose a heuristic based on an efficient egalitarian allocation key for ex-ant community management, i.e., while considering storage units. Our heuristic returns high-quality solutions compared to optimal centralized solutions on real-world data; the average gap between these solutions is 1%. The solutions are bi-level feasible as each lower-level model is solved optimally. In addition, we improve the collective self-consumption compared to the actual management implemented in Smart Lou Quila, going from 80% to 97%

Some critical perspectives of this work are electric mobility and grid services. Concerning mobility, for example, we would like to charge the community’s electric vehicles with the quantities injected into the grid, considering the travel schedule. As for grid services, we can imagine a form of collaboration between CC and the main grid, which would enable the

community to supply neighboring households or communities, depending on the state of the overall grid. Finally, the system includes uncertainty at several levels; we have handled the deterministic case for simplicity's sake. Future work may consider them.

## Declaration

Conflict of interest: The authors declare that they have no financial interests or personal relationships that could have influenced the work presented in this article.

## References

- [1] S. Conti, R. Nicolosi, S. A. Rizzo, and H. H. Zeineldin. Optimal dispatching of distributed generators and storage systems for mv islanded microgrids. *IEEE Transactions on Power Delivery*, 27(3):1243–1251, 2012.
- [2] Stephan Dempe and F Mefo Kue. Solving discrete linear bilevel optimization problems using the optimal value reformulation. *Journal of Global Optimization*, 68:255–277, 2017.
- [3] Scott T DeNegre and Ted K Ralphs. A branch-and-cut algorithm for integer bilevel linear programs. In *Operations research and cyber-infrastructure*, pages 65–78. Springer, 2009.
- [4] Pengfei Du, Li Ran, Daosen Zhai, Ruiling Ren, and Qi Zeng. Admm-based distributed online algorithm for energy management in hybrid energy powered cellular networks. *IEEE Access*, 7:83343–83353, 2019.
- [5] Davide Fioriti, Antonio Frangioni, and Davide Poli. Optimal sizing of energy communities with fair revenue sharing and exit clauses: Value, role and business model of aggregators and users. *Applied Energy*, 299:117328, 2021.
- [6] Matteo Fischetti, Ivana Ljubić, Michele Monaci, and Markus Sinnl. On the use of intersection cuts for bilevel optimization. *Mathematical Programming*, 172(1-2):77–103, 2018.
- [7] Gabriela Hug, Soumya Kar, and Chenye Wu. Consensus + innovations approach for distributed multiagent coordination in a microgrid. *IEEE Transactions on Smart Grid*, 6(4):1893–1903, 2015.

- [8] Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.
- [9] Soumya Kar and Gabriela Hug. Distributed robust economic dispatch in power systems: A consensus + innovations approach. In *2012 IEEE Power and Energy Society General Meeting*, pages 1–8, 2012.
- [10] Mengmou Li. Generalized lagrange multiplier method and kkt conditions with an application to distributed optimization. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 66(2):252–256, 2019.
- [11] Qiang Li, Yuexi Liao, Kunming Wu, Leiqi Zhang, Jiayang Lin, Minyou Chen, Josep M. Guerrero, and Derek Abbott. Parallel and distributed optimization method with constraint decomposition for energy management of microgrids. *IEEE Transactions on Smart Grid*, 12(6):4627–4640, 2021.
- [12] Stefano Lilla, Camilo Orozco, Alberto Borghetti, Fabio Napolitano, and Fabio Tossani. Day-ahead scheduling of a local energy community: An alternating direction method of multipliers approach. *IEEE Transactions on Power Systems*, 35(2):1132–1142, 2020.
- [13] Yun Liu, Hoay Beng Gooi, and Huanhai Xin. Distributed energy management for the multi-microgrid system based on admm. In *2017 IEEE Power & Energy Society General Meeting*, pages 1–5, 2017.
- [14] Leonardo Lozano and J Cole Smith. A value-function-based exact approach for the bilevel mixed-integer programming problem. *Operations Research*, 65(3):768–786, 2017.
- [15] Daniel K Molzahn, Florian Dörfler, Henrik Sandberg, Steven H Low, Sambuddha Chakrabarti, Ross Baldick, and Javad Lavaei. A survey of distributed optimization and control algorithms for electric power systems. *IEEE Transactions on Smart Grid*, 8(6):2941–2962, 2017.
- [16] James T Moore and Jonathan F Bard. The mixed integer linear bilevel programming problem. *Operations research*, 38(5):911–921, 1990.
- [17] Stefano Moret, Frédéric Babonneau, Michel Bierlaire, and François Maréchal. Decision support for strategic energy planning: A robust optimization framework. *Eur. J. Oper. Res.*, 280(2):539–554, 2020.

- [18] Anthony Roy, Jean-Christophe Olivier, François Auger, Bruno Auvity, Salvy Bourguet, and Emmanuel Schaeffer. A comparison of energy allocation rules for a collective self-consumption operation in an industrial multi-energy microgrid. *Journal of Cleaner Production*, 389:136001, 2023.
- [19] Mariam Sangaré, Eric Bourreau, Bernard Fortz, Amaury Pachurka, and Michael Poss. Loads scheduling for demand response in energy communities. *Computers & Operations Research*, 160:106358, 2023.
- [20] Sajad Aliakbari Sani, Olivier Bahn, and Erick Delage. Affine decision rule approximation to address demand response uncertainty in smart grids’ capacity planning. *Eur. J. Oper. Res.*, 303(1):438–455, 2022.
- [21] Miadreza Shafie-Khah and Pierluigi Siano. A stochastic home energy management system considering satisfaction cost and response fatigue. *IEEE Transactions on Industrial Informatics*, 14(2):629–638, 2018.
- [22] Gulai Shen, Zachary E. Lee, Ali Amadeh, and K. Max Zhang. A data-driven electric water heater scheduling and control system. *Energy and Buildings*, 242:110924, 2021.
- [23] Matthieu Stephant, Dhaker Abbes, Kahina Hassam-Ouari, Antoine Labrunie, and Benoît Robyns. Distributed optimization of energy profiles to improve photovoltaic self-consumption on a local energy community. *Simulation Modelling Practice and Theory*, 108:102242, 2021.
- [24] Philipp Ströhle and Christoph M Flath. Local matching of flexible load in smart grids. *European Journal of Operational Research*, 253(3):811–824, 2016.
- [25] Antonis G. Tsikalakis and Nikos D. Hatziargyriou. Centralized control for optimizing microgrids operation. *IEEE Transactions on Energy Conversion*, 23(1):241–248, 2008.
- [26] Wim van Ackooij, Jérôme De Boeck, Boris Detienne, Stefania Pan, and Michael Poss. Optimizing power generation in the presence of micro-grids. *Eur. J. Oper. Res.*, 271(2):450–461, 2018.
- [27] Kunming Wu, Qiang Li, Jiayang Lin, Yongli Yi, Ziyu Chen, and Minyou Chen. Fast distributed lagrange dual method based on accelerated gradients for economic dispatch of microgrids. *Energy Reports*, 6:640–648, 2020. 2020 The 7th International Conference on Power and Energy Systems Engineering.

- [28] Pan Xu and Lizhi Wang. An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions. *Computers & Operations Research*, 41:309–318, 2014.