A distributed and parallel 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 and parallel scheduling method for demand response in energy communities with distributed generation and storage. 2023. hal-04188878

HAL Id: hal-04188878
https://hal.science/hal-04188878
Preprint submitted on 27 Aug 2023

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

Mariam Sangaré¹, Eric Bourreau¹, and Michael Poss¹

¹ LIRMM, University of Montpellier, Montpellier, France, {eric.bourreau,michael.poss,mariam.sangare}@lirmm.fr

Abstract

This paper presents a distributed optimization methods 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 for the dimensions at hand, we develop a heuristic iterative algorithm where incentives are sent to the followers. Specifically, to maximize the collective-self consumption rate, we exploit the notion of allocation key traditionally used a posteriori for economic gain sharing to build an a priori incentive to Demand response. The proposed distributed management method returns a solution within 5% cost of the optimal centralized realistic benchmark's solution of an instance built over a month with accurate historical data from our demonstrator located in the south of France.

Keywords— Distributed management, Demand response, Collective self-consumption, Bilevel optimization.

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 [3, 4], and surplus trading on the other [10, 24, 11]. In this context, 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 to maximize local consumption of locally-produced energy.

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 i.e., each member can own or not some local tool. An energy community in this work refers to individuals connected to the same distribution network that produce, consume, and exchange energy. We specifically consider renewable energy communities as defined by [3]. A community coordinator (CC) ensures the community’s good functioning: in terms of collective self-consumption; for example, an aggregator collects the energy surpluses. In such a community, producers can store (if they own a storage system) or exchange their surpluses with other members or the aggregator. No additional links exist between members apart from the green energy exchange ones, i.e., they do not exchange their private information and preferences. CC is the intermediary between members. It collects specific information and shares incentives to ensure the whole community’s good management. Each member remains directly connected to the primary grid and collects energy when needed. In this context, CC maximizes collective self-consumption by optimizing the
periodic day-ahead incentives while considering the aggregator’s periodic buying prices and the main grid’s selling prices. CC must build incentives to Demand response (DR) since the members do not directly communicate. DR is a load profile modulation granted by an individual in return for a certain amount of consideration. Thus, CC receives each member’s hourly energy demand and prosumers’ energy availability to compute the 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 [7] fully.

As shown in Figure 1, a member can be a traditional consumer, a prosumer with a battery storage system (BSS), or a consumer owning BSS. Following [33], we assume each member has the required control and exchange equipment for this study. Each member must perform certain tasks that we classify into three classes according to their characteristics. The first class contains tasks that allow modulation of certain 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 desired time. We use function $\theta_j$ to evaluate the temperature in member $i$’s room where task $j \in J_{iA}$ is performed.

- The temperature variation function derived from the Newton cooling law [1] $\theta_1$ is:
  \[
  \theta_1(p_h) = \theta_1(p_{h-1}) + \frac{\Delta}{C_r}(p_h - U(\theta_1(p_{h-1}) - T_{out}^h)),
  \]
  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 second. $T_{out}^h$ is the external temperature at period $h$.

- For a water heater, using the method from [30], the temperature variation function $\theta_2$ is given by
  \[
  \theta_2(p_h) = \frac{\Delta_h}{M_{c_p}}(-\frac{S}{R} [\theta_2(p_{h-1}) - T_{room}^h] + v p_h - 1000 M_w c_p [\theta_2(p_{h-1}) - T_{in}]) + \theta_2(p_{h-1}),
  \]
  where $c_p$ is the isobaric specific heat capacity of water (kcal/kg·°C), $v$ 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 $m^2·°C/W$ 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 $T_{room}^h$ is the temperature of the ambient environment. 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 $j \in J_{iA}$ that is performed in room $k$ of member $i$ has a power $\bar{p}_{ijk} > 0$, expressed in kW. In this work, we discretize $[0, \bar{p}_{ijk}]$ by considering only the integer values in $[1, \bar{p}_{ijk}]$ ($0$ is not considered as in that case, the device is turned off). Our numerical experiments consider devices with $\bar{p}_{ijk} = 2$. 

![Figure 1: Community.](image)
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 $i$ must decide a schedule in a set of feasible schedules $S_{ij}$ to perform task $j$. The last class, called type C class, contains uncontrollable tasks. Type C tasks' periodic energy consumption is estimated for each member during the planning horizon. The question we address in this paper is: how to efficiently perform the tasks in a community to maximize the collective self-consumption rate?

Optimization is crucial at all levels of energy system planning. Long-term planning is addressed by [27, 22]. The authors of [27] study a model for extending a power system. The goal is to determine the investment to expand a multi-commodity power system. To do so, the authors consider Demand Response as a capacity reserve. The resulting problem is intrinsically uncertain since DR remains uncertain even over short-term planning. Planning in energy systems is subject to many uncertainties making robust models untenable given the systems’ size. In this context, [22] presents a robust optimization framework considering all uncertain parameters. In medium-term planning, [34] presents a bilevel problem of a single leader and several followers. 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, [32] and our work are in short-term planning. The authors of [32] 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 exist in the literature on energy system planning: 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 individual, are widely used in the literature [5, 33] as well as the aforementioned [27, 34]. While they can return optimal solutions, they have some drawbacks:

- The large computational capacity requirement.
- The information bottlenecks which lead to long transfer times.
- The question of social acceptance, since people may be reluctant to accept instructions from a third party regarding the use of their devices [29].

In our context, distributed optimization refers to each member locally determining their loads. Therefore, distributed optimization is a relevant alternative for large electrical systems. Indeed, the computational effort is not borne by an individual, but each member locally solves an 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 [6, 18, 19, 20, 31], dual decomposition [21] and consensus + innovations [13, 15].

Regarding ADMM, [19] proposes a method for day-ahead scheduling and compares the centralized and distributed approaches. Reference [18] presents a distributed model for multi-microgrids that cooperate to reduce the aggregated operational cost. The authors of [31] 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) management constraints. On a ring network composed of several agents, the next agent $i$ starts to calculate and update only after receiving information from the previous agent, which can be problematic for more complex problems. The authors of [6] then present a parallel and distributed method for energy management to increase ADMM’s convergence rate.

Turning to Consensus + innovation, [13] presents an approach to coordinate the loads, energy generation, and storage for a set of auto-sufficient agents that compose 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 [15] 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.

Finally, the ADMM works in a cascade, making it ineffective for large systems. The authors of [35] present a distributed Lagrangian dual method that rapidly converges based on Nesterov accelerated
gradient and dynamic set-sizes. The author of [17] generalizes the Lagrangians to a wider class of
functions that satisfies the strong duality between primal and dual problems then, derives the generalized
Karush–Kuhn–Tucker conditions for this generalized Lagrange multiplier method which is helpful to
provide consensus protocols.

In this paper, we introduce a bilevel formulation to manage the community:

\[
\begin{align*}
    x^*_{L} & = \min_{x_{L} \in \mathcal{X}_{L}} f_{L}(x_{L}, x_{F}) \quad (1a) \\
    \text{s.t.} & \quad x_{F} \in \arg \max_{x_{F} \in \mathcal{X}_{F}} g_{F}(x_{L}, x_{F}), \quad (1b)
\end{align*}
\]

where \(x_{L}\) and \(x_{F}\) are, respectively, the leader’s and followers’ decision variables, \(f_{L}\) is the leader’s
objective function which in our case is the energy coming from the primary grid. Each follower’s objective
function is the gain which is negative if the member’s global need is greater than the energy availability;
it is positive otherwise. This bilevel formulation is hardly solvable due to integer variables and the
problem’s size. Thus, we introduce a heuristic where CC provides the surplus allocation rule called:
allocation key as an incentive. 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 [25]. 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 her load, need, withdraws, and/or storage and injection schedules.
Then, she sends her energy need and availability information to CC, which updates its incentives. Since
there are no direct information links between each pair of members, prosumer \(i\) 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 aggregator; otherwise, the remaining required energy
is collected from the primary grid. The main aim is to consume local generation as much as possible
locally. As the BSS deteriorate with use, the coordinator must provide some selling prices which cover
the marginal degradation cost if it wants them to provide flexibility.

The rest of the paper is structured as follows Section 2 details the formulation of each level of
formulation (1). Section 3 presents an iterative heuristic, returning the efficient solution. Section 4
presents the experimental results. The paper concludes with a conclusion in Section 5.

2 Mathematical formulation

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

2.1 Leader’s problem

CC determines the periodic day-ahead the incentive to maximize the collective self-consumption rate.
This incentive that we note \(F^{\text{RCom}}_{\text{h}}\) is the maximum amount of energy member \(i\) can draw from the
community at period \(h\). The leader shares the energy between members via \(F^{\text{RCom}}_{\text{h}}\) and the remaining
energy at period \(h\) is denoted \(I_{h}\) which is injected into the main grid. Recall \(f_{L}\) is the leader’s objective
function, which designates the total amount of energy collected from the main grid. Minimizing \(f_{L}\)
is equivalent to maximizing the collective self-consumption rate. The leader’s problem is:

\[
x^*_L = \min_{x_L \in \mathcal{X}_L} f_L(x_L, x_F), \quad (2)
\]

where \(x_F\) represent followers’ response to leader’s decision, and \(x_L = (F^{\text{RCom}}, I)\). More explicitly, the
leader’s problem is:

\[
\min \sum_{i \in N} \sum_{h \in H} C_{ih} \quad (3a)
\]
Each follower $i$’s objective is to determine her consumption, injection, and storage schedules to maximize her gain $g_F$. Let $X_F$ be the feasible region of the lower level’s decision variable $x_F$. Thus:

$$x_F^* = \max_{x_F \in X_F} g_F(x_L, x_F)$$

(4)
Constraints (6a) and (6b) avoid the simultaneous BSS’s charge and discharge. Variable $T_{ijkh}$ is equal to 1 if and only if task $j \in J_{iA}$ is at progress at period $h$ in member $i$’s room $k \in K_{ij}$ and the device is on power level $p$,

$x_{ij}^c \in \{0, 1\}$ is equal to 1 if and only if schedule $s \in S_{ij}$ is chosen for member $i$’s task $j \in J_{iB}$, $z_{ih} \in \{0, 1\}$ is equal to 1 if and only if $i$’s BSS is charging in time period $h$, $w_{ih} \in \{0, 1\}$ is equal to 1 if and only if $i$’s BSS’s state changes from discharging or inactive to charging at period $h$,

$T_{ijkh} \geq 0$ the temperature in room $k$ of $i$ where task $j \in J_{iA}$ is performed at period $h$, $q_{ih} \in \mathbb{R}$ is the amount of energy injected into/out $i$’s BSS at period $h$, with $q_{ih} \leq 0$ if $b$ discharges, $q_{ih} \geq 0$ if $b$ charges, and $q_{ih} = 0$ if $b$ is inactive,

$E_{bih} \geq 0$ is the amount of electricity available in battery $b \in B_i$ of $i$ at the end of period $h$, $F_{i}^{out} \geq 0$ is the amount of energy injected by member $i$ at period $h$,

$F_{i}^{in} \geq 0$ is the amount of energy drawn from the community by member $i$ at period $h$, $C_{ih} \geq 0$ is the amount of energy withdrawn from the grid by member $i \in N$ at period $h$

$\tilde{G}_i \in \mathbb{R}$ is the gain of member $i$ when operating in the community.

Table 2: Each member’s variables.

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

$$\sum_{s \in S_{ij}} x_{ij}^s = 1 \quad j \in J_{iB} \quad (5a)$$

$$\forall j \in J_{iA}, k \in K_{ij}, \sum_{p \in P_{ijk}^A} x_{ijkh}^A = 1 \quad h \in H \quad (5b)$$

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

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

$$T_{ijkh} = \tilde{T}_{ijh}^{all} \quad (5e)$$

Recall a type A task is performed to regulate the temperature in certain rooms (room or water heating), and the type B task must be performed under time window and consumption level constraints. Constraint (5a) refers to type B tasks. Indeed, for each type B task, exactly one schedule must be selected. Constraints (5b)–(5c) 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 to be in the desired comfort zone, and (5e) sets the initial rooms’ temperature.

Another individual $\forall i \in N$ constraints concern the BSS management constraint:

$$\Delta_h d_{i} P_{i}^{out}(z_{ih} - 1) \leq q_{ih} \quad h \in H \quad (6a)$$

$$q_{ih} \leq \Delta_h c_{i} P_{i}^{in} z_{ih} \quad h \in H \quad (6b)$$

$$z_{ih} - z_{i(h-1)} \leq w_{i(h-1)} \quad h \in 2, \ldots, |H| \quad (6c)$$

$$\frac{E_{ih} - q_{i(h-1)}}{E_{i(h-1)}} = \eta_i \quad h \in 1, \ldots, |H| \quad (6d)$$

$$\sum_{h \in H} w_{ih} \leq \phi_i \quad (6c)$$

$$E_{ih} \leq \Gamma_i \quad h \in H \quad (6f)$$

$$E_{i0} = \xi_i \quad (6g)$$

Constraints (6a) and (6b) avoid the simultaneous BSS’s charge and discharge. Variable $q_{ih}$ is the amount injected if positive. Otherwise, it is 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 $i$ maximizes her gain $\tilde{G}_i$. Constraint (7a) calculates member $i$’s gain that must not
deteriorate by more than $\beta$% of $i$’s state when she does not exchange energy (7b). Prosumers gain
also depends on the amount injected in the aggregator, we denote $\tau_{ih}$ as a parameter that designate
member $i$’s contribution to this amount at period $h$.

\[
\tilde{G}_i = \sum_{h \in H} \left[ v_h^{\text{Com}} F_{ih}^{\text{out}} - v_h^{\text{Com}} F_{ih}^{\text{in}} + \tau_{ih} v_h^{\text{GES}} I_h - v_h^{\text{MG}} C_{ih} \right] \Delta_h 
\]  

(7a)

\[
\frac{\tilde{G}_i - G_i}{|G_i|} \geq \beta.
\]  

(7b)

In addition, each member $i$ has some global operational constraints at each period $h \in H$ which are:

\[
\sum_{j \in J, k \in K_{ij}} \sum_{p \in P_{ijk}} p \cdot x_{ijk}^{A} + \sum_{j \in J, h \in S_{ij}} P_{ijh} C_{ijh} x_{ijh}^{A} + \sum_{j \in J} q_{ih} = F_{ih}^{\text{gen}} + C_{ih} + F_{ih}^{\text{s}} - F_{ih}^{\text{out}}
\]  

(8a)

\[
F_{ih}^{\text{out}} \leq F_{ih}^{\text{gen}} + F_{ih}^{\text{Con}} \Omega_{i}
\]  

(8b)

\[
F_{ih}^{\text{in}} + C_{ih} \leq \pi_{i}
\]  

(8c)

\[
F_{ih}^{\text{in}} \leq F_{ih}^{\text{Con}}
\]  

(8d)

\[
F_{ih}^{\text{out}} \leq \pi_{i}
\]  

(8e)

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

Finally, we have the variables domains defined as follows: $\forall i \in N, j \in J_{A}, h \in H, p \in P_{ijk}, k \in K_{ij}, j' \in J_{B}, s \in S_{ij},$

\[
x_{ijk}^{A} \in \{0, 1\}
\]  

(9a)

\[
x_{ij}^{A} \in \{0, 1\}
\]  

(9b)

\[
z_{ih}, w_{ih} \in \{0, 1\}
\]  

(9c)

\[
F_{ih}^{\text{out}}, F_{ih}^{\text{in}}, C_{ih}, E_{ih} \geq 0, q_{ih} \in \mathbb{R}
\]  

(9d)

Overall, the lower level problem is:

\[
\max \quad \tilde{G}_i 
\]  

\[
\text{s.t.} \quad (5), (6), (7), (8), (9).
\]  

2.3 The full bilevel problem

The bilevel problem’s generic form is:

\[
x^* = \min_{x_L \in X_L} f_L(x_L, x_F)
\]  

(11a)

\[
s.t. \quad x_F \in \arg \max_{x_F \in X_F} g_F(x_L, x_F),
\]  

(11b)

which contains some integer variables in the lower level. Therefore, classical approaches like KKT
reformulation cannot be used. Despite the recent progresses for solving mixed-integer bilevel linear
program [8, 16], some based on intersection cuts [9], these problems remain extremely difficult to solve
easily and hardly capable of addressing the large problems we encounter in this work. Therefore, we
propose a greedy algorithm where each member operates independently of the others, and the coordinator
coordinates the demand response.
3 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

\[
\text{max } G \\
\text{s.t.} (5), (6), (7a), (8), (9).
\]

Model (12) does not contain constraint (7b) since the prosumer’s gain depends on the ability to exchange energy. 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, and who sends the received energy. Thus, at each iteration, CC will provide the information \( F_{\text{RCom}} \) to incentivize the members to shift their consumption to period \( h \in H \) where \( F_{\text{RCom}} \) exceeds zero.

3.1 Allocation keys

Allocation keys are traditionally used as an after-the-fact means of distributing financial gains among members who inject their surplus during one month. Used a posteriori, allocation key is not an incentive since members cannot decide at this point. In this paper, we use allocation keys for day-ahead scheduling. We use this notion to incentivize members to modulate their consumption during their local day-ahead planning. Therefore, in this paper, the allocation key is an algorithm that returns values that indicates to each member the maximum amount of energy she can collect from the community.

**Definition 1.** An allocation key is an algorithm \( K \) that, given \( C_{ih} \) and \( F_{\text{out}} \), computes \( F_{\text{RCom}} \) for \( i \in N, h \in H \).

The key allows CC to indicate the energy availability state to each member without sharing private information with the members. Several a posteriori key exist in the literature [25], some currently used in the French electrical power system. Among the keys in the literature, we have the allocation in prorate information with the members. Several a posteriori key exist in the literature [25], some currently used.

- **Fair repartition (K1).** Presented in Algorithm 3 for \( l = 1 \), this key shares the energy according members’ demand. Lines 7–10 in Algorithm 3 address the case where the total injection \( \sum_{i \in N} F_{ih}^{\text{out}} \) exceeds the demand \( \sum_{i \in N} C_{ih} \) for \( h \in H \). In this case, each member \( i \) receives the amount \( C_{ih} \) at period \( h \). In lines 14–31, we consider 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 29–31 the residual is insufficient to serve each member, i.e., \( u \cdot C_{\text{min}} \geq \text{residual} \), where \( u \) is the number of unsatisfied members so, we share the residual uniformly among the remaining members. In lines 34–35, we handle the case where the energy available is insufficient to satisfy the total need and not sufficient to allocate \( C_{\text{min}} \) to each member, so we share the energy available uniformly between members. Finally, in lines 38–40, value \( F_{\text{RCom}}^{ih} \) equals zero for each member \( i \) if the total injection equals zero at period \( h \).

- **Peer need (K2).** This key consists in determining the candidates for the energy reception and then sharing (uniformly) the power between these members. Member \( i \) is a candidate at period \( h \) if she does not inject energy, i.e., if \( F_{ih}^{\text{out}} = 0 \). 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 \( \sigma_{ih} \) is equal to 1 if member \( i \in N \) is a candidate to receive energy from other members at period \( h \in H \), in other words, if \( F_{ih}^{\text{out}} = 0 \). Value \( \sigma_{ih}^{\text{tot}} \) is the number of candidates to receive energy from the community at period \( h \in H \).

We set \( F_{ih}^{\text{in}} = \sum_{i \in N} F_{ih}^{\text{in}} \) and \( F_{ih}^{\text{out}} = \sum_{i \in N} F_{ih}^{\text{out}} \) for \( h \in H \) for simplicity in Algorithm 2.
Notice that keys to the one who consumes the most while sharing the surplus. Key $K_2$ may encourage consumption modulation. Note that by using the key modulate their consumption in the previous setting, as shown in the results section. In contrast, key $F$ schedules after receiving is the amount of remaining energy after each iteration using key $E$. Energy might receive more than required at certain times since the surplus is allocated uniformly, hence $i$ member available using the allocation key, sends values $F_i$ to each member who updates the local schedules.

### Algorithm 1: Key $K_2$ (per need).

**Input:** $F_{\text{out}}$

**Output:** initial $F_{\text{RCom}}$

\[
\sigma_{\text{Tot}}^i = |N|, \quad \sigma_{ih} \leftarrow 1 \text{ for } i \in N, h \in H
\]

for $h \in H$ do

for $i \in N$ do

if $F_{\text{out}}^{ih} > 0$ then

\[
\sigma_{ih} \leftarrow 0, \quad \sigma_{h}^\text{Tot} \leftarrow \sigma_{h}^\text{Tot} - 1
\]

for $i \in N$ do

\[
F_{\text{RCom}}^{ih} \leftarrow \sigma_{ih} \sum_{\forall i' \in N} F_{\text{out}}^{i'h} / \sigma_{h}^\text{Tot}
\]

### Algorithm 2: Update $K_2$.

**Input:** $F_{\text{in}}$, $F_{\text{out}}$, and $F_{\text{RCom}}$

**Output:** updated $F_{\text{RCom}}$

\[
\sigma_{h}^\text{Tot} = |N|, \quad \sigma_{ih} \leftarrow 1 \text{ for } i \in N, h \in H
\]

for $h \in H$ do

for $i \in N$ do

if $F_{\text{RCom}}^{ih} - F_{\text{in}}^{ih} > 0$ then

\[
\sigma_{ih} \leftarrow 0, \quad F_{\text{RCom}}^{ih} \leftarrow F_{\text{in}}^{ih}
\]

\[
\sigma_{h}^\text{Tot} \leftarrow \sigma_{h}^\text{Tot} - 1
\]

for $i \in N$ do

\[
F_{\text{RCom}}^{ih} \leftarrow F_{\text{RCom}}^{ih} + \sigma_{ih}(F_{\text{out}}^{ih} - F_{\text{in}}^{ih}) / \sigma_{h}^\text{Tot}
\]

- **Fair repartition (K3).** Key $K_1$ 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 keys $K_1$ and $K_2$ in a key called $K_3$. Key $K_3$ is presented in Algorithm 3 for input $l = 3$. We add some lines into $K_1$ that allow determining the candidates as defined previously. Thus, $K_3$ allocates energy according to consumption to members who need energy.

- **In prorate to consumption (K4).** This key consist in sharing the surplus in prorate to consumption between members in need. Thus,

\[
F_{\text{RCom}}^{ih} = \frac{\sigma_{ih} C_{ih}}{\sum_{i' \in N; i',h > 0} C_{i'h}} \sum_{i' \in N} F_{\text{out}}^{i'h},
\]

where $\sigma_{ih}$ equals one if member $i$ is candidate to receive energy as defined in $K_2$.

- **In prorate to investment (K5).** This key consist in sharing the surplus between the members who need energy in prorate to their investment in storage and generation tools. Thus, let us note $\Sigma_i$ as member $i$’s total investment to acquire some tools,

\[
F_{\text{RCom}}^{ih} = \frac{\sigma_{ih} \Sigma_i}{\sum_{i' \in N; i',h > 0} \sum_{i' \in N} F_{\text{out}}^{i'h}}.
\]

Notice that keys $K_1$ and $K_3$ are fair in the sens of [14], i.e., the member who consumes the less is favored to the one who consumes the most while sharing the surplus. Key $K_1$ does not encourage members to modulate their consumption in the previous setting, as shown in the results section. In contrast, key $K_2$ may encourage consumption modulation. Note that by using the key $K_2$, some members needing energy might receive more than required at certain times since the surplus is allocated uniformly, hence the need to update values $F_{\text{RCom}}$ at each iteration to ensure more efficient management. Let us denote $F_{\text{out}}^{ih} = \sum_{i \in N} F_{\text{out}}^{ih}, h \in H$ and $F_{\text{in}}^{ih} = \sum_{i' \in N} F_{\text{in}}^{ih}, h \in H$ for each period $h \in H$. Thus, $F_{\text{rest}}^{ih} = F_{\text{out}}^{ih} - F_{\text{in}}^{ih}$, is the amount of remaining energy after each iteration using key $K_2$. Since the members determine their schedules after receiving $F_{\text{RCom}}$, these values should be dynamic to avoid giving energy to members that do not need it. Indeed, CC must reallocate $F_{\text{rest}}^{ih}$. We update key $K_2$ as presented in Algorithm 2. Member $i$ will not receive more energy then $F_{\text{in}}^{ih}$ at period $h$ if $F_{\text{RCom}}^{ih} - F_{\text{in}}^{ih} > 0$.

### 3.2 The overall algorithm

Each member initially solves the load scheduling MILP with no energy exchanges between the members, i.e., $F_{\text{RCom}}^{ih} = 0$ for $i \in N, h \in H$. That allows members to determine their initial periodic energy needs and availability. The members then send their injection and need information to CC, who shares the energy available using the allocation key, sends values $F_{\text{in}}^{ih}$ to each member who updates the local schedules,
Algorithm 3: key K1 and K3 (fair repartition).

**Input**: C, F\textsubscript{out} and l ∈ {1, 3}

**Output**: F\textsubscript{Rec}\textsubscript{Com}

1. if l = 3 then σ\textsubscript{Tot} = |N| for h ∈ H; σ\textsubscript{ih} = 1 for i ∈ N, h ∈ H

2. for h ∈ H do

3. if \( \sum_{i \in N} F\textsubscript{ih} > 0 \) then

4. for i ∈ N do

5. if \( F\textsubscript{ih} > 0 \) and l = 3 then

6. \( \sigma\textsubscript{ih} \leftarrow 0; \quad \sigma\textsubscript{Tot} \leftarrow \sigma\textsubscript{Tot} - 1; \quad C\textsubscript{ih} \leftarrow +\infty \)

7. if \( \sum_{i \in N} F\textsubscript{ih} \geq \sum_{i \in N} C\textsubscript{ih} \) then

8. for i ∈ N do

9. if l = l then

10. \( F\textsubscript{Rec}\textsubscript{Com} \leftarrow C\textsubscript{it} \)

11. else

12. \( F\textsubscript{Rec}\textsubscript{Com} \leftarrow C\textsubscript{it} \cdot \sigma\textsubscript{ih} \)

13. else

14. \( C\textsubscript{min} \leftarrow \min_{i \in N}(C\textsubscript{ih}) \)

15. if l = 1 then u ← |N| else u ← σ\textsubscript{Tot}

16. if \( u \cdot C\textsubscript{min} \leq \sum_{i \in N} F\textsubscript{ih} \) then

17. for i ∈ N do

18. if l = 1 then idx\textsubscript{i} ← i else idx\textsubscript{i} ← i \cdot \sigma\textsubscript{ih}

19. residual ← \( \sum_{i \in N} F\textsubscript{ih} \)

20. while \( u \cdot C\textsubscript{min} \leq \text{residual} \) do

21. m ← 0

22. for i ∈ N do

23. if idx\textsubscript{i} ≠ 0 then

24. \( F\textsubscript{Rec}\textsubscript{Com} \leftarrow F\textsubscript{Rec}\textsubscript{Com} + C\textsubscript{min} \)

25. if \( C\textsubscript{ih} = C\textsubscript{min} \) then

26. m ← m + 1; idx\textsubscript{i} ← 0; \quad C\textsubscript{ih} \leftarrow +\infty \)

27. residual ← residual − u \cdot C\textsubscript{min}

28. u ← u − m; \quad C\textsubscript{min} ← \min_{i \in N}(C\textsubscript{ih})

29. for i ∈ N do

30. if idx\textsubscript{i} ≠ 0 then

31. \( F\textsubscript{Rec}\textsubscript{Com} \leftarrow F\textsubscript{Rec}\textsubscript{Com} + \text{residual}/u \)

else

32. for i ∈ N do

33. if l = 1 then

34. \( F\textsubscript{Rec}\textsubscript{Com} \leftarrow (\sum_{i \in N} F\textsubscript{ih}^\text{out} / u) \)

else

35. \( F\textsubscript{Rec}\textsubscript{Com} \leftarrow (\sum_{i \in N} F\textsubscript{ih}^\text{out} / u) \cdot \sigma\textsubscript{ih} \)

else

36. for i ∈ N do

37. \( F\textsubscript{Rec}\textsubscript{Com} \leftarrow 0 \)
participation in the global production, which multiplies the global need.

physical characteristic in terms of asset owing plus a variable quantity depending on the member

all over the horizon. These members will not receive more energy over the horizon: their solutions stay

in this paper. Observe that, for $K$ we set

We add constraints (14) to the members’ model (12).

5 The remaining energy and storage

The members that own BSS should be encouraged to store energy at certain periods of the planning

horizon. Indeed, since CC maximizes the consumption of locally produced energy and the member

the local profit, the prosumers can inject more energy than required to satisfy the global energy need.

If $\sum_{i \in N} F^\text{out}_i$ is greater than the energy requirement at period $h$, a portion of it can be consumed in the community, and the other is injected into the main grid (MG) or stored. Being decentralized, prosumers and CC cannot foresee what portion is consumed locally. Therefore, CC can only provide storage incentives when needed. Thus, members owning a BSS are encouraged to store if the need is less than the energy available or if CC wants the community to provide energy to the main network. The storage incentive must prioritize local consumption, hence, the interest in setting limits to the maximum a member who owns a battery can store. Let us introduce $\rho_i$ as the value which limits member $i \in N$ injection at period $h \in H$. We assume that $\rho_i$ is proportional to quantity depending on member $i$’s physical characteristic in terms of asset owing plus a variable a quantity depending on the member $i$’s participation in the global production, which multiplies the global need.

$$\rho_i = \delta_i + \sum_{i' \in N} F^\text{Gen}_{i' h} C_{i h} \quad h \in H.$$  \hfill (13)

We set $\rho_i$ for each $i \in N, h \in H$ such that it is proportional to the energy need, and inversely proportional to the community’s generation as presented in constraint (13), where $\delta_i$ 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 (14) to the members’ model (12).

$$F^\text{out}_h \leq \rho_i \zeta_i + \pi_i (1 - \zeta_i) \quad h \in H,$$  \hfill (14)

where $\zeta_i$ is equal to 1 if member $i \in N$ owns a storage unit, 0 otherwise and $\delta_i$ a physical parameter related to member $i$’s generation and storage systems’ size. Figure 6 presents the periodic $F^\text{rest}$ and $Need$ before and after adding constraints (14).
4 Numerical experiments

We use a realistic instance based on Smart Lou Quila’s seven-member demonstrator [23]. The planning horizon consists of a day sliced into 24 equal-length periods. The first six members’ real asset-owning characteristics are presented in Table 3. The last member does not have any assets. Members subscribed power are respectively $\pi = [36, 9, 6, 9, 6, 9, 9]$ kVa.

<table>
<thead>
<tr>
<th>Member 1</th>
<th>Member 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Photovoltaic (PV)</td>
<td>PV</td>
</tr>
<tr>
<td>PV Capacity</td>
<td>6.12kWp</td>
</tr>
<tr>
<td>Battery</td>
<td>Yes</td>
</tr>
<tr>
<td>Number</td>
<td>1 unit</td>
</tr>
<tr>
<td>Capacity</td>
<td>9.8kWh</td>
</tr>
<tr>
<td>Initial state of charge</td>
<td>4.5kWh</td>
</tr>
<tr>
<td>Efficiency</td>
<td>97.5%</td>
</tr>
<tr>
<td>Power</td>
<td>5kW</td>
</tr>
<tr>
<td>Periodic discharge rate</td>
<td>1%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member 3</th>
<th>Member 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV</td>
<td>No</td>
</tr>
<tr>
<td>PV Capacity</td>
<td>0kWp</td>
</tr>
<tr>
<td>Battery</td>
<td>Yes</td>
</tr>
<tr>
<td>Number</td>
<td>1 unit</td>
</tr>
<tr>
<td>Capacity</td>
<td>9.8kWh</td>
</tr>
<tr>
<td>Initial state of charge</td>
<td>4.5kWh</td>
</tr>
<tr>
<td>Efficiency</td>
<td>97.5%</td>
</tr>
<tr>
<td>Power</td>
<td>3.7kW</td>
</tr>
<tr>
<td>Periodic discharge rate</td>
<td>1%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Member 5</th>
<th>Member 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV</td>
<td>Yes</td>
</tr>
<tr>
<td>PV Capacity</td>
<td>3.2kWp</td>
</tr>
<tr>
<td>Battery</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 3: Production and storage assets description in the community.

We use the energy generation data collected on July 23rd 2022. Figure 2 presents prosumers’ individual production and the community’s total periodic production. The members’ loads are artificially generated. In other words, we generate some loads of the different types namely type A, B, and C. The following array $\nu$ shows the requested type A tasks by each member. We consider two different type A tasks i.e. $|J_A| = 2 \forall i \in N$; $\forall j \in J_A, j = 1$ means the corresponding task is room heating and $j = 2$ means that the requested task is water heating. Finally, $\nu_{ijk}$ is equal to 1 if task $j \in J_A$ is executed by member $i$ in room $k$, 0 otherwise. We refer to [26] for more details:

![Figure 2: Generation on July 23rd 2022.](image-url)
Table 4 presents members 2 to 7 house room characteristics. For simplicity we assume that each room $k$ of each member has an initial temperature equal to $22^\circ C$ and that the temperature of these rooms must lie in the interval $[t_{low}^{ik}, t_{up}^{ik}]$ during the time interval $[34,48]$. 

$$t_{low}^{ik} = \begin{pmatrix} 20 & 19 & 20 & 20 & 21 & 20 & 19 \\ 20 & 19 & 20 & 20 & 20 & 20 & 19 \end{pmatrix} \quad t_{up}^{ik} = \begin{pmatrix} 25 & 25 & 25 & 25 & 25 & 25 & 25 \\ 25 & 25 & 25 & 25 & 25 & 25 & 25 \end{pmatrix}$$

Each member has a single water heater, and Table 5 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 (kcal/h·m$^2$·°C). Finally, column $T_{water}$ presents the initial temperature of each member’s water heater. The members’ water must reach $60^\circ C$ at period 16. The efficiency of the electricity-to-heat transformation $\nu = Mc_p/(Mc_p r + \Delta SKr)$. Also, for simplicity, we assume that $\nu_{ih}$ which is member

Table 4: Rooms physical characteristics.

<table>
<thead>
<tr>
<th>Member</th>
<th>room</th>
<th>Surface</th>
<th>$C_p$</th>
<th>$U$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member 2</td>
<td>room 1</td>
<td>9m$^2$</td>
<td>297</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>room 2</td>
<td>15m$^2$</td>
<td>495</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>room 3</td>
<td>18m$^2$</td>
<td>594</td>
<td>24</td>
</tr>
<tr>
<td>Member 3</td>
<td>room 1</td>
<td>18m$^2$</td>
<td>594</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>room 2</td>
<td>9m$^2$</td>
<td>297</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>room 3</td>
<td>9m$^2$</td>
<td>297</td>
<td>12</td>
</tr>
<tr>
<td>Member 4</td>
<td>room 1</td>
<td>12m$^2$</td>
<td>396</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>room 2</td>
<td>20m$^2$</td>
<td>660</td>
<td>26.6</td>
</tr>
<tr>
<td></td>
<td>room 3</td>
<td>12m$^2$</td>
<td>396</td>
<td>16</td>
</tr>
<tr>
<td>Member 5</td>
<td>room 1</td>
<td>25m$^2$</td>
<td>825</td>
<td>33.3</td>
</tr>
<tr>
<td></td>
<td>room 2</td>
<td>10m$^2$</td>
<td>330</td>
<td>13.3</td>
</tr>
<tr>
<td></td>
<td>room 3</td>
<td>12m$^2$</td>
<td>396</td>
<td>16</td>
</tr>
<tr>
<td>Member 6</td>
<td>room 1</td>
<td>18.5m$^2$</td>
<td>610.5</td>
<td>24.6</td>
</tr>
<tr>
<td></td>
<td>room 2</td>
<td>9m$^2$</td>
<td>297</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>room 3</td>
<td>10m$^2$</td>
<td>330</td>
<td>13.3</td>
</tr>
<tr>
<td>Member 7</td>
<td>room 1</td>
<td>15m$^2$</td>
<td>495</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>room 2</td>
<td>10m$^2$</td>
<td>330</td>
<td>13.3</td>
</tr>
<tr>
<td></td>
<td>room 3</td>
<td>14m$^2$</td>
<td>462</td>
<td>18.6</td>
</tr>
</tbody>
</table>

Table 5: Water heaters characteristics.

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

$i \in N$ non-controllable consumption at period $h$, is equal to one for each member at each period.
4.1 Keys comparison

We present our numerical results in this section. First, we compare the key calculation methods, and an illustration of the need for CC to control storage. Then, we show the impact of this control on the distributed solution. Each member’s local scheduling problem (12) is reformulated using Special Ordered Set variables as presented in Section 7.1 of [26] with a time limit \(\text{time limit} = 20\) seconds. Table 6 presents the distributed solution for our instance. We do not update \(K_2\). In this table, Column \(\text{Avail}\) is the amount of photovoltaic energy that can be shared in the community or injected into the main grid, and column \(\text{Loss}\) is part of \(\text{Avail}\) that needs to be injected into the main grid. Table 6 shows

<table>
<thead>
<tr>
<th>Key</th>
<th>obj kWh</th>
<th>CPU</th>
<th>Available kWh</th>
<th>Loss kWh</th>
</tr>
</thead>
<tbody>
<tr>
<td>K1</td>
<td>125.62</td>
<td>0.20</td>
<td>27.97</td>
<td>13.78</td>
</tr>
<tr>
<td>K2</td>
<td>115.50</td>
<td>0.20</td>
<td>27.97</td>
<td>3.65</td>
</tr>
<tr>
<td>K3</td>
<td>117.96</td>
<td>0.20</td>
<td>27.97</td>
<td>6.11</td>
</tr>
<tr>
<td>K4</td>
<td>115.50</td>
<td>0.25</td>
<td>27.97</td>
<td>3.65</td>
</tr>
<tr>
<td>K5</td>
<td>119.30</td>
<td>0.25</td>
<td>27.97</td>
<td>7.44</td>
</tr>
</tbody>
</table>

Table 6: Key comparison.

that the distributed management with keys \(K_2\) and \(K_4\) 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 sharing scheme based on consumption will not incentive these members to shift the load. On the other hand, 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 \(K_2\), and two iterations for \(K_1\) and \(K_3\). Curves \(\text{Available}\) and \(\text{Used}\) represent respectively the amount of green energy injected by prosumers and the amount of green energy available which is locally consumed. \(\text{Total need}\) represents the total energy needed in the community for direct consumption or to be stored for ulterior consumption. The area between curves \(\text{Available}\) and \(\text{Used}\) is the energy injected into the main grid. It corresponds to column \(\text{Loss}\) in Table 6. Allocating energy to members who need it reduces the area between available and \(\text{Used}\) curves, leading to a good distributed solution.

Table 7 presents the solution instance using updated \(K_2\). Iteration 1 in column 1 corresponds to the case where \(F^{\text{Rec}} = 0\). Each column shows each member’s total energy collected from the main grid during the planning horizon, one row per member. Row \(\text{Sum}\) presents the energy collection for the whole community during the planning horizon. Updating keys provides good solutions for distributed
management since it leads to better allocation. Consequently, consumers shift more loads at available energy periods leading to a good solution since the objective value of the centralized optimum solution is 111.16 kWh. However, there is a need to incentive 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 than required at certain periods. For example, if we increase the total production $\sum_{i \in N} P_{Gen_{ih}}$ for $h \in H$ by 50%, the glutton algorithm leads to a solution with 79.34 kWh versus 64.02 kWh the optimal centralized solution’s objective value, which is an increase of 24%. The remaining energy $F_{rest}$ is presented by Figure 5. 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.

4.2 Storage control

As shown in Figure 5, there is a need to incentivize the prosumers owning a BSS to store energy at certain periods. We express this incentive as a limitation of the periodic injection of these members. Indeed, if they cannot inject, they will store for the next periods. We introduce constraint (14) to achieve it. Then, at each iteration, CC calculates $\rho_{ih}$ for $i \in N, h \in H$. In this paper, we set $\delta_i$ proportional to the size of member $i \in N$ compared to the other generating members who own the BSS. Each member $i$ owning a BSS has an initial $\delta_i = 1$ kWh. Suppose member $i'$ produces twice as much as the other members, $\delta_{i'} = 2$ kWh. Three members own the BSS, namely members 1, 2, and 3, according to the previous

<table>
<thead>
<tr>
<th>Members</th>
<th>Iteration</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td>7.6</td>
<td>7.6</td>
<td>7.6</td>
<td>7.6</td>
<td>7.6</td>
<td>7.6</td>
<td>7.6</td>
<td>7.6</td>
<td>7.6</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>34</td>
<td>26.84</td>
<td>26.21</td>
<td>25.75</td>
<td>25.66</td>
<td>25.58</td>
<td>25.56</td>
<td>25.54</td>
<td>25.54</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>31</td>
<td>23.84</td>
<td>23.21</td>
<td>22.75</td>
<td>22.66</td>
<td>22.58</td>
<td>22.56</td>
<td>22.54</td>
<td>22.54</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>34</td>
<td>26.84</td>
<td>26.27</td>
<td>26.09</td>
<td>26.06</td>
<td>26.04</td>
<td>26.03</td>
<td>26.03</td>
<td>26.03</td>
</tr>
<tr>
<td>Sum</td>
<td></td>
<td>139.83</td>
<td>115.5</td>
<td>113.5</td>
<td>112.35</td>
<td>112.14</td>
<td>111.94</td>
<td>111.9</td>
<td>111.86</td>
<td>111.86</td>
</tr>
</tbody>
</table>

Table 7: Distributed management solution.

Figure 4: Optimal distributed solution with updated $K_2$.

Figure 5: Storage requirement illustration.
affirmation $\delta = [2, 1, 1, 0, 0, 0, 0]$. We set the local scheduling time limit to 50 seconds for each member at each iteration.

Since each member maximizes the profit, without injection limitation, the prosumers will directly inject their surplus. That leads to important energy injected into the main grid (the area between the red and green curves, which corresponds to the curve $F_{\text{rest}}$ respectively in Figures 6a and 6b). Limiting prosumers owing BSS makes these members stock to reinject at other times, which allows having better solutions, as displayed on Figure 6b.

4.3 Test on real data

This section presents numerical results using accurate historical data collected from Smart Lou Quila’s demonstrator. 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 the a priori and a posteriori management and the keys. In the a posteriori case, the member $n$ solves the problem with all scheduling variables fixed except $C_{ih}, F^\text{out}_{nh}$ and $F^\text{in}_{nh}$ for $n \in N, h \in H$. In the a priori management, the variables related to storage are not fixed in addition to these previous variables.

We collect the generation and load data for one month, specifically, February 2023 for winter and August 2021 for summer. Figure 7a presents the community’s cumulative load and generation for the summer instance. The data are collected every 30 minutes over the horizons, respectively, 1344 and 1488 periods for February 2023 and August 2021. Each local problem is solve with a time limit $\text{time\_limit} = 50$ seconds at each iteration.

- **A posteriori comparison.** Here, we consider only the summer instance since it is the one that allows a clear a posteriori comparison; the prosumers inject more. For this instance, the community draws 4677.11 kWh from the main grid and injects 846.11 kWh into it if the members do not exchange the energy. Notice that a distributed solution is optimal if all members optimally solve their local problem at each iteration. If members exchange energy, the optimal distributed solution using key $K1$ leads the community to collect 4240.87 KWh from the main grid, and 48.4% of the injection (846.11 kWh) is not locally consumed. Key $K2$ returns an optimal solution of 4194.44 kWh drawn from MG, and 43% of the surplus goes to the main network. In the same line we obtain an optimal solution of 4209.16 kWh using key $K3$, 45% of the surplus is injected into the main grid. Finally, the keys $K4$ and $K5$ lead respectively to 4198.39 and 4369.30 collected from the main grid, respectively 43.4% and 63.6% of the surplus goes to the main grid.

- **A priori comparison.** For the winter instance, using key $K1$, the community collects 7474.9 kWh. The total injection of the community is 182.8 kWh, of which 25.3% is injected into the main grid mainly because the energy allocated to some members was not consumed. Updated $K2$ increases collective self-consumption, and only 0.13% of the local generation goes to the main grid. Key $K3$, which improves key $K1$ by allocating energy only to members who need it leads to 7453.78 kWh and 13.19% of the available energy injected into the main grid. For the summer instance, if the prosumers do not exchange their surplus in the local community, a total of 4560 kWh is collected from the main grid, and 714.77 kWh are injected into the main grid by the prosumers over the month. If members exchange their surplus, the optimal centralized solution leads to 3867.88 kWh collected from MG. For distributed management, on the one hand, using key $K1$, the community
collects 4153.15 kWh for the main grid. On the other hand, the members inject 714.77 kWh of their energy generation over the month, and 42.45% of this amount goes to the main grid. On the other hand, the community collects 4063.07 kWh from MG using updated $K_2$, and 28.7% of the injection goes to the main grid during summer. Key $K_3$ leads to 4133.72 kWh collected from the main grid, and 39.73% of the energy available goes to the main grid. In the line keys $K_4$ and $K_5$ lead respectively to 4079.66 and 4234.52 kWh collected from the main grid and 31.6% and 53% of the injection goes respectively to the main grid. Finally, limiting the BSS owners’ injection using key $K_2$ leads to 4062.85 kWh collected from MG while members inject 707.67 kWh from which 28.11% goes to the main grid.

Key $K_2$ returns better solutions because it allows, on the one hand, a better surplus allocation. On the other hand, it allows storage for ulterior usage. Indeed, keys $K_1$, $K_3$ and $K_4$ are related to the individual consumption level and $K_5$ is not related to the operations. In opposition, $K_2$ shares the surplus between the candidates uniformly, if a member $i$ owning a BSS receives at period $h$ an amount $f_{ih}$ greater than her consumption, i.e., $f_{ih} \geq C_{ih}$, she can store the excess for her ulterior usage (see Figure 7b). These BSS owners can store for the community only under CC signals on prices and injection limitations.

Experience has shown that, unlike the conventional approach, proactive management can effectively establish DR in a community. However, its effectiveness depends on the surplus allocation rules. On the other hand, there is a conflict between equity and maximizing the rate of collective self-consumption for a priori management. We recall that equity in this work consists in prioritizing the member who consumes the least when sharing the energy surplus. In this case, the keys $K_2$, $K_4$, $K_5$ are not fair. However, $K_2$ is the best solution, as it allows storage for later consumption.

Notice that we do not address the pricing issue in this work, i.e., we do not consider how prices $v_{Com}^h$ and $v_{Com}^h$ for $h \in H$ 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. That is why we use the following prices in euros per kWh:

\[
\begin{align*}
\bar{v}_h^\text{MG} &= v_h^\text{GES} = 0.1685, \\
\bar{v}_h^\text{Com} &= v_h^\text{Com} = 0.1400, \\
\bar{v}_h^\text{Cos} &= 0.065, \\
\tilde{v}_h^\text{GES} &= 0.065, \\
\tilde{v}_h^\text{Com} &= 0.12.
\end{align*}
\]

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

5 Conclusion

In this paper, we present a parallel and distributed management method for large energy communities, which leads to some reasonable solutions compared to centralized solutions. Furthermore, we introduce an efficient allocation key for a priori management, which leads to a solution at 5.2% from the optimal centralized solution on real world data. We finally emphasize the conflict between fair surplus allocation and the high self-consumption rate that rises for certain definitions of fairness.

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 at the aggregator, 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.

References


