Electrical Power and Energy Systems 117 (2020) 105643

Contents lists available at ScienceDirect

Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

Cooperative energy management of a community of smart-buildings: A Blockchain approach

Olivier Van Cutsem, David Ho Dac, Pol Boudou, Maher Kayal

Department of Electrical Engineering (STI), Ecole Polytechnique Fédérale de Lausanne, Lausanne 1005, Switzerland

ARTICLE INFO

Keywords:
Cooperative energy management
Smart-buildings
Smart-community
Blockchain
Demand response
Smart-contract
Linear optimization
Decentralized algorithm

ABSTRACT

Demand Response (DR) is progressively moving from a centralized, unidirectional structure to a set of advanced decentralized mechanisms that better balance distributed supply and demand. This paper presents a decentralized cooperative DR framework to manage the daily energy exchanges within a community of Smart-Buildings, in the presence of local Renewable Energy Sources (RES). The proposed algorithm taps into the flexibility of the participants to let them decide of a day-ahead community power profile, and subsequently ensures the forecast tracking during the next day. In practice, the algorithm is fully decentralized by the Blockchain technology, that enables a trusted communication medium among the participants and enforces autonomous monitoring and billing via Smart-Contracts. With such an energy management framework, participating Smart-Buildings can together aim at a common objective, such as carbon-free resources usage or aggregated grid services, without depending on a centralized aggregator/utility. Simulations on realistic Swiss building models demonstrate that nearly all the renewable production resources could be harnessed locally through the presented framework, compared to selfish individual optimization. Under a quadratic cost of grid electricity, the considered community profile could dramatically be flattened, hence avoiding costly peaks at the grid interface. A scalability analysis shows that, considering the current public Ethereum Blockchain, the framework could handle a community size of up to 100 Smart-Buildings.

1. Introduction

The increasing world demand in electricity is envisioned to be entirely supplied by sustainable Renewable Energy Source (RES) in order to counteract the global climate change. However, intrinsic volatility and uncontrollably of decentralized RES production pose severe challenges to the current electrical grid, that must ensure stability at any time. Progressively, the centralized grid sees a change in paradigms, transitioning from a system dispatching a production portfolio following the electrical demand to a Smart-Grid that handles a portfolio of controllable demand to match an uncontrollable supply [1].

To assist this transition, Smart-Buildings have recently emerged as a solution to leverage the flexibility offered by the various entities commonly found in buildings. Equipped with the right hardware and Information and Communication Technology (ICT) they can provide active Demand Response (DR) to the electrical grid [2,3]. DR regroups a set of mechanisms divided into *incentive* and *price-based* programs, that specify various signals to be exchanged between the grid and the consumers in order to shape the power profile of the latter. Many works have tackled the problem at individual building level, demonstrating their capability to adapt their power consumption to grid signals while ensuring occupant comfort [4–7].

Beyond individual building optimization, there is a need of handling the problem at the community level. By doing so, local resources such as Photovoltaic (PV) production can optimally be harnessed while reducing the overall peak power demand. Many optimization frameworks have emerged to collectively manage the energy of multiple users [8–12]. Nevertheless, these solutions require a central agent that collects user information to subsequently dispatch optimal set points to each of them. Even though generic simplified models can be used to represent buildings flexibility [9,12], centralized solutions still face issues of privacy, single point of failure, scalability, and market entry of small prosumers. Furthermore, the growing penetration of distributed RES leads to the need of decentralized and distributed DR solutions that become complex to solve centrally when considering a large community of flexible assets.

Game-Theory (GT) [13] and Peer-to-Peer (P2P) [14] energy trading have extensively been applied to energy scheduling in local microgrids/communities. On the one hand, GT defines a conceptual framework in which the individual actions of rational participants optimize a community objective [13,15–17]. Mohsenian-Rad et al. [16] decentralized the central grid planning optimization problem, such that every

E-mail address: olivier.vancutsem@epfl.ch (O. Van Cutsem).

https://doi.org/10.1016/j.ijepes.2019.105643
Received 8 July 2019; Received in revised form 30 August 2019; Accepted 21 October 2019

Nomenclature

Constants & parameters

α_b	Battery leakage coefficient (s^{-1})
η_b^+, η_b^-	Battery charging and discharging efficiency
a_l	Local production electric cost ($/kWh)
a_g^q	Quadratic cost of grid import ($/kWh^2)
a_g^l	Linear cost of grid import ($/kWh)
a_g^c	Constant cost of grid import ($)
$\overline{C}_b^k$	Max. k^{th} battery capacity (kWh)
$\underline{C}_b^k$	Min. k^{th} battery capacity (kWh)
$C_{b,a}^k$	Charge of the k^{th} EV upon arrival (kWh)
$C_{b,l}^k$	Charge of the k^{th} EV when leaving (kWh)
d_w^k	Water drawn from the k^{th} hot water tank (l/s)
dt	Sampling period (s)
H	Number of intervals in the planning forecast
l_*^+, l_*^-	Snapshot of import/export community forecast (kW)
N_{sb}	Number of Smart-Buildings in the community
N_{res}	Number of RES in the community
n_{pv}	Number of PV cells
$\overline{P}_{th}^k$	Max. power of the k^{th} thermal load (kW)
$\overline{P}_b^k$	Max. charging power of the k^{th} battery (kW)
$\underline{P}_b^k$	Max. discharging power of the k^{th} battery (kW)
$\underline{P}_{pv}^n$	Nominal power of the PV system (kW)
$\hat{\mathcal{P}}_d^k$	Power profile of the k^{th} deferrable load (kW)
$t_d^{s,k}$	Min. starting time of the k^{th} deferrable load (h)
$t_d^{e,k}$	Max. ending time of the k^{th} deferrable load (h)
T_a	Outside air temperature (°C)
$\mathcal{T}_{out}^k$	The set of periods in which the k^{th} EV is unplugged
$t_{b,l}^k$	The leaving time of the k^{th} EV (h)
$t_{b,a}^k$	The arriving time of the k^{th} EV (h)

Variables

e	Building net demand (kW)
p_d^k	k^{th} deferrable load power (kW)
u_{th}^k	k^{th} thermal load power (kW)
u_d^k	k^{th} deferrable load starting time (h)
u_b^k	Power of the k^{th} battery (kW)
$u_b^{+,k}$	Charging power of the k^{th} battery (kW)
$u_b^{-,k}$	Discharging power of the k^{th} battery (kW)
x_b^k	Charge state of the k^{th} battery(kWh)
y^+, y^-	Community grid import/export (kW)
y_l^-	Community power generation (kW)

Indices and notations

$\hat{\bullet}$	Vector $[\bullet_0, \ldots, \bullet_n]$
$\bullet[h]$	Value at discrete time period h
$\bullet^+, \bullet^-$	Positive/negative power demand related value
$\underline{\bullet}, \overline{\bullet}$	Minimum/Maximum value

participant in a microgrid solves locally a load scheduling problem taking into account the prediction sent by the others. The quadratic structure of the price of electricity incentivizes the whole community to reduce the aggregated Peak-to-Average Ratio (PAR). On the other hand, P2P energy trading represents the virtual exchange of electricity among community participants, with the aim to locally match production and consumption[14]. P2P energy generally lays on GT principles to fix the price of energy transaction. In [18], a shared Energy Sharing Provider (ESP) is in charge of deciding the local prices, through an iterative process that involves all the local participants. Each consumer optimizes its objective function that is modelled as a combination of energy cost and inconvenience in load shifting.

Renowned for the cryptocurrency applications [19], the Blockchain has rapidly proven capabilities for energy trading and optimization in microgrids. Blockchains are distributed ledgers shared by participants that can securely store digital transactions, without the need of a central agent [20]. These transactions are aggregated into blocks, linked to one another by cryptography methods to form a chain of immutable information. The Blockchain technology traditionally relies on "miners" to validate new blocks emitted by participants by solving a common complex algorithm that ensures a tamper-proof system. When adding Smart-Contracts into the Blockchain, like the Ethereum technology [21], decentralized algorithms can practically be deployed. A Smart-Contract is a piece of executable code shared by every node that defines immutable rules, running directly in the Blockchain. Practically, they are stored in specific blocks in the chain and the rules they define trigger subsequent logic to write data in the rest of the chain. This replaces the need of a centralized trusted entity to hold the algorithm logic, and can therefore foster the fast deployment of innovative community DR solutions.

Many significant energy trading projects using the Blockchain technology have been deployed worldwide, notably the Brooklyn Microgrid projects [22] and various research demonstrators [23–25]. In addition to the project description, Mengelkamp et al. formally present the seven market components that any efficient microgrid energy market framework should incorporate [22]. Authors of [26] demonstrated that the Blockchain technology represents a reliable mechanism for energy trading, compared to traditional centralized transactive energy schemes.

The use of Blockchain in microgrid goes beyond local energy trading, as thoroughly reviewed in [27,20]. For instance, authors in [28] were the first to use Smart-Contracts in distributed optimization, relying on them to play the role of an ADMM coordinator. In [29], authors applied Blockchain to deploy a GT algorithm that solves a DR problem and discuss how P2P trading could be incorporated in their work. Pop et al. [30] developed a decentralized solution to manage and monitor DR with the use of Blockchain. Their Smart-Contracts penalize the gap between the expected baseline and the actual consumption, and manage in real-time the microgrid imbalance.

In the present work, we propose an innovative generic framework to manage the energy in a community of flexible Smart-Buildings with local RES production. Unlike P2P energy trading [22–25] that only optimizes individual costs, the framework at hand allows participants to collectively optimize any generic objective, such as grid services or promoting local RES energy consumption. Practically, the framework works in two phases. During the planning phase, the participants iterative propose a forecast of their power profile - consumption and/or production - until a consensus is collectively found. Compare to the iterative algorithm in [16], the planning phase also includes RES production and a generic bottom-up model of the buildings. Then, during the online phase, the participants ensure that their power profiles matches as much as possible their planning forecast.

Although sharing some similarities, our work is different from the frameworks of Noor et al. [29] and Pop et al. [30]. In this study, we use a generic building model that better represents end-user flexibility such as thermal inertia, and we directly included local RES in the day-ahead optimization problem. Unlike [30] that used past data to construct the building baseline against which the participant is rewarded/penalized, the presented day-ahead decentralized algorithm defines the building baseline itself, allowing more flexibility in the decision. Moreover, we presented a reward/penalty decision based on the community behavior, instead of individual participant actions. The use of Blockchain, and more particularly Ethereum Smart-Contracts, enables both the decentralization of the energy management algorithms among untrustworthy participants and the monitoring of the community in real-time. Beyond the interest of Blockchain for price-based P2P energy trading leveraged in [29], we used it to enable smart-communities to collectively manage their energy in a flexible way according to their common interests. Finally, the simulation code is publicly available [31], and its modularity allows any developer to include additional models.

The paper is organized as follow. Section 2 presents the generic Smart-Building model. Then, Section 3 describes the community optimization framework with its generic community objectives, and the various Smart-Contracts to decentralize the logic. Section 4 then provides results of simulations and a general discussion. The paper is concluded in Section 5.

2. Smart-building model

In this section, a generic building model is presented. The prosumers in the framework are assumed to be equipped with adequate sensors/actuators, a smart-meter, and a Building Management System (BMS) to enable their energy management. Any variable/parameter that is not explained in the text can be found in the Nomenclature section.

A Smart-Building is an entity in the community that consumes or produces a total power $e[h]$ (kW) at a given time period h, which can be broken down as follow:

$$e[h] = p_{nc}[h] + p_g[h] + u_f[h] \tag{1}$$

where p_{nc} represents the non-controllable part of the building load consumption (kW), p_g the behind-the-meter power generation (kW), and u_f the total flexible load (kW). The latter sums up the following components:

$$u_f[h] = \sum_k^{n_{th}} u_{th}^k[h] + \sum_k^{n_d} p_d^k[h] + \sum_k^{n_b} u_b^k[h]$$

where n_{th} is the number of thermal loads, n_d the number of deferrable loads, and n_b the number of batteries present in the building. The rest of this section presents the models impacting the variable $e[h]$.

2.1. Thermal loads

The variable $\hat{\boldsymbol{u}}_{\boldsymbol{th}}$ (kW) influences the temperature of air zones where home occupants live or the temperature of the water in hydronic pipes/tanks, used by home occupants. The corresponding loads are Heating, Ventilation, and Air-Conditioning (HVAC) system, Heat Pumps (HP), and Electrical Water Heaters (EWH).

The conditioned zones/hydronic system state and water tank state evolve as follow [32]:

$$\hat{\boldsymbol{x}}_{\boldsymbol{th}}[h+1] = A\,\hat{\boldsymbol{x}}_{\boldsymbol{th}}[h] + B_u\,\hat{\boldsymbol{u}}_{\boldsymbol{th}}[h] + B_d\,\hat{\boldsymbol{d}}_{\boldsymbol{th}}[h] \tag{2}$$

$$\hat{\boldsymbol{T}}[h] = C\,\hat{\boldsymbol{x}}_{\boldsymbol{th}}[h] \tag{3}$$

where $\hat{\boldsymbol{x}}_{\boldsymbol{th}}$ (°C) regroups both the constrained thermal air/water temperatures and intermediate model states, such as wall temperatures. Thermal model disturbance vector $\hat{\boldsymbol{d}}_{\boldsymbol{th}}$ is made of outside temperature, internal load heat gain, and solar heat gain. The matrices A, B_u, B_d contain the building model parameters, such as Resistive-Capacitive (RC) equivalent model parameters and thermal load efficiency. More details about the thermal model can be found in Appendix A.

The air zone and water temperatures should ideally be kept within min/max limits and the thermal load power is physically limited:

$$\hat{\boldsymbol{T}}^{\boldsymbol{m}}[h] \leqslant \hat{\boldsymbol{T}}[h] \leqslant \hat{\boldsymbol{T}}^{\boldsymbol{M}}[h] \tag{4}$$

$$0 \leqslant \hat{\boldsymbol{u}}_{\boldsymbol{th}}[h] \leqslant \hat{\overline{\boldsymbol{P}}}_{\boldsymbol{th}} \tag{5}$$

where $T^{k,m}/T^{k,M}$ (°C) are the k^{th} air zone or water tank temperature limits, specified by the user.

2.2. Deferrable loads

A deferrable load refers to any appliance whose discrete starting time $\hat{\boldsymbol{u}}_{\boldsymbol{d}}$ (h) can be controlled and that cannot be interrupted once started. This category regroups mainly residential loads like washing machines, dryers, and washers. When switched on, the deferrable load power consumption is given by its predefined load profile:

$$p_d^k[h] = \begin{cases} \mathscr{P}_d^k[h-u_d^k] & \text{if } u_d^k \leqslant h \leqslant u_d^k + |\mathscr{P}_d^k| \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $|\mathscr{P}_d^k|$ refers to the duration of the deferrable load profile. The model is constrained by home occupants preferences that can specify a minimum starting time and a maximum ending time:

$$\hat{t}_d^s \le \hat{u}_d \le \hat{t}_d^e - |\hat{\mathcal{P}}_d| \tag{7}$$

2.3. Energy storage system

The continuous variable $\hat{\boldsymbol{u}}_{\boldsymbol{b}}$ (kW) influences the State-of-Charge (SoC) of chemical batteries used in Electrical Vehicles (EV). An integrative model describes the SoC evolution [12]:

$$\hat{\boldsymbol{x}}_{\boldsymbol{b}}\left[h+1\right] = \left(\alpha_b\,\hat{\boldsymbol{x}}_{\boldsymbol{b}}[h] + \eta_b^+\,\hat{\boldsymbol{u}}_{\boldsymbol{b}}^+[h] + \frac{1}{\eta_b^-}\,\hat{\boldsymbol{u}}_{\boldsymbol{b}}^-[h]\right)dt \tag{8}$$

$$\hat{\boldsymbol{u}}_{\boldsymbol{b}}[h] = \hat{\boldsymbol{u}}_{\boldsymbol{b}}^+[h] + \hat{\boldsymbol{u}}_{\boldsymbol{b}}^-[h] \tag{9}$$

Battery charging/discharging power and capacity are physically limited. In order to ensure that $u_b^{k,+}[h]$ and $u_b^{k,-}[h]$ are not simultaneously non-null, binary variables s_b^k[h] are injected in the constraints:

$$\hat{\underline{\boldsymbol{C}}}_{\boldsymbol{b}} \leqslant \hat{\boldsymbol{x}}_{\boldsymbol{b}}[h] \leqslant \hat{\overline{\boldsymbol{C}}}_{\boldsymbol{b}} \tag{10}$$

$$0 \leqslant \hat{\boldsymbol{u}}_{\boldsymbol{b}}^+[h] \leqslant \hat{\boldsymbol{s}}_{\boldsymbol{b}}[h]\,\hat{\overline{\boldsymbol{P}}}_{\boldsymbol{b}} \tag{11}$$

$$0 \geqslant \hat{\boldsymbol{u}}_{\boldsymbol{b}}^-[h] \geqslant (1-\hat{\boldsymbol{s}}_{\boldsymbol{b}}[h])\,\hat{\underline{\boldsymbol{P}}}_{\boldsymbol{b}} \tag{12}$$

When the battery is used in an EV, initial and final conditions apply on SoC when unplugged, and at time of arrival and departure:

$$u_b^k[h] = 0 \quad \forall\, h \in \mathscr{T}_{out}^k \tag{13}$$

$$x_b^k[t_{b,a}^k] = C_{b,a}^k, \quad x_b^k[t_{b,l}^k] = C_{b,l}^k \tag{14}$$

2.4. Solar PV panel

Environmental conditions influence the power p_g locally generated by PV array installation. The PV output power is linearly modelled as follow [33]:

$$p^g[h] = f_{PV}(G[h], T_a[h])\; n_{pv}^{cells}\; P_{pv}^n \tag{15}$$

where $f_{PV}(.)$ is a function that modulates the PV array nominal power, depending on outside solar irradiance and cell temperature difference from standard condition $\Delta T[h]$:

$$f^{PV} = \frac{G[h]}{G_n}\left(1 + \alpha_i\;\Delta T[h]\right)\left(1 + \alpha_u\;\Delta T[h]\right)$$

where G_n is the nominal radiation $\frac{W}{m^2}$, α_i and α_u are the temperature sensitivity ($\frac{\%}{°C}$) of the PV output current and voltage, respectively.

2.5. Uncontrollable load

Building occupants and environmental conditions influence uncontrollable load behavior p_{nc}. Their power profile is supposed to be a given input parameter to the system.

3. Smart-buildings community framework

The community encompasses the local energy actors, the physical power system through which electricity flows, and the energy management mechanisms (cf Fig. 1). A common approach consists in using a local aggregator to centrally collects flexible building parameters and RES data to optimize a given objective, both in planning and real-time operations [3]. Fig. 1a depicts such a situation, where the centralized aggregator agent handles the communication with the grid operator.

The decentralized approach taken in this paper is depicted in Fig. 1b. Without the aggregator agent, the goal of the community is to agree on a consensus that optimizes a given shared objective function. The aggregator is entirely replaced by the Blockchain environment and every energy actor only interacts with the Blockchain. The latter orchestrates events that will trigger the appropriate Smart-Contracts functions to enable cooperative energy management.

The role of every participant in the community is to dispatch its flexible assets in order to meet an aggregated objective. The proposed mechanism works in two distinct phases. In the *day-ahead* phase, the community agrees upon an optimal planning of the aggregated load profile, transferred up to grid operator. To ensure that the actual community consumption is as close as possible to its planning, the *online* phase strives to track the *day-ahead* aggregated profile during the day.

3.1. Day-ahead energy planning phase

The planning profile is the result of the community cost function minimization problem, solved in a decentralized fashion. At time instant h, the community can buy local electricity $y_l^-[h]$ at a cost $f_l^c(y_l^-[h])$ and electricity coming from the grid $y^+[h]$ at a cost $f_{G^+}^c(y^+[h])$. If it doesn't import electricity from the grid, it might export $y^-[h]$ to the grid for a gain $f_{G^-}^c(y^-[h])$.

The responsibility to optimize the flexible assets falls into the Smart-Buildings themselves, instead of a central aggregator. Each participant in the community hence computes a local optimum for their planning, given an intermediate forecast of other nodes, to then shares their updated decision with the rest of the community. Iteratively, the Smart-Buildings will therefore adapt their power forecast, based on forecast actions taken by the others, in order to optimize an aggregated cost function. The resulting algorithm can be seen as a GT problem [13], in which the players are incentivized to change their power consumption, given grid utility electricity prices or a common community goal.

The local optimal planning of the i^{th} Smart-Building is given by the input sequence $\hat{\boldsymbol{u}}^{\boldsymbol{i}}$ solving:

$$\min_{\hat{\boldsymbol{u}}}^{\boldsymbol{i}} \sum_{h=0}^{H-1} f_{G^+}^c\left(y^+[h]\right) + f_l^c\left(y_l^-[h]\right) + f_{G^-}^c\left(y^-[h]\right) \tag{16}$$

s. t. $Eq.\ (2) and\ (8)\ init.\ state\ at\ h = 0$

$\forall\ h = 0...H-1$:

$$e^i[h] + l_{*-i}^+[h] + l_{*-i}^-[h] = y^+[h] + y^-[h] \tag{17}$$

$$e^{i,-}[h] + l_{*-i}^-[h] = y_l^-[h] \tag{18}$$

$$0 \leqslant y^+[h] \leqslant M\ s_y[h] \tag{19}$$

$$0 \geqslant y^-[h] \geqslant -M\ (1 - s_y[h]) \tag{20}$$

$$Eq.\ (1) - (15)\ at\ time\ h$$

where l_{*-i}^+ and l_{*-i}^- represent the demand and production forecast of the community minus the i^{th} Smart-Building, respectively, and $e^{i,-}$ is the net power export of the i^{th} building. The vector $\hat{\boldsymbol{u}}^{\boldsymbol{i}} = [\hat{\boldsymbol{u}}_{th}^{\boldsymbol{i}}, \hat{\boldsymbol{u}}_{d}^{\boldsymbol{i}}, \hat{\boldsymbol{u}}_{b}^{\boldsymbol{i}}]$ regroups the control input variables for all the flexible entities of the i^{th} building. Eq. (17) models the power balance at the grid entry point of the community, while Eq. (18) defines the total local production.

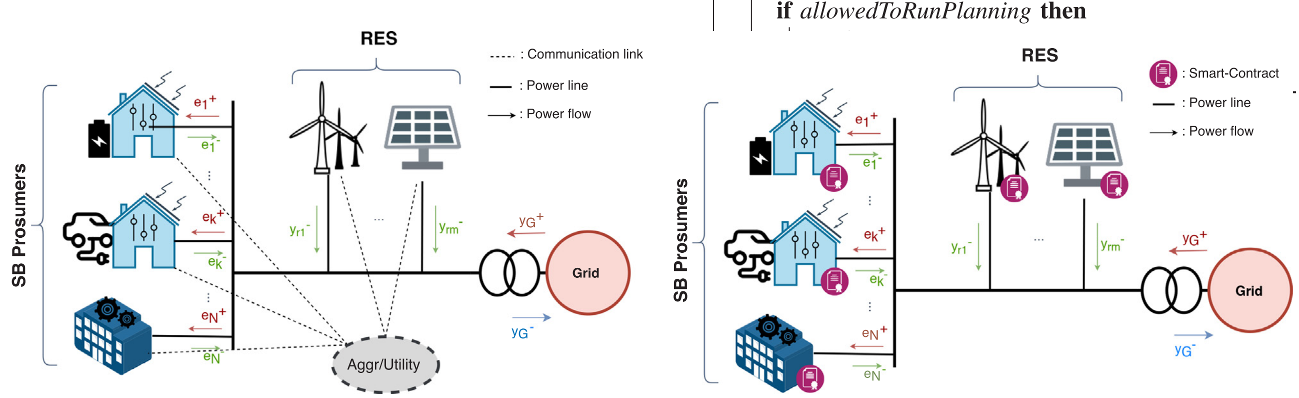
Binary variables $s_y[h]$ along with constant M ($\sim 10^6$) are used in Eqs. (19) and (20) to ensure that variable representing grid importing and exporting don't take simultaneously a non-null value at time period h, via the big-M method [34]. It's worth noticing that slack variables are used on the air/water temperature constraints in order to ensure feasibility of the problem but removed from Eq. (16) for the sake of clarity.

Practically, the decentralized algorithm sequences and information broadcasting are enabled by Smart-Contract 1. The functions called through the keyword "emit" are located at the building premise, while the Smart-Contracts run in the Blockchain. A periodic event (every 24 h) is used to trigger the *day-ahead* planning phase, by calling the corresponding function *startPlanningPhase* in the contract. To orchestrate the iterations of the algorithm, the function *updatePlanning* of the contract is in charge of periodically reading whether a new planning-related transaction has been written in the Blockchain. In that case, the Smart-Buildings update their community planning knowledge and then, if allowed to do so, run their own optimization, by solving Problem (16). Eventually, the decentralized planning phase will naturally be over when all the participants write on the Blockchain the message "no planning change". The resulting community planning is then the aggregation of the last transactions containing individual planning forecast data:

$$\hat{\boldsymbol{y}}_{\boldsymbol{pp}} = [y^+[0] - y^-[0],\ ...,y^+[H-1] - y^-[H-1]] \tag{21}$$

Smart-Contract 1. Day-Ahead planning logic

function `startPlanningPhase` (*participant* i) **:**
  emit *runPlanningAndBroadcast*(i)
function `updatePlanning` (*participant* i) **:**
  if *new forecast* (l_j^+, l_j^-) **then**
    emit *updateCommunityForecast*(i, l_j^+, l_j^-)
    if *allowedToRunPlanning* **then**



(a) Centralized aggregator approach (b) Autonomous decentralized approach

Fig. 1. Smart-Buildings community with local RES: structure, communication, and power flows.

Algorithm 1. Planning update of building i

let (*variable*) l^{+}_{*-i}, l^{-}_{*-i} = [0,...,0]
let (*variable*) e_i = [0,...,0]

function `updateCommunityForecast` (*power* l^+, l^-):
  $l^{+}_{*-i} \leftarrow l^+$, $l^{-}_{*-i} \leftarrow l^-$

function `runPlanningAndBroadcast` ():
  $e_i \leftarrow$ solve dec. planning Eq.(16) given $(l^{+}_{*-i}, l^{-}_{*-i})$
  if e_i changes **or** significant objective change **then**
    broadcast new forecast e_i on the Blockchain
  else
    broadcast "*no planning change*" on the Blockchain

This decentralized planning algorithm represents an autonomous DR scheme in the sense that all the participants tap into their flexibility to shape their power profile. Traditional DR programs generally refer to a baseline against which building are compared to check whether they appropriately responded. Unlike traditional methods that look at the last days power consumption to create this baseline, this iterative algorithm naturally leads to the declaration of such a baseline by each building, a day in advance. Then, the second part of this autonomous DR scheme consists in rewarding/penalizing participant with respect to that baseline.

3.2. Online phase: tracking and monitoring

The community is incentivized to ensure that the actual community grid power imports and exports follow the planning decided a day in advance.

3.2.1. Community billing

Smart-Contracts (2) and (3) are deployed to monitor the real-time buildings power profiles and to individually bill the participants, respectively.

In Smart-Contract (2), an event is periodically emitted to collect individual participant power consumption/production. The monitoring is carried out by *pMonitoring*(), which gathers the whole community state before calling *communityMonitoring*(). The latter compares the power profile of the entire community to the one that was announced through the cooperative planning. If the difference exceeds a given threshold, the participants must individually be penalized, for their aggregated behavior deviates too much from the planning they all agreed to follow. However, the actual billing is not yet carried out: instead, tracking errors are stored in the vector *gap* and the billing is deferred until the end of the day. The function *trackingErrorEvent*() notifies the participants that the community failed to track their planning at time instant *h*.

Smart-Contract 2. Online monitoring (periodic, every *dt*)

let (*constant*) ϵ_{thres}[]
let (*previously computed*) $\hat{y}_{pp}$[]
let (*variable*) power[:][:], gap[:][:] $\leftarrow$ 0

function `pMonitor` (*participant* i, *time* h, *power* p):
  power[i][h] $\leftarrow p$
  validate *participant i*
  if all participants *validated* **then**
    call `communityMonitoring` (h)

function `communityMonitoring` (*time* h):
  $\epsilon_{track} \leftarrow (\sum_j$ power[j][h] $- \hat{y}_{pp}[h])$
  if ϵ_{track} *exceeds* $\epsilon_{thres}[h]$ **then**
    emit *trackingErrorEvent*(h, ϵ_{track})
    for each *participant i* **do**
      gap[i][h] $\leftarrow$ power[i][h] - $e_{pp,i}[h]$

The Smart-Contract (3) ensures that each participant is properly billed individually at the end of the day as follows:

$$B_D^i = c_D^+ \cdot \sum_{h=0}^{H-1} e_i^+[h] - c_D^- \cdot \sum_{h=0}^{H-1} e_i^-[h] - \sum_{h=0}^{H-1} f_l^c\left(e_i^-[h]\right) \tag{22}$$

$$c_D^+ = \frac{\sum_{h=0}^{H-1} f_{G^+}^c\left(y_G^+[h]\right) + f_l^c\left(y_l^-[h]\right)}{\sum_{j=1}^{N}\sum_{h=0}^{H-1} e_k^+[h]}$$

$$c_D^- = \frac{\sum_{h=0}^{H-1} f_{G^-}^c[h]}{\sum_{j=1}^{N}\sum_{h=0}^{H-1} e_k^-[h]}\Bigg)$$

where c_D^+ and c_D^- represent the average daily community prices ($/kWh) of buying or selling energy from/to the grid, respectively. The function *electricityBilling*() in Smart-Contract (3) implements such a volumetric billing.

Smart-Contract 3. Accounting (periodic, every day)

let (*constant*) requiredAmount[]
let (*previously computed*) gap[][], power[][]
let (*variable*) balance[:] $\leftarrow$ 0

function `electricityBilling` ():
  for each *building i* **do**
    bill $\leftarrow$ compute B_D^i as Eq.(22) *given* power
    emit *billEvent*(i, bill)

function `pool` (*participant* i, *amount* a) **payable**
  require a >= requiredAmount[i]
  balance[i] += a
  add i to whitelist
  if whitelist *complete* **then**
    authorize `incentiveBilling` ()

function `incentiveBilling` ():
  for *each building* i **do**
    p^-, r^+ $\leftarrow$ `PRComputation` (*i, gap[i]*)
    balance[i] += $(r^+ - p^-)$
    balance[grid_id] -= $(r^+ - p^-)$

In addition to the daily volumetric bill, the participants in the community are also individually rewarded or penalized depending on the online planning tracking quality. The Smart Contract (3) rewards/penalizes the individual buildings as a result of the community behavior, through the functions *pool*() and *incentiveBilling*(). The event *pool*() is called by the community participants, with a certain amount of Blockchain currency to enable the transaction, as specified by the keyword ”payable”. As all the community actors have filled up their balances, the actual tracking reward/penalty mechanisms can be executed by *incentiveBilling*(). Variables p^- and r^+ represent the corresponding penalty and reward applied to the balance of the i^{th} participant. The index *grid_id* stands for the grid operator to which the community provides tracking services.

3.2.2. Forecast tracking

as for the implementation on the online tracking of the forecast, this could be done by solving a decentralized Model Predictive Control

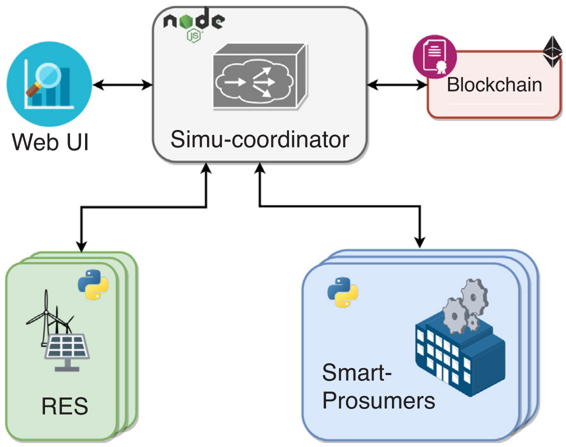


Fig. 2. Blockchain-based decentralized simulation setup.

(MPC) to reduce the error between the community profile and the forecast planning, iteratively executed by each flexible asset:

$$\min_{\{\hat{\mathbf{u}}^{\mathbf{i}}\}} \sum_{h=0}^{H_o-1} \left(y^+[h] - y^+_{pp}[h]\right)^2 + \left(y^-[h] - y^-_{pp}[h]\right)^2 \quad (23)$$

s. t. *Community model and constraints as in Eq.* (16)
Updated environmental data forecast

where H_o is the receding horizon. However, the large latency inherent to Blockchain won't allow such a decentralized MPC to be deployed at a large scale. Instead, each energy actor can opt to track their individual day-ahead forecast $e_{pp,i}$ through a local MPC:

$$\min \hat{u}^i \sum_{h=0}^{H_o-1} (e_i[h] - e_{pp,i}[h])^2 \quad (24)$$

s. t. $Eq.\ (1) - (15)\ at\ time\ h = 0, ..., H_o - 1$
Updated environmental data forecast

The Smart-Contract (2) and (3) pave the way to incentivize the decentralized tracking of the *day-ahead* planning. However, the specific implementation, such as the details of *PRComputation*() methods and the *online phase* practical implementation, goes beyond the scope of this paper. An entire framework, involving tailored penalties/reward functions, would be needed for an effective cooperative online tracking. Instead, this study focuses merely on the planning phase mechanisms and the Blockchain deployment.

3.3. Community objective

The objective function shared by all participants in Eq. (16) influences the behavior of the entire community, with respect to grid services and local resources use. The framework at hand allows participants to join any program, according to his/her own personal interest. This paper analyzed two different community programs, presented below.

3.3.1. Price-based DR and Grid-services

In this program, community participants try to minimize their daily own bill according to Eq. (22). Pricing represents a traditional means for the grid operator to influence the whole community planning phase. The cost of importing electricity from the grid is given by:

$$f^c_{G^+}(x, h) = a^q_g[h]\ x^2 + a^l_g[h]\ x + a^c_g, \quad (x \geqslant 0)$$

where the coefficients a^q_g, a^l_g vary over time. This generic form allows to take into account multiple influencing factors. Firstly, the constant term a^c_g ($) encompasses infrastructure cost. Secondly, the linear term a^l_g ("$"$/kWh$) can either be a constant energy price, a static Time-of-Use (TOU) retail price, or could follow the wholesale market prices. Lastly, the quadratic coefficient a^q_g ("$"$/(kWh)^2$) accounts for second order effects [28], such as quadratic dependency of some power plants with respect to their generated power, or the losses in the lines due to long distance energy transport. Such a quadratic dependency will have the effect to flatten the overall grid demand.

The cost of buying electricity from local RES is supposed to be time-invariant:

$$f^c_l(x) = -a_l\ x, \quad (x \leqslant 0)$$

The gain (negative cost) of exporting electricity to the grid assumes that the locally generated energy is sold to the grid at a lower price than the price to buy it locally:

$$f^c_{G^-}(x) = \alpha_s\ a_l\ x, \quad (x \leqslant 0,\ 0 \leqslant \alpha_s < 1)$$

3.3.2. Green community

This program solely aims to increase the use of local resources, regardless of their cost. The cost of buying electricity from local RES is therefore set to zero:

$$f^c_l(x) = 0$$

By joining this program, participants will foster the integration of RES into the grid, such as residential PV and windmills, and ensure that their production matches as much as possible the participants consumption.

4. Case study and discussion

This section presents a realistic test case and discusses the results of the decentralized planning algorithm described in Section 3. The code developed for this project is in open access at the repository [31] and has been run on a single Intel Core i7-4710HQ CPU (2.50 GHz × 8) with 8 GB of DDR4 RAM.

The simulation setup architecture decouples the simulation coordination from the models described in Section 2, each of them running as an independent process. Fig. 2 depicts the main components of the simulation environment.

A *NodeJS* server is at the heart of the system, redirecting the simulation messages via ZeroMQ, connecting the community to the Ethereum Blockchain via Web3JS API [35], and binding a web interface with the simulation for display and control. The *ganache-cli* private Blockchain, using EthereumJS [36], simulates a full client behavior for the purpose of these simulations. Both RES and Smart-Building simulated entities share the same *Python* class, in order to implement the same decentralized logic flow. However, while RES simply reads forecast files, a Smart-Building process is more advanced. The package CVXPY [37] models the equations presented in Section 2, and the linked Gurobi solver [38] is used to compute the solution of the local planning problem given by Eq. (16).

The test case consists in a community of N_{sb} Smart-Buildings and N_{res} RES. The parameters of the buildings are derived from the Swiss standard Minergie [39], that specifies a set of constraints on the maximum electrical power use, thermal insulation requirements, air renewal, and other useful data facilitating the model parameters extraction task. The EV parameters are derived from Tesla Model S, and the vehicles are supposed to be plugged-in during the day. The behind-the-meter PV system takes parameters from Solar's DIAMOND CS6X-310 manufacturer datasheet. Probability distributions on the parameters (e.g., temperature preferences, room size, U-values, arrival/leaving time of EV, PV size) allow to generate a fleet of similar, yet different buildings. The external temperature and irradiance are shown in Fig. 3, retrieved from MeteoSwiss.

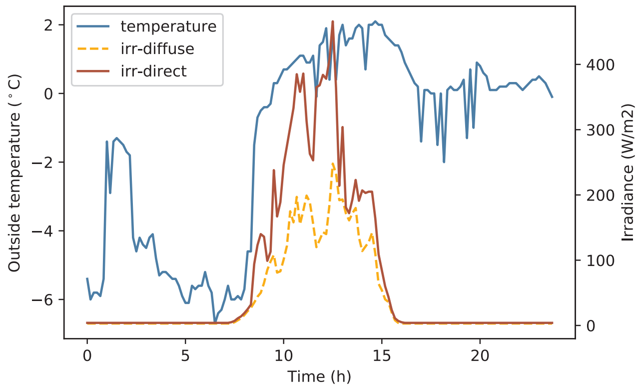


Fig. 3. Outside temperature and irradiance used for simulation.

4.1. Algorithm scalability analysis

The decentralized planning algorithm converges once all the actors don't observe any change in their forecast planning or in their own objective function. This section explores the convergence rate for an increasing community size. Due to the decentralization logic, the order in which entities take the hand (referred to as sequence) can randomly vary depending on the computational time of each node and other factors. Therefore, for each community size N_{sb}, the planning algorithm has been launched multiple times with a random sequence, leading to statistical distribution of the convergence rate. Fig. 4a (top) shows the number of iterations needed to complete the decentralized *day-ahead* algorithm Eq. (16) as a function of the community size N_{sb}, with the convex price-based objective. The trend clearly indicates a linear dependency between the number of iterations, and hence the number of transactions in the Blockchain, with respect to the community size. In the public Ethereum Blockchain, a new block takes on average 15 s to be written and the system waits for 3 blocks to actually validate the transaction. On the tested machine, an iteration needs on average 15 s to complete the updated forecast. A community of 30 buildings could therefore take up to 2 h to carry out the planning phase via the Ethereum Blockchain. Practically, the planning phase should execute during the afternoon, in order to get reliable forecast, and finish before the end of the day for the grid to be able to use it. Considering that window of 6 h, the proposed framework is therefore limited to a maximum of roughly 100 buildings. However, one must bear in mind that this statement is valid for the considered community cost function, the building models, and the chosen public Blockchain.

Since the sequence influences to some extend the convergence rate, it's interesting to analyze how does the order impact the steady-state result. One defines the steady-state cost function disparity $D_c(n)$ as the gap between the optimal solution of a sequence of N_{sb} buildings and the most optimal solution, after convergence:

$$D_c(n) = \lim_{k\to\infty} f_c^n(k) - \min_{x\in S_N}\left(\lim_{k\to\infty} f_c^x(k)\right)$$

where $f_c^n(k)$ is the community objective function at iteration k, S_N is the set of all possible permutations of $\{1...N_{sb}\}$ and n is a specific sequence in S_N. Fig. 4a (bottom) plots this metric as a function of N_{sb}. One observes that the steady-state cost function disparity is less than 0.3% regardless the size of the community. The sequence therefore does not impact the planning cost function of the community, meaning that any participant can take the lead on the decentralized algorithm without impacting the aggregated cost.

4.2. Minergie building community test case

The rest of this section considers a community of 8 Minergie Smart-Buildings and 1 local RES. The previous subsection concluded that the algorithm is not impacted by any particular sequence, and this is highlighted on Fig. 4 for the small community at hand. In this specific case, the algorithm actually converges after 3 rounds (24 iterations).

This section highlights how the algorithm is impacted by the presence of local RES, the building assets, and the type of optimization objective. Two scenarios have been considered for the community:

- *Green community with Market price*. The community consumes local resources in priority, and the cost of importing electricity from the grid is proportional to the Swiss Day-Ahead Auction Market prices [40] ($a_g^q = 0$).
- *Grid-services with Quadratic price*. The community is subject to both a linear price of energy proportional to the Day-Ahead Market prices and a quadratic price of energy. The quadratic coefficient a_g^q is set to 0.03125 ($/kWh^2) throughout the day, such that the quadratic term prevails on the linear one when the community exceeds a certain demand threshold.

For each of the aforementioned scenarios, three community configurations are simulated: (1) the presence of a wind-powered RES, (2) 50% of the buildings are equipped with PV panels, and (3) 50% of the buildings are equipped with PV panels and EV (not necessarily the same buildings). Community-level metrics can then be derived from the simulation results:

- **RES consumption**: the ratio between the daily use of energy generated locally and the daily energy generated locally.
- **PAR**: the ratio between the maximum grid power demand and the mean grid power demand.
- **Community cost**: the community cost corresponding to the *day-ahead* planning consensus.

Tables 1 and 2 regroup the simulation results considering the presented community configurations, for both scenarios *Green community (Market prices)* and *Grid-services (Quadratic price)*, respectively. The label "Individual" stands for the non-coordinated simulation, *i.e.* the Smart-Buildings optimize their consumption regardless the behavior of the other; the label "Cooperative" regroups the results of the cooperative planning algorithm. Fig. 5 helps understanding the community load profiles for both individual and cooperative logic. It's worth noticing that the simulation doesn't account for uncontrollable load, which would change the PAR as an uncontrollable baseline would appear. The simulations therefore represent a best case scenario, in which all the loads are perfectly controllable. The presented results thus set the highest limits of the community metrics, for a more realistic model/experiment would worsen the results.

The results of the *Green community* program (Table 1) indicate that

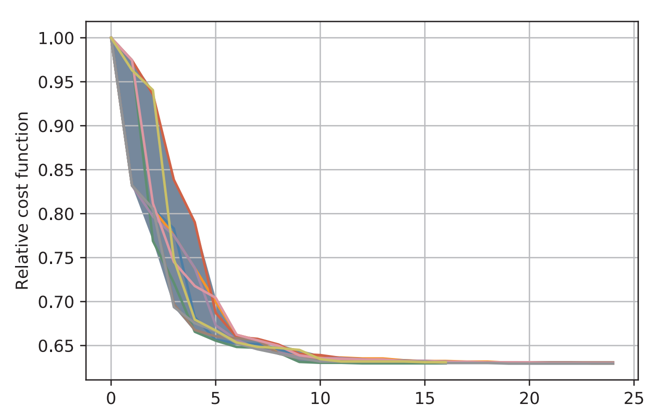


Fig. 4. Planning phase convergence for a community of 8 buildings. Individual curves refer to a given day-ahead algorithm sequence.

Table 1
Cooperative algorithm applied to *Minergie* community - Scenario *Greeen community with Market-price*.

	Scenarios					
	(1) Local RES, no PV, no EV		(2) No RES, 50% PV, no EV		(3) No RES, 50% PV, 50% EV	
Metrics	Individual	Cooperative	Individual	Cooperative	Individual	Cooperative
RES consumption (%)	56.29	94.25	0.008	100	14.65	100
Community cost ($)	29.07	24.16	20.35	19.74	39.05	36.1
PAR	8.61	8.79	9.15	9.6	7.16	8.1

the community could optimally consume the local resources, compare to a selfish algorithm. In the presence of 1 independent RES and no PV (Scenario 1), roughly half of the local resource were sold back to the grid. The decentralized cooperative algorithm allowed a local consumption of nearly 95%. When half of the community has PV installed (Scenario 2), the surplus that could not be self-consumed was almost entirely sold to the grid (RES consumption $< 1\%$). The cooperative algorithm manages to tap into the flexibility of the Smart-Buildings to entirely balance local generation with local consumption. When adding EV (Scenario 3), their profiles naturally match a bit more the local production, but most of the latter is still sold to the grid. Once again, the decentralized planning phase managed to harness all the local resources. Overall, since the price of RES (including PV) is lower than the market price, the cooperation results in a lower community cost. However, as the considered buildings are entirely made of flexible components, they all consume when the price of energy is lower, leading to a high PAR. The addition of a quadratic component solves this issue.

Adding a quadratic component to the linear pricing allows to really harness the full potential of the decentralized planning algorithm (*Grid-services* program. As each Smart-Building is concerned by the decision of the others, both for local resources use and grid consumption peak, the overall community profile is flattened (cf. Fig. 5, second and third column). In Table 2, the PARs are therefore tremendously reduced, closer to unity. In the individual selfish scheme, the costs are now much higher due to lack of communication among the community actors, and such a tariff structure would therefore be unrealistic in a non-collaborative community.

4.3. Discussion

Price structure and fairness. The community framework can provide a service to the grid manager, that can locally tune the prices to shape the entire aggregated profile. To do so, the grid operator can decide on either the linear or the quadratic coefficient. The linear coefficient influences the flexible energy distribution over the day, while the quadratic coefficient dictates the PAR of the system. To a larger extend, the grid manager could handle multiple of these smart-communities and balance the demand and supply via the price of electricity across the set of communities. The billing scheme proposed in Eq. (22) fosters the buildings to cooperatively take part in the decentralized *day-ahead* consensus, because the community aggregated data are taken into account rather than individual actions. This means that it does not reward nor penalize who consumes/produces how much and at what time, but rather spread the actions of everyone on the resulting community cost.

Prediction uncertainty. The presented test case assumes perfect forecast knowledge and exact models for prediction, as well as ignored non-controllable loads, which is unrealistic especially at residential level subject to many uncertainties. The threshold discussed in Smart-Contract (2) could be a solution to this problem. By joining an uncertainty to its forecast planning, the Smart-Building could tell the community and hence the grid about the reliability of its prediction. The Smart-Contract (2) could then integrate directly this uncertainty as the threshold to reward/penalize the Smart-Building.

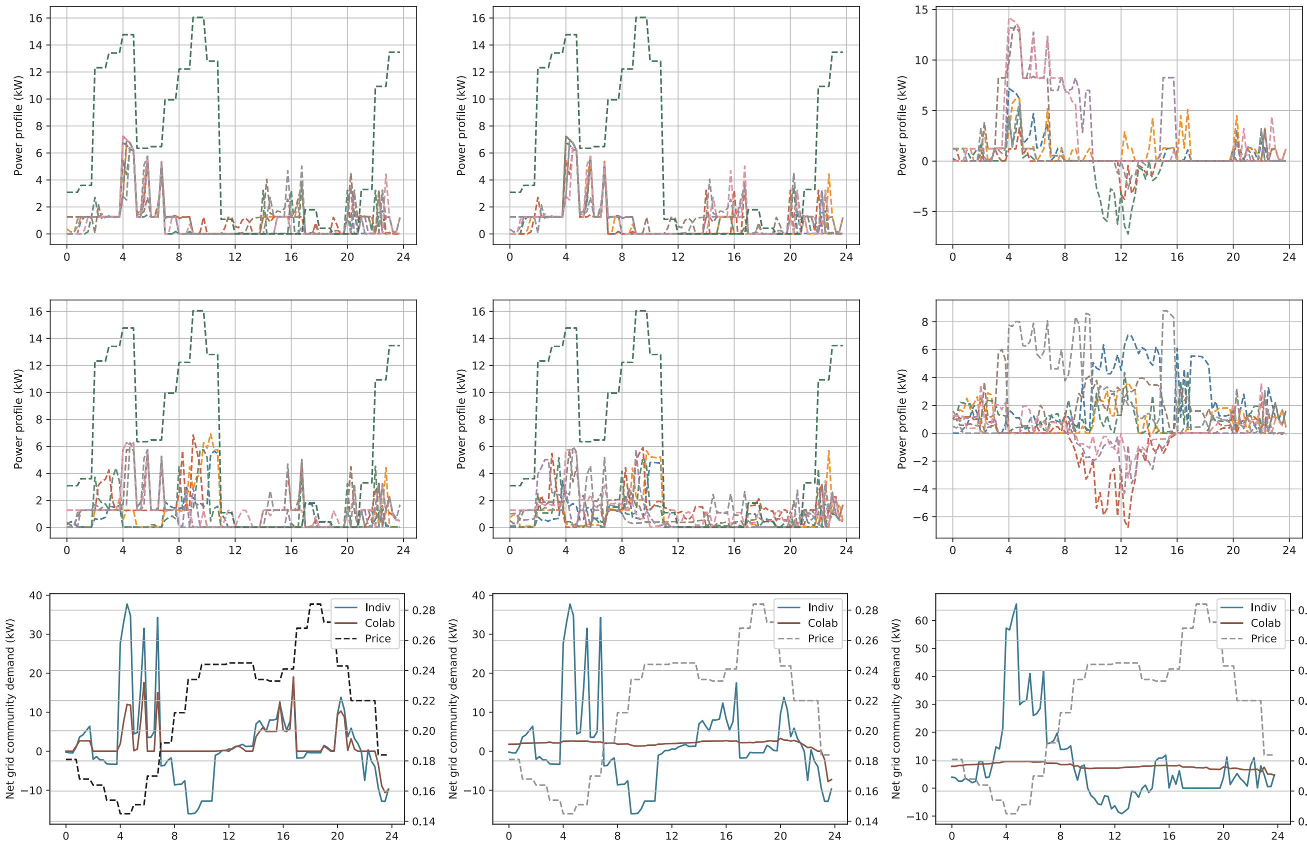
Decentralized algorithm robustness. Concerning the robustness, the presented Smart-Contracts would need to include more mechanisms to prevent non-responding nodes from jeopardizing the entire planning operation. Malicious participants could therefore deliberately block the algorithm. Such an issue could be addressed by a smarter way to pick the next building forecast, for example through bids, or simply by privatizing the Blockchain.

Reproducibility and privacy. The framework enables the participation of heterogeneous decentralized actors, possibly gathering various building types and control techniques (e.g., MPC in commercial buildings, load scheduling in homes). This important feature circumvents the need of a central controller that must know the details of every entities it supervises. The privacy of data is therefore ensured, and it also reduces the engineering time to reproduce the framework in a different environment.

Blockchain. The type of Blockchain (private, semi-private, or public) depends on the objective of the community. In the case of the sole optimization of local resources use, a public Blockchain is suitable as it allows any participant to join. When providing services to the grid, the aggregator/utility is likely to deploy a semi-private Blockchain to regulate the participants. Although the current consensus system for public Blockchain (Proof-of-Work) is debatable when applied to energy efficiency [27], Blockchain has a great potential for accelerating the decentralization of a large and complex system such as power system. The decentralized capability of Blockchain allows to implement bottom-up solutions, without depending on grid operators or waiting for changes in policies. Equipped with the right technology and motivated to better consume energy, communities of people could fasten the pace of renewable integration and DR.

Table 2
Cooperative algorithm applied to *Minergie* community - Scenario *Grid-services with quadratic price*.

	Scenarios					
	(1) Local RES, no PV, no EV		(2) No RES, 50% PV, no EV		(3) No RES, 50% PV, 50% EV	
Metrics	Individual	Cooperative	Individual	Cooperative	Individual	Cooperative
RES consumption (%)	56.29	96.96	0.008	100	14.65	100
Community cost ($)	96.08	27.97	88.67	37.73	254.14	90.45
PAR	8.61	1.61	9.15	1.37	7.16	1.22



(g) Market prices, with RES (dashed green)

(h) Quadratic prices, with RES (dashed green)

(i) Quadratic prices, with behind-the-meter PV

Fig. 5. Load profiles resulting from the Cooperative Planning Phase in the community of 8 Minergie buildings: (top row) non-cooperative load profiles (middle row) cooperative load profiles (bottom row) aggregated profiles. In the (top row) and (middle row) graphs, each colored dashed-line corresponds to a community participant power profile, either a RES or a smart-building profile, and their aggregation leads to the (bottom row) graphs.

Deployment cost. In addition to their BMS, Smart-Buildings require an energy management application to effectively shape their power profile. Optimization of individual building profile runs into this local application, generally by leveraging prices of energy emitted by the energy provider to provide local grid support [41]. In comparison to the individual optimization, the proposed cooperative framework replaces the price signal and the individual objective function by a community cost function and an exchange of power forecast information. It also replaces the utility monitoring/billing function by automated Smart-Contracts. The added cost therefore solely depends on the underlying network, in this case the Blockchain itself. Practically, running a Smart-Contracts implies fees to the participants, that vary with the state of the Blockchain system at the running time, as well as the type of Blockchain itself as discussed in the previous paragraph. In the first incarnations of the proposed framework, such a cost might be high due to the intense energy need of the promising Blockchain technology, but we envision this cost to significantly decrease as it gains popularity and maturity.

5. Conclusion

This paper presented a decentralized framework to manage the electrical consumption in a community of Smart-Buildings and local RES. Smart-Contracts allowed the participants to collaboratively decide on a planning profile that minimizes the overall aggregated cost, through a succession of local optimization processes. This planning can greatly benefit the grid operator in its day-ahead dispatch, and the online tracking reduces the need of additional capacity reserve. Moreover, the simulation results showed that the algorithm fostered the local use of energy and, under special tariff structure, the peak grid demand could be reduced. Finally, the scalability analysis highlighted that it can be applied to a community of up to 100 Smart-Buildings, given the current state of Ethereum.

The proposed framework, and technology like Blockchain in general, changes the paradigms in DR tailored to energy arbitrage. It provides a group of heterogeneous Smart-Buildings and RES with the capability to consume energy in a smarter way, without the need of a central entity that must know the details of every entity it supervises. As the future calls for an intense electrification of buildings and transportation, these decentralized assets must be efficiently incorporated in the global grid by taking into account their aggregated behavior. Beyond the grid services presented in this paper, this Blockchain-based solution allows the participants themselves to cooperatively work towards an overall decarbonized electrical grid.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by Qatar Environment & Energy Research Institute (QEERI). The author would like to thank Dr. Rachid Cherkaoui for the valuable discussions and advice.

Appendix A. Smart-building model

A.1. Thermal zones and hydronic system

The home considered in this study is made of two main air zones and a hydronic system to provide heat through a set of water pipes. The model of the temperature evolution of one zone and the connected water circuits is the result of the discretization of the following model [42] and is detailed in Eq. (25). In that equation, the state vector $\hat{\boldsymbol{x}}_{\boldsymbol{z}}(t) = [T_z(t),\ T_f(t),\ T_{wr},\ T_{ws}]^T$ contains the zone node temperature T_z, the floor node temperature T_f, the return water temperature T_{wr}, and the supply water temperature T_{ws}. The disturbance vector $\hat{\boldsymbol{d}}(t) = [P_{irr}(t),\ P_L(t),\ T_a(t)]$ regroups the sun heat gain P_{irr}, the internal load heat gain P_L, and the outside temperature T_a. The RC parameters are identified from the building construction and the hydronic design.

$$\frac{d\hat{\boldsymbol{x}}_{\boldsymbol{z}}(t)}{dt} = \begin{bmatrix} -\frac{1}{C_z}\left(\frac{1}{R_{zf}}+\frac{1}{R_{za}}\right) & \frac{1}{C_z R_{zf}} & 0 & 0 \\ \frac{1}{C_f R_{zf}} & -\frac{1}{C_f}\left(\frac{1}{R_{zf}}+\frac{1}{R_{zw}}\right) & \frac{1}{C_f R_{zw}} & 0 \\ 0 & \frac{1}{C_{wr} R_{zw}} & -\frac{1}{C_{wr}}\left(\frac{1}{R_{zw}}+\frac{1}{R_w}\right) & \frac{1}{C_{wr} R_w} \\ 0 & 0 & \frac{1}{C_{ws} R_w} & -\frac{1}{C_{ws}}\left(\frac{1}{R_w}\right) \end{bmatrix} \hat{\boldsymbol{x}}_{\boldsymbol{z}}(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{\eta_{hp}}{C_{ws}} \end{bmatrix} \hat{\boldsymbol{u}}_{\boldsymbol{th}}(t) + \begin{bmatrix} \frac{1}{C_f} & \frac{1}{C_f} & \frac{1}{C_f R_{za}} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \hat{\boldsymbol{d}}(t) \tag{25}$$

Minergie equivalent RC model of the floor-heated system.

A.2. Water tank model

The Hot Water Tank is modelled as a volume of water with a homogeneous temperature, across the whole tank. The discretized equation can therefore be expressed as:

$$\begin{aligned} T_{wh}[h+1] =& e^{-\frac{dt}{R_h C}}\ T_{wh}[h] + \left(1 - e^{-\frac{dt}{R_h C}}\right) R_h\ u_{ewh}[h] \\ &+ \left(1 - e^{-\frac{dt}{R_h C}}\right) R_h\ [U \quad d_w^h] \begin{bmatrix} T_a[h] \\ T_c[h] \end{bmatrix} \end{aligned}$$

where C is the water tank thermal capacity, $R_h = (U + d_w^h)^{-1}$ is the equivalent water-to-exterior resistance that takes into account the thermal losses U and water withdrawal d_w^h at time instant h. T_a and T_c stand for the ambient air temperature and the inlet cold water, respectively. The efficiency of the electrical system is assumed to be 1.

References

[1] Ma R, Chen HH, Huang YR, Meng W. Smart grid communication: Its challenges and opportunities. IEEE Trans Smart Grid 2013;4(1):36–46.

[2] Bazydło G, Wermiński S. Demand side management through home area network systems. Int J Electrical Power Energy Syst 2018;97(October 2017):174–85.

[3] Palensky P, Member S, Dietrich D, Member S. Demand side management: demand response, intelligent energy systems, and smart loads. IEEE Trans Industr Inf 2011;7(3):381–8.

[4] Mohsenian-Rad AH, Leon-Garcia A. Optimal residential load control with price prediction in real-time electricity pricing environments. IEEE Trans Smart Grid 2010;1(2):120–33.

[5] Gudi N, Wang L, Devabhaktuni V. A demand side management based simulation platform incorporating heuristic optimization for management of household appliances. Int J Electrical Power Energy Syst 2012;43(1):185–93.

[6] Vardakas JS, Zorba N, Verikoukis CV, Member S. A Survey on Demand Response Programs in Smart Grids: Pricing Methods and Optimization Algorithms. IEEE Commun Surveys Tutorials 2015;17(1):152–78.

[7] Kim D, Braun JE. Development, implementation and performance of a model predictive controller for packaged air conditioners in small and medium-sized commercial building applications. Energy Build 2018;178:49–60.

[8] Callaway DS, Hiskens IA. Achieving controllability of electric loads. Proc IEEE 2010;99(1).

[9] Wei T, Member S, Zhu Q, Yu N. Proactive demand participation of smart buildings in smart grid. IEEE Trans Comput 2016;65(5):1392–406.

[10] Iria J, Soares F, Matos M. Optimal supply and demand bidding strategy for an aggregator of small prosumers. Appl Energy 2018;213(June 2017):658–69. https://doi.org/10.1016/j.apenergy.2017.09.002.

[11] Ayón X, Gruber JK, Hayes BP, Usaola J, Prodanović M. An optimal day-ahead load scheduling approach based on the flexibility of aggregate demands. Appl Energy 2017;198:1–11.

[12] Hao H, Wu D, Lian J, Yang T. Optimal coordination of building loads and energy storage for power grid and end user services. IEEE Trans Smart Grid 2018;9(5):4335–45.

[13] Saad W, Han Z, Poor HV, Basar T. Game-theoretic methods for the smart grid: An overview of microgrid systems, demand-side management, and smart grid communications. IEEE Signal Process Mag 2012;29(5):86–105.

[14] Zhang C, Wu J, Zhou Y, Cheng M, Long C. Peer-to-peer energy trading in a microgrid. Appl Energy 2018;220(March):1–12.

[15] Pilz M, Al-Fagih L. Recent advances in local energy trading in the smart grid based on game-theoretic approaches. IEEE Trans Smart Grid 2017;10(2):1363–71.

[16] Mohsenian-Rad AH, Wong VWS, Jatskevich J, Schober R, Leon-Garcia A. Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid. IEEE Trans Smart Grid 2010;1(3):320–31.

[17] Chakraborty P, Baeyens E, Khargonekar PP, Poolla K, Varaiya P. Analysis of solar energy aggregation under various billing mechanisms. IEEE Trans Smart Grid 2018.

[18] Liu N, Yu X, Wang C, Li C, Ma L, et al. Energy-sharing model with price-based demand response for microgrids of peer-to-peer prosumers. IEEE Trans Power Syst 2017;32(5):3569–83.

[19] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. Cryptography Mailing list at https://metzdowd.com, pp. 1–9, 2009.

[20] Andoni M, Robu V, Flynn D, Abram S, Geach D, Jenkins D, et al. Blockchain technology in the energy sector: A systematic review of challenges and opportunities. Renew Sustain Energy Rev 2019.

[21] Ethereum Foundation. Ethereum's white paper; 2014. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper.

[22] Mengelkamp E, Gärttner J, Rock K, Kessler S, Orsini L. Designing microgrid energy markets A case study: The Brooklyn Microgrid. Appl Energy 2018;210:870–80.

[23] Hahn A, Singh R, Liu CC, Chen S. Smart contract-based campus demonstration of decentralized transactive energy auctions. In: 2017 IEEE Power and energy society innovative smart grid technologies conference, ISGT 2017; 2017.

[24] Cutler D, Kwasnik T, Balamurugan S, Booth National BS. A Demonstration of Blockchain-based energy transactions between laboratory test homes. In: 2018 ACEEE summer study on energy efficiency in buildings; 2018. p. 1–13.

[25] Pipattanasomporn M, Kuzlu M, Rahman S. A Blockchain-based platform for exchange of solar energy: laboratory-scale implementation. In: Proceedings of the conference on the industrial and commercial use of energy, ICUE, vol. 2018-October, no. October, 2019. p. 1–9.

[26] Aitzhan NZ, Svetinovic D. Security and privacy in decentralized energy trading

through multi-signatures, Blockchain and anonymous messaging streams. IEEE Trans Dependable Secure Comput 2018;15(5):840–52.
[27] Meisel M, Fotiadis L, Wilker S, Treytl A, Sauter T. "Blockchain Applications In Microgrids: an overview of current projects and concepts," 43rd. Annual conference of the IEEE industrial electronics society. 2017. p. 6153–8.
[28] Munsing E, Mather J, Moura S. "Blockchains for decentralized optimization of energy resources in microgrid networks," in 1st. Annual IEEE conference on control technology and applications. 2017.
[29] Noor S, Yang W, Guo M, van Dam KH, Wang X. Energy demand side management within micro-grid networks enhanced by blockchain. Appl Energy 2018.
[30] Pop C, Cioara T, Antal M, Anghel I, Salomie I, Bertoncini M. Blockchain based decentralized management of demand response programs in smart energy grids. Sensors 2018;18.
[31] Van Cutsem O, Ho Dac D, Boudou Perez P. Elab blockchain-based decentralized smart-buildings community simulation project. https://gitlab.com/olivancu/blockchain-dr-microgrid.git, 2018.
[32] Li X, Wen J. Review of building energy modeling for control and operation. Renew Sustain Energy Rev 2014;37:517–37.
[33] Vergura S. A complete and simplified datasheet-based model of pv cells in variable environmental conditions for circuit simulation. Energies 2016;9(5).
[34] Big-M method. Boston, MA: Springer US; 2001, p. 62–62. [Online]. Available: https://doi.org/10.1007/1-4020-0611-X_72.
[35] web3.js. Ethereum javascript api," 2019. [Online]. Available: https://web3js.readthedocs.io/en/1.0/.
[36] Ethereum Javascript Community. Ethereumjs; 2018. [Online]. Available: https://ethereumjs.github.io/.
[37] Diamond S, Boyd S. CVXPY: A Python-embedded modeling language for convex optimization. J Machine Learn Res 2016;17(83):1–5.
[38] Gurobi Optimization L. Gurobi optimizer reference manual; 2019. [Online]. Available: http://www.gurobi.com.
[39] Beyeler F, Beglinger N, Roder U. Minergie: The swiss sustainable building standard. Innovations: Technol Governance Globalizat 2009;4(4):241–4.
[40] Epex Spot SE. Swiss day ahead auction market data, https://www.epexspot.com/en/market-data/dayaheadauction, 2019.
[41] Barbato A, Capone A. Optimization models and methods for demand-side management of residential users: A survey. Energies 2014;7(9):5787–824.
[42] Vrettos E, Lai K, Oldewurtel F, Andersson G. Predictive control of buildings for demand response with dynamic day-ahead and real-time prices. In: Control Conference (ECC), 2013 European; 2013. p. 2527–34.