 doi:10.20944/preprints202210.0303.v1

Article

Flexible Loads Scheduling Algorithms for Renewable Energy Communities

Tiago Fonseca [1,*], Luis Lino Ferreira [2,*], Jorge Landeck [3], Lurian Klein [4], Paulo Sousa [5]

[1] School of Engineering of the Polytechnical Institute of Porto, Porto, Portugal; calof@isep.ipp.pt
[2] School of Engineering of the Polytechnical Institute of Porto, Porto, Portugal; llf@isep.ipp.pt
[3] LIBPhys, Department of Physics, University of Coimbra, Coimbra, Portugal; jlandeck@fis.uc.pt
[4] Cleanwatts, MIT Portugal Programme, Energy for Sustainability Initiative, Coimbra, Portugal; lklein@cleanwatts.energy
[5] School of Engineering of the Polytechnical Institute of Porto, Porto, Portugal; pbs@isep.ipp.pt
* TF calof@isep.ipp.pt, LLF llf@isep.ipp.pt

Abstract: Renewable Energy Communities (RECs) are emerging as an effective concept and model to empower the active participation of citizens on the energy transition, not only as energy consumers, but also as promoters of environmentally friendly energy generation solutions. This paper aims to contribute to the management and optimization of individual and community Distributed Energy Resources (DER). The solution follows a price and source-based REC management program, in which consumers day-ahead flexible loads (Flex Offers) are shifted according to electricity generation availability, prices and personal preferences, to balance the grid and incentivize user participation. The heuristic approach used in the proposed algorithms allows the optimization of energy resources in a distributed edge and fog approach with a low computational overhead. The simulations performed using real world energy consumption and flexibility data of a REC with 50 dwellings show an average cost reduction of 10.6% and an average increase of 11.4% in individual self-consumption. Additionally, the case-study demonstrates promising results regarding grid load balancing and the introduction of intra-community energy trading.

Keywords: Energy Community, Scheduling, Renewable energy, Flex-Offers, Algorithms

1. Introduction

Traditional energy grids have been heavily dependent on the burning of fossil fuels, like coal or natural gas, to generate electricity. This type of electricity production has therefore a negative impact on the environment while also posing geopolitical challenges for countries that must rely upon others for obtaining these vital resources. Additionally, energy generation at plants distant from consumers cause losses in its distribution infrastructure, diminishing efficiency and increasing running costs [1].

Renewable Energy Sources (RES) emerge as a green, reliable, and economically viable solution for electricity production. Given the replenishable nature of its sources, such as the sun and wind, RES enable citizens and governments to become more self-sufficient as the energy can be produced individually in a distributed manner. Distributed Energy Resources (DER) are closer to consumers, which also substantially reduces traditional distribution losses. As a result, renewable energy has undertaken a significant increase in adoption, with its growth forecast to speed up in the next five years [2]. Households, office dwellings and factories also play an increasingly prominent role in this transition [3], by installing photovoltaic panels and injecting their surplus production into the grid. Such kind of end-users are now being designated as prosumers, as they take on the role of both producers and consumers of energy simultaneously.

However, the high penetration of RES poses new difficulties to grid operators in managing and maintaining the necessary grid balance, as there is a time imbalance between peak demand and RES production and due to the highly fluctuating operation characteristic of these sources. In addition, at certain times of the year, namely Spring and Summer, there is an over generation risk which may force grid operators to curtail RES or implement negative electricity prices to force demand upwards, leading to higher operating costs and thus reducing both the environmental and economic benefits of renewable sources.

Energy Flexibility Management offers a partial solution to this problem, minimizing the impact of the introduction of RES in energy grids and preserving its economic and environmental benefits. Renewable Energy Communities (REC) offer a powerful framework in which energy sharing between members is possible and where Flexibility Management can be further explored. In this context, Flexigy [4], the project in which this research work has been conducted, aimed to develop an integrated platform for managing the energy flexibility of consumers and prosumers belonging to REC.

This paper builds on the 3-Level smart-grid architecture for REC management introduced in [5] detailing the developed algorithms to schedule the energy flexibility of home appliances, considering the member preferences and the appliances' consumption profiles. The scheduling is performed, at each one of three architecture Levels: (i) prosumer Level (on a single house or office dwelling), (ii) REC Level (to which the prosumer belongs) and (iii) at grid Level. Our approach focuses on delivering fast and scalable algorithms that do not use excessive computing resources, so that they can be applied closer to the end user in an edge and fog low-cost implementation.

The proposed method is validated using a dataset composed of energy flexibility profiles, consumption, and production of fifty dwellings. The data was collected on real-world appliances during the project. In the end, results are presented, and the benefits of the solution thoroughly analyzed.

This paper is organized into multiple sections. Section 2 presents the state-of-the-art concepts and projects related to demand side response and energy flexibility. Section 3 overviews the developed system model. Section 4 details the flex offer scheduling algorithms. Section 5 presents and discusses the results obtained from a real-world simulation. Section 6 addresses the main conclusions and future work.

2. State of the Art

Traditionally the energy grid supplies electricity to consumers through a unidirectional flow originated from very large energy production facilities, centrally controlled. However, with the introduction and evolution of DERs, like photovoltaic panels at prosumers households, demand response (DR) and Energy Flexibility Management have emerged as viable alternatives to manage these energy resources efficiently, helping to shape future smart grids.

The authors in [6] present a comprehensive review of technologies, data management, cybersecurity and how different pricing modalities can be applied in a modernized power grid. Moreover, [7], [8] study and formulate peer-to-peer energy trading in smart grids, analyzing energy routing algorithms, discussing blockchain as an enabling technology and identifying future work, such as implementing a unified messaging framework.

There are also numerous works reviewing the state of the art of DR in the literature. For example, [9] examined the benefits of DR in smart grids, while [10] focused on the developments of energy scheduling and communication technologies for DR. [11] presents a systematic literature review of the history, definitions, programs, and future development opportunities in DR. Additionally, the authors discuss the introduction of smart energy communities as a new DR participant with considerable load flexibility.

In fact, RECs, a type of smart energy communities, are groups of geographically close citizens participating in distributed energy generation as a strategy to reduce costs (self-production and sharing), but also as a novel approach to offer grid balance resources, by taking advantage of the flexibility of several electrical appliances (e.g., water heaters, HVAC systems, dishwashers), and storage. In the Flexigy project, RECs are considered as a group of prosumers connected to the same local medium- to low-voltage transformer.

 doi:10.20944/preprints202210.0303.v1

Several studies focused on small scale demand flexibility through the scheduling of home appliances. For example, [12] introduces a nonlinear optimization model for the scheduling of typical home appliances with a time-of-use electricity tariff, while [13] assesses the impacts of time-of-use tariffs on residential electricity demand and peak shifting.

Additionally, [14] approaches residential day-ahead energy scheduling for demand response in smart grids by formulating an optimization problem that, based on the service provider's electricity prices given-ahead of time, presents a solution with the desired trade-off between cost and comfort. However, the report only tests six appliances (3-schedulable and 3-non-schedulable), leading to concerns of solution applicability in real-world energy communities with hundreds of scheduling devices which result in major computational and time requirements to solve the optimization problem. This has been one of our main concerns for the proposed algorithms. Moreover, the authors in [15] propose an adaptive day-ahead load optimization and control solution with an edge and fog Internet of Things (IoT) architecture.

Domestic thermal loads such as thermal accumulators HVAC systems have also the target of research as flexible resources for DR used in RECs. These devices can be used to store excess electricity production as thermal energy considering the limits of user comfort and appliances' capacity. The authors in [13] present a peak shaving solution that predicts water usage profiles from dwelling load patterns, computes thermal losses to determine the water temperature in the tank, and consequently forecasts an optimal consumption profile. Moreover, [16] applies a fuzzy adaptive competitive algorithm as a load control model for scheduling AC units while minimizing the user's thermal comfort, while [17], [18] introduce a model predictive control (MPC) algorithm to schedule a dwellings AC units considering variable weather, occupancy, and electricity prices.

As reviewed, various works have addressed small demand flexibility scheduling. However, most of them rely on heavy optimization algorithms that require large computing resources and may take long computing time when scheduling real-world energy communities with hundreds or thousands of devices. As such, our work focuses on delivering an integrated platform for the management and optimization of renewable energy communities, unifying dwelling-Level DR, user energy flexibility and peer-to-peer community energy sharing, while maintain a distributed edge and fog architecture with low computational needs.

3. System Model and Architecture

Consider a REC where a set of prosumers can share the excess production energy between themselves and the utility grid, to promote renewable energy consumption and minimize overall costs. As described in detail in [5] at each prosumer house, there are smart devices capable of switching on/off some appliances and recording its consumption in 15-minute time slices (TSs), or smaller. These devices communicate with an edge or cloud device where scheduling decisions are taken to optimize local consumption according to i) each prosumer profile/strategy; ii) the energy flexibility of the monitored appliances and, iii) the electricity prices for the day ahead.

The next sections present the energy flexibility and Flex-Offer (FO) concepts and an overview of the prosumer profiles, which were the basis for the development of the algorithms. Additionally, the system architecture is reviewed.

3.1. Energy Flexibility

Energy flexibility, which is the capability to shift the activation of certain loads (appliances) thus changing the overall consumption profile of a facility (home) is the key concept behind the development of the scheduling algorithms.

By taking advantage of these algorithms, the platform can schedule the activation of certain loads in order to optimize the usage of the locally generated energy in individual and collective terms.

3.1.1 Flex Offer Concept

This work is based on the Flex-Offer (FO) concept, which was introduced in [19]. In its simplest form, a FO is a standardized model to represent a generic energy flexibility abstraction expressing an amount of energy or an energy profile, a duration, a price, the earliest start time, and the latest start time. Three FO examples follow:

- "Consumption of 5 kWh during 3 hours between 01:00 and 05:00, for a price of 0.25 €/kWh";
- "Consumption follows the energy profile in Figure 1, no price specified".

In these cases, the FO represent flexible electric loads (e.g., charging electric vehicles, heat pumps, equipment for domestic use) and production units (e.g., discharging batteries, photovoltaic panels).

A FO can be formally defined as a tuple:

$$f\_def = ([t_{es,}t_{ls}], \langle s^1, s^2 \ldots., s^s \rangle), \quad (1)$$

Where:

$$s^i = [a^i_{min}, a^i_{max}]$$

In equation 1, t_{es} represents the earlier start time and t_{ls} represents the latest start time for the FO. The second parameter is a list that contains a sequence of slices s that represent the energy profile of the device. Each one of these slices s^i is an energy range between a^i_{min} and a^i_{max}, usually represented in kWh which can be positive if the device consumes energy or negative if the device produces energy. We assume that the duration of each slice is a 1-time unit, adjustable to multiple sampling frequencies. In our use-case power consumption/production is sampled at 15 min intervals and defined by TimeSliceSize.

The main interest of a FOs is on having it scheduled using some criteria. The main result is that scheduled FO will also have its scheduling, i.e. the time at which the device should be turned on t_{sch}.

Consequently, equation 1 can be updated as follows in equation 2:

$$f\_sch = ([t_{es,}t_{ls}, t_{sch}], \langle s^1, s^2 \ldots., s^s \rangle) \quad (2)$$

Figure 1 displays a visual representation of a FO energy profile and respective scheduling with the t_{es} and the t_{ls} defining a time flexibility interval. The FO energy requirements are represented by energy slices (s_i). The slice energy flexibility is detailed by the difference between the a^i_{min} and a^i_{max}. The t_{sch} represents the time at which the FO was scheduled.

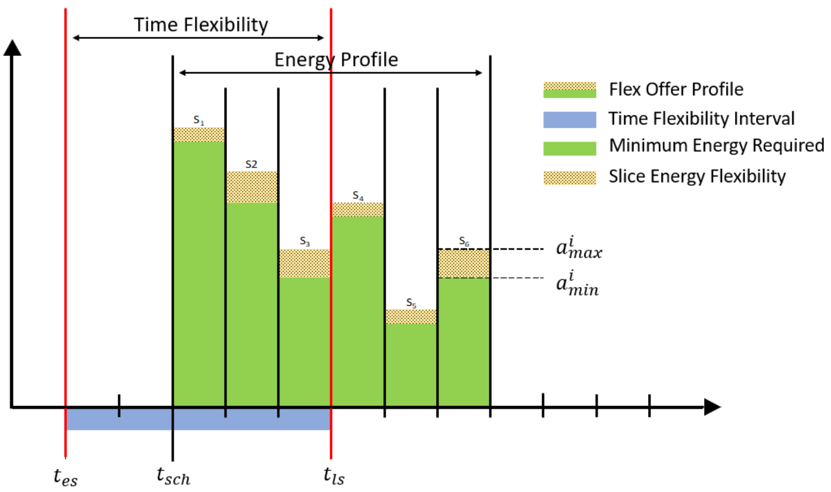

Figure 1. FO Example

3.1.2. Device Flexibility and Flex Offers Types

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 October 2022 doi:10.20944/preprints202210.0303.v1

In terms of flexibility, devices can be categorized according to two factors, present in Figure 1: (i) slice energy flexibility and (ii) time flexibility. More specifically, three distinct kinds of devices are defined, originating the three different types of FO used in this work:

- **Fixed Devices** are devices whose consumption period and amount of energy consumed cannot be modified (e.g., televisions and lights). **Fixed FOs** are used to translate these devices in the system. A Fixed FO can be formally restricted by:

$$t_{sch} = t_{es} \;\; and \; s^i a^i_{min} = s^i a^i_{max} \tag{3}$$

- **Shiftable Devices** are time flexible devices, meaning that the consumption time can be shifted within certain limits without modifying the load profile (e.g., washing machine, dishwasher). These devices offer an opportunity to optimize grid load management. Shiftable FOs translate shiftable devices into the system. A **Shiftable FO** is subject to:

$$t_{es} \; \leq \; t_{sch} \leq t_{ls} \; and \; s^i a^i_{min} = s^i a^i_{max} \tag{4}$$

- **Elastic Devices** are the most flexible, being fully adjustable in terms of usage time and instantaneous power consumption (e.g., heater, electric car). Similar to shiftable devices, elastic devices provide grid load management capabilities to a greater extent. Elastic FOs translate elastic devices into the system. An **Elastic FO** is restricted by:

$$t_{es} \; \leq \; t_{sch} \leq t_{ls} \tag{5}$$

3.2. Prosumer Profiles

Prosumer profile introduced in [5] are defined so that each prosumer can customize its objectives according to what best fits his goals and beliefs when participating on a REC. From an energy consumption point of view, there are three distinct profiles from which a prosumer can choose:

- **Bold Profile** the consumer only wants to maximize its renewable energy consumption regardless of the electricity price;
- **Cautious Profile** the consumer wants to buy energy always at the lowest total cost possible, whatever its source;
- **Local Community Supporter Profile** the consumer maximizes REC consumption irrespective of its price.

From the energy production side, whose strategy for selling the prosumer excess production can be one of the following:

- **Go-Ahead Profile** the producer wants to sell all his renewable electricity generation.
- **Tactical Profile** the producer only wants to sell its surplus of renewable generation after optimizing self-consumption.

3.3. System Architecture

As stated before, the developed algorithms follow a three-level approach introduced on [5]. This architecture aims to integrate prosumer profiles in the scheduling solution while allowing a distributed edge and fog implementation of the community energy management. The levels of this architecture are the following:

- **Level 1** - Prosumer level: executed for each prosumer to minimize the energy costs and maximize the individual renewable energy self-consumption.
- **Level 2** - Local community level: executed at the REC level to minimize overall energy costs and optimize the renewable energy-based supply via peer-to-peer energy trading and collective renewable self-consumption.
- **Level 3** - Grid level: groups small-scale flex-offers at the REC level or between RECs to respond to specific market requests from different stakeholders.

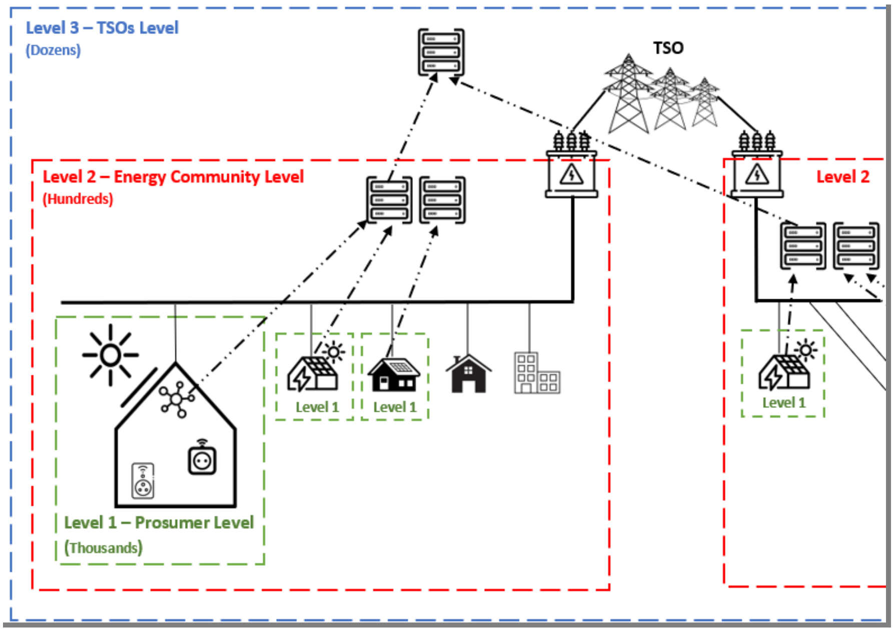

Figure 2. Three Level Architecture [5]

Figure 2 presents the system architecture from a logical point of view. Level 1, depicted in green in Figure 2 represents each prosumer dwelling with energy consumption from multiple home appliances, and, eventually, energy self-production from PV panels or other renewable sources. At this level the system collects the flexibility of different appliances on the prosumer premises and expresses this flexibility as FOs and optimizes individual self-consumption according to prosumer profiles. The algorithm can be run directly at the prosumer house (e.g.: IoT hub) in an edge computation approach, retaining data confidentiality and effectively distributing computing, as it does not need to be run on the cloud. FOs left unscheduled in this level at each edge node (each prosumer) are then sent to fog computer, handling community needs at Level 2.

Level 2, illustrated by a red dashed line in Figure 2, represents a REC connected to a single medium- to low-voltage energy transformer. At this level all the FOs generated at the community dwellings (Level 1), including the FOs partially or not fully scheduled at Level 1 are scheduled using the REC aggregated self-production. Once again, this algorithm can be run in a distributed manner at the fog level (e.g., a fog device implemented at each community). After the scheduling is performed by the algorithms operating at this level, the schedule of the community FOs is sent to the edge nodes, which will orchestrate the devices accordingly.

Finally, Level 3, depicted in blue in Figure 2, aggregates the different REC communities FOs, which were not fully or were partially fulfilled at Level 1 or 2, and sells those aggregate FOs directly on a flexibility market. Aggregation is required to generate FOs with higher power, which can be offered on balancing markets [20]. This level can be run on cloud servers, where one or more communities are combined.

4. Flex Offer Scheduling Algorithms

Following the introduced energy flexibility concept, user profiles and architecture, algorithms for the three scheduling levels are detailed in the next sections.

4.1. Level 1

Level 1 is executed for all FOs from prosumers who have chosen the Tactical profile and aim to maximize their energy self-consumption while minimizing the total cost. We assume that the cost of self-consumption is zero. The diagram in Figure 3 depicts the workflow of Level 1 algorithm.

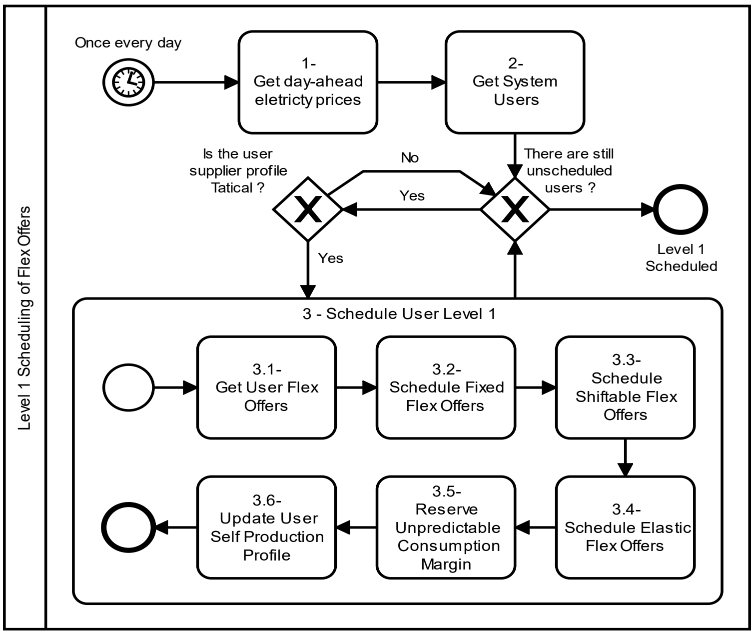

Figure 3. Level 1 workflow

This Level only includes prosumers with a Tactical profile, which have self-production capabilities. As such, the first step of Level 1 algorithms, which might be running in an edge device inside the prosumer dwelling, is to forecast the day-ahead self-production. Once forecasted, the algorithm fetches community and grid prices for the day ahead from the REC fog device.

Moreover, a forecast of the dwellings' unpredictable consumptions is generated so that the system can reserve part of the user production for unpredictable energy consumption (e.g., turning on a computer, using a vacuum cleaner or turning on the lights). This way, the self-produced energy consumption is always maximized at the prosumer level.

Finally, the production profile is updated accordingly to the scheduled consumption. In the following sections, the algorithms developed to schedule the distinct types of FOs at Level 1 are presented.

4.1.1. Level 1 Schedule of Fixed FOs

Algorithm 1 describe the solution designed to schedule fixed FOs consumptions in an optimized manner by using the prosumer self-produced energy, block 3.2 in Figure 3. Since self-produced energy is free for the prosumer, it is always more advantageous, for any buyer profile, to use the maximum self-produced energy as possible when a FO is of type Fixed FO. As such, this algorithm tries to always schedule the maximum forecasted self-produced energy at any given time.

Algorithm 1. schedSelfConsumptionFixedFO. Algorithm used at Level 1 to schedule Fixed FOs.

```
Input:
   fo - Consumption Fixed FO
   prod - Multidimensional array containing: (i) user production and, (ii)
   energy prices for each time slice
   prosumer - Prosumer
Output:
   prod - The updated prosumer production profile
```

```
1   Function schedSelfConsumFixedFO (fo, prod, prosumer)
2      t <- fo.tes
3      eProfile <- fo.getEProfile2Sched()
4      sched <- new Schedule(fo.tes)
5      For each eSlice in eProfile Do
6         e2Sched <- getMaxEConsum(prod, t, eSlice)
7         If e2Sched > 0 Then
8            sched.AddSlice(t, e2Schedule)
9         End if
10        t <- t + TimeSliceSize
11     End for
12     prod <- scheduleFO(sched, prod, fo)
13     Return prod
14  End
```

Given that a Fixed FO has no energy flexibility, its earliest start time (t_{es}) is considered the scheduling Time (t_{schd}) (line 2). In line 3 the Fixed Flex Offer consumption profile is fetched to an auxiliar variable. Moreover, a new Schedule object is created, with its start time set to the FO t_{es}.

Next, for each energy slice of the FO energy profile, the algorithm verifies how much energy consumption can be scheduled using self-production (line 6). If some or all the energy can be scheduled using the self-production, a slice is added to the schedule. This slice specifies the time, energy amount and price of the schedule energy consumption. Finally, at the *scheduleFO* method (line 12), both the FO and the production energy profile are updated, discounting the energy scheduled, and the FO schedule is saved.

4.1.2. Level 1 Schedule of Shiftable FOs

Algorithm 2 describes the algorithm designed to schedule shiftable FOs, also at Level 1, block 3.3 in Figure 3. At this Level, the biggest concern was not only to maximize self-consumption on all occasions, but instead, the algorithm should reflect the prosumer buyer profile. In effect, it can be more monetarily rewarding for a user with a Cautious buyer profile to schedule the FO with less self-consumption if the price paid for the surplus is significantly less at that slice, instead of having more self-produced energy but end up paying more for the surplus scheduled at Level 2.

As such, the approach shown in Algorithm 2 focuses on prosumers buyer profiles, as it heuristically tries to find the best fit for the FO consumption.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 October 2022 doi:10.20944/preprints202210.0303.v1

Algorithm 2: schedSelfConsumpShiftableFO

```
Input:
   fo – Consumption Shiftable FO
   prod – Multidimensional array containing: (i) user production and, (ii)
   energy prices for each time slice
   prosumer – Prosumer
Output:
   prod – The updated production profile
```

```
1   Function schedSelfConsumShiftableFO (fo, prod, prosumer)
2      cost <- MAXVALUE
3      sched <- new Schedule(fo.tes)
4      For i = fo.tes; i < fo.tls; i = i + TimeSliceSize Do
5         t <- i
6         auxSched <- new Schedule(i)
7         sum <- 0
8         eProfile <- fo.getEProfile2Schedule()
9         For each eSlice in eProfile Do
10           e2Sched <- getMaxEConsum(prod, t, eSlice)
11           consumSurplus = eSlice.energy - e2Sched
12           sum = sum + checkProfileCost(consumSurplus, t, eSlice, prosumer)
13           auxSched.AddSlice(t, e2Sched)
14           t <- t + TimeSliceSize
15        End for
16        If sum < cost Then
17           sched <- auxSched
18           cost <- sum
19        End If
20     End for
21     prod <- scheduleFO(sched, prod, fo)
22     Return prod
23  End
```

A cycle is executed to check at which of the time slices comprised between the FO t_{es} and t_{ls} is more financially advantageous to schedule the start of the FO execution (t_{sch}) (lines 5 to 20).

At the start of the loop, a set of auxiliary variables is created each time a new candidate t_{sch} is evaluated (lines 5 to 8). Next, the solution price is determined by calculating the price of the energy surplus of each time slice (lines 9 to 15). To determine it, the algorithm starts by finding the maximum self-produced energy that can be consumed by the slice and consequently the consumption surplus. Then, with the help of the *checkProfileCost* method (line 12), the electricity consumption price is summed to the total price of the solution.

The *checkProfileCost* method is the solution presented in this work to be able to optimize the Level 1 self-consumption solution without disregarding both the electricity prices at other Levels and the prosumer buyer profiles. This method uses the forecast of day-ahead prices and calculates the cost for the prosumer based on its profile:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 October 2022 doi:10.20944/preprints202210.0303.v1

- For users with a Cautious profile, the cost returned at any given time, is calculated based on the cost of the surplus energy multiplied by the grid price for that time. As such, an estimate for the scheduling of surplus energy at higher Levels is returned.
- For users with a Community Supporter profile, the cost returned at any given time, is calculated based on the cost of the surplus energy multiplied by the REC day-ahead prices at that time.
- For users with a Bold profile, the cost is indeed how much non-renewable energy is consumed in surplus of self-consumption. As such, the method returns the total amount of surplus energy in this case.

In the end, if the cost of the solution being evaluated (either price or amount of surplus energy) is lower than the cost of the previously saved schedule (line 16), both the *schedule* and *cost* variables are updated with the new solution values (lines 17 and 18).

After the best schedule is found, the *scheduleFO* method saves it and updates the FO and the self-production energy profile, accordingly, subtracting the energy scheduled at each slice to the slice available energy.

4.1.3. Level 1 Schedule of Elastic FOs

This work also focuses on bringing environmental benefits and optimizing the operational cost of elastic devices such as thermal accumulators and air conditioners by scheduling their day-ahead energy consumption according to its time of use tariffs and the prosumer profiles. Future work will be developed concerning battery storage and other forms of elastic energy flexibility. Algorithm 3 details the heuristic algorithm designed to create a FO for elastic devices, which is later scheduled at the same Level as a fixed FO.

Algorithm 3. schedElasticDevice

```
Input:
   prosumer -Prosumer to which the device belongs
   tMax – Maximum temperature defined by the user to maintain his comfort
   tMin – Minimum temperature defined by the user to maintain his comfort
   tStart – Temperature at the start
   prices - List with the energy self-production values of the user and energy
   prices of the different grid suppliers available.
   powerCom - average power consumption per time slice.
Output:
   FO – The created fixed FO for scheduling
```

```
1   Function generateHeuristicElasticEProfile
2      t <- new Date(0,0,0)
3      temp <- tStart
4      totalCost <- 0
5      While (auxtime < end) Do
6         nextCoolDownTime = getNextCoolDownTime(tMin, temp, t)
7         If isLowestPriceUntilNextCooldown(nextCoolDownTime, prices)Then
8            newTemp <- calculateNewTemp()
9            If newTemp < tMax Then
10                temp <- heatUp ()
11                consump.add(powerCon, t)
12           Else
13                temp <- coolDown()
14           End
15        Else
16           temp <- coolDown()
17        End
18        t <- t + TimeSliceSize
19     End While
20     FO <- new FO(fixed, consumptions)
21     Return FO
22  End
```

The heuristic approach to solve elastic devices scheduling can be simply explained as an attempt to use the thermal appliance as a conditioned thermal battery.

For example, a client has a water heater that must maintain water between a specified comfort range of temperatures, t_{min} and t_{max}. Our approach focuses on heating-up the water at the slices with the lowest price before the water cools down below t_{min}. However, the water cannot be heat up above t_{max}. If the water is below t_{min}, the algorithm heats up disregarding the price, until meeting desired comfort levels.

When for example a client has self-production, the most cost and environmentally effective way to use its energy resources is to use surplus energy, which is free, to heat up water, successfully storing renewable energy as heat.

Algorithm 3 does exactly that. First a set of auxiliary variables are created (lines 2 to 4), including a variable holding the actual temperature of the device. Then, in a loop (lines

5 to 19) each time slice is examined, as follows. First, the next cool down time is calculated (line 6), based on temperature change equations previously inserted on the system for this specific device.

The cool down time is the predicted time at which it is forecasted that the temperature of the water goes below t_{min}. Note that the calculation of the forecast of the cooldown time can be improved over time, for example with client hot water consumption patterns. This way the algorithm can more efficiently calculate the cooldown time and maintain comfort temperatures whilst optimizing energy consumptions.

Next, the program checks if the current slice price is the lowest until the cooldown time (line 7). If so, energy is used to heat up water, and the new temperature is calculated. Otherwise, no energy is used, and the water continues to cooldown (line 13). In the end (line 20 and 21), a new Fixed FO is created and returned to be scheduled with algorithm 1 with the consumptions scheduled by this algorithm. Note that it results in a Fixed FO since the start time is already defined, resulting a FO without time flexibility, but it can maintain some consumption flexibility.

The main result of Level 1 scheduling can be a set of unscheduled FOs, together with another set partially fulfilled FOs, which change from being flexible or elastic to fixed FO. Alternatively, it is also possible that all FOs from a prosumer are fulfilled, and no further scheduling is performed for FOs from this prosumer. Or a mix of both alternatives.

4.2. Level 2

Level 2 starts by getting the users' production surplus to generate a community energy production profile. Then, it collects and shuffles in a random order all unscheduled FOs of Level 1. A FO is considered unscheduled when there is still energy left unscheduled. Finally, the FOs pending from the previous Level are scheduled according to the prosumer buyer profile and the FO type (steps 1.4, 1.5, and 1.6 in the diagram in Figure 4).

Note that in this Level, the FOs scheduling order is randomly selected, addressing the equity problem that may arise from scheduling always in the same order, as the first to be scheduled may benefit from a large community excess production available than the last (considering that a typical RES does not produce the energy enough to satisfy the consumption of all REC members).

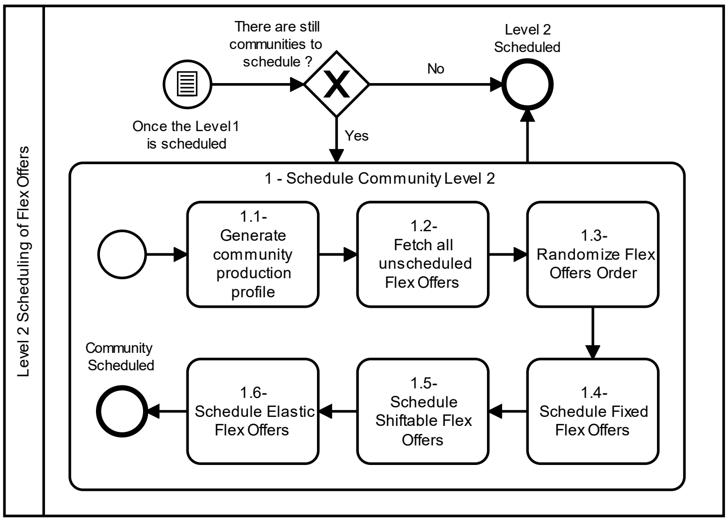

Figure 4. Level 2 workflow

4.2.1. Level 2 Schedule of Fixed FOs

Algorithm 4 presents the pseudocode designed to schedule Fixed FOs at Level 2, which are scheduled before all others given their reduced flexibility.

Algorithm 4. schedLevel2FixedFO

```
Input:
  fo - Consumption Fixed FO
  prod - Multidimensional array containing: (i) community production, (ii)
  energy prices for each time slice
Output:
  prod - Updated production profile
```

```
1   Function schedFixedFO (fo, prod)
2     t <- fo.tes
3     sched <- new Schedule(t)
4     2eProfile <- fo.getEProfile2Sched()
5     prosumer <- getFOProsumer(fo)
6     For each eSlice in eProfile Do
7       e2Sched <- eSlice.energy
8       sched <- schedSlice(t, e2Sched, prosumer, prod, sched)
9       t <- t + TimeSliceSize
10    End for
11    prod <- scheduleFO(sched, prod, fo)
12    Return prod
13  End
```

Once again, since a Fixed FO has no energy flexibility, its t_{es} is also the resulting scheduling time t_{sch} (line 2). Then, the algorithm initializes a variable with the FO energy profile and another with the user buyer profile. Next, the algorithm schedules each energy slice of the FO energy profile using the *schedSlice* method. This method is analysed further ahead in algorithm 5. It guarantees an adequate energy schedule according to the user profile. Finally, at the *scheduleFO* method, both the FO and the production energy profile are updated, discounting the energy scheduled, and the FO schedule is saved in the database.

4.2.2. Level 2 Schedule of Shiftable FOs

Algorithm 4 describes the pseudocode designed to schedule the Level 2 Shiftable FOs.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 October 2022 doi:10.20944/preprints202210.0303.v1

Algorithm 4. schedLevel2ShiftableFO

```
Input:
    fo - Consumption Shiftable FO
    prod - Multidimensional array containing: (i) community production, (ii)
    energy prices for each time slice
Output:
    prod - Updated production profile
```

```
1   Function schedShiftableFO (fo, prod)
2       consumPrice <- MAXVALUE
3       eProfile <- fo.getEProfile2Schedule()
4       sched <- new Schedule(fo.start)
5       prosumer <- getFOProsumer(fo)
6       For i = fo.tes; i < fo.tls; i = i + TimeSliceSize Do
7         t <- i
8         auxSched <- new Schedule(i)
9         sum <- 0
10        eProfile <- fo.getEProfile2Sched()
11        For each eSlice in eProfile Do
12          e2Sched <- eSlice.energy
13          auxSched <- schedSlice(t, e2Sched, prosumer, prod, auxSched)
14          sum <- sum + auxSched.getPrice(t)
15          t <- t + TimeSliceSize
16        End for
17        If sum < consumPrice Then
18          sched <- auxSched
19          consumPrice <- sum
20        End If
21      End for
22      prod <- scheduleFO(sched, prod, fo)
23      Return prod
24  End
```

The cycle in lines 6 to 21 is executed to check in which of the time slices comprised between the t_{es} and t_{ls} is more monetarily advantageous to plan the FO t_{sch} (lines 17 to 20). Finally, the *scheduleFO* method saves the best schedule in the database, updates the FO and the production energy profile accordingly.

4.2.3. Level 2 Schedule of Elastic FOs

As described previously in section 4.1.3 the elastic scheduling algorithms are executed at Level 1 for the users with forecasted self-production available. For all other user's elastic devices, the scheduling is done at Level 2. The algorithm used is similar to the one used at Level 1, consequently it will not be described in here.

4.3. Level 3

The Level 3 algorithms schedule FO at grid Level, but they are out of the scope of this paper as this topic has been extensively researched before.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 October 2022 doi:10.20944/preprints202210.0303.v1

These algorithms work by aggregating small FOs into larges FO which can be scheduled at grid Level or submitted to a flexibility market. This schema allows the participation of small consumers in demand response, which otherwise would not have a significant impact on energy grid balancing as traditionally energy-intensive industrial users and large customers have by intentionally modifying their consumption patterns.

The authors in [21] theorize about a voluntary local flexibility market where users sell their flexibility, which is then grouped by energy aggregators and sold, reducing costs for all involved stakeholders.

For example, [15] introduces an optimal scheduling algorithm based on load constraints linked to the dwelling occupant comfort. Similarly, [16] uses aggregation of energy flexibility expressed by market players as the key to balancing energy supply and demand. After their creation and acceptance, the FOs are aggregated, preserving their flexibility. Afterward, the scheduling is performed based on forecasts to achieve a greater balance of the grid. Next, the FOs are disaggregated and returned to the prosumer. Once the execution is carried out, billing is conducted, and depending on the benefits of the FO for the utility company, an incentive may be provided to the prosumer.

5. Case Study

This section presents the case study used to test the algorithms and evaluate a set of environmental objectives and economic benefits accomplished by the introduction of management and optimization of REC members' energy consumption and production.

5.1. Simulation Approach and Test Data

The carried-out simulation follows the approach illustrated in Figure 5. At first, the system is feed with data related with: historical energy consumption patterns, energy prices, weather information and users' FOs for the next day. In the end, the system outputs the user's FOs schedule according to the algorithm presented in this paper, which maximizes the consumption of both user and REC self-production energy, while meeting the users' preferences.

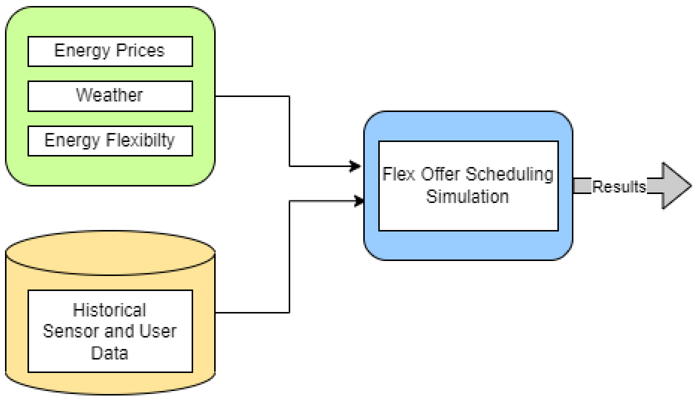

Figure 5. Simulation approach

The energy prices used for the simulation were obtained from the Iberian wholesale energy market, OMIE, with a 1-hour granularity. The energy prices for transactions inside the REC were set at 80% of the OMIE price during the period, a 20% discount compared to OMIE prices. Finally, and to evaluate the effectiveness of the scheduling algorithm, the average of the daily price was considered as the flatline tariff for energy consumption, enabling the comparison between the cost before and after the application of the scheduling algorithms. Figure 6 shows the energy prices per kWh used in this simulation.

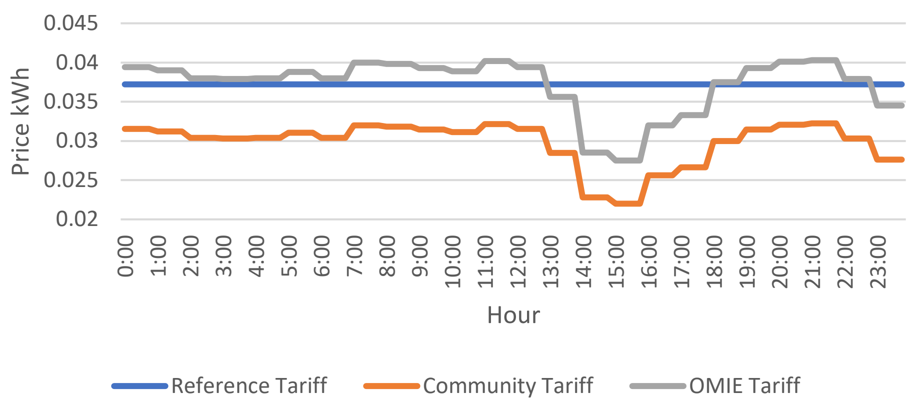

Figure 6. Simulation energy prices

User consumption flexibility is based on real-world data collected with the help of smart-energy meters for 50 different dwellings. The testing data accounts for a total of 137 consumption FOs, in which 42 are fixed FOs corresponding to consumptions of computers and fridges, 71 are shiftable FOs characterizing the flexibility of appliances such as washing machines and dishwashers, and 24 are elastic FOs describing the flexibility of water heaters.

Regarding energy self-production FOs, three different test scenarios were studied:

- Scenario 1: 20% of the REC dwellings have self-production;
- Scenario 2: 40% of the REC dwellings have self-production;
- Scenario 3: 60% of the REC dwellings have self-production.

The testing data also encompasses a real-world mix of all user/prosumer profiles. From a buyer perspective there were 22 Cautious, 19 Bold and 9 Local Community Supporter. From a supplier point of view there were 44 Tactical and 6 Go-Ahead profiles. Table 1 summaries the testbed data information.

Table 1. Summary of case study test data

Number of Dwellings	50 Dwellings
Dwellings with Self Energy Production	(1) 10 dwellings (20%) (2) 20 dwellings (40%) (3) 30 dwellings (60%)
Number of Fixed FOs	42
Number of Shiftable FOs	71
Number of Elastic FOs	24
Types of Prosumer Buyer Profiles	22 Cautious 19 Bold 9 Community Supporter
Types of Prosumer Supplier Profile	44 Tactical 6 Go-Ahead

5.2. Results and Evaluation

After applying the scheduling algorithms, the obtained results show a significant improvement, both economically and environmentally, not only for end-users, but also for all involved players in the energy market value chain. This section details the obtained results.

5.2.1. Environmental Results

To access the degree of accomplishment of environmental objectives, two Key Performance Indicators (KPIs) are assessed:

- KPI 1 – User self-consumption
- KPI 2 – REC consumption

KPI 1 measures the total amount of user energy self-consumption, in kWh, of each community member with available self-production, and compares, in percentage, the values before and after the algorithms are applied. Note that users with Go-Ahead profiles are not considered, since all their self-production is sold, and its surplus consumption is not optimized by the algorithms. Table 2 depicts the average increase of user self-consumption before and after the algorithms were applied, for each test case scenario.

Table 2. Increase of user energy self-consumption (KPI 1)

Scenario	Average Increase of User Self Consumption (%)
1	16.4
2	8.9
3	8.8

These results show that an increase on self-consumption was achieved by all test case scenarios, as on average, each user consumed 11.4% more of self-produced energy after the algorithms presented in this paper, mainly Level 1, were applied to their dwelling. Figure 7 shows in more detail, the KPI 1 results attained for each user in test scenario 3.

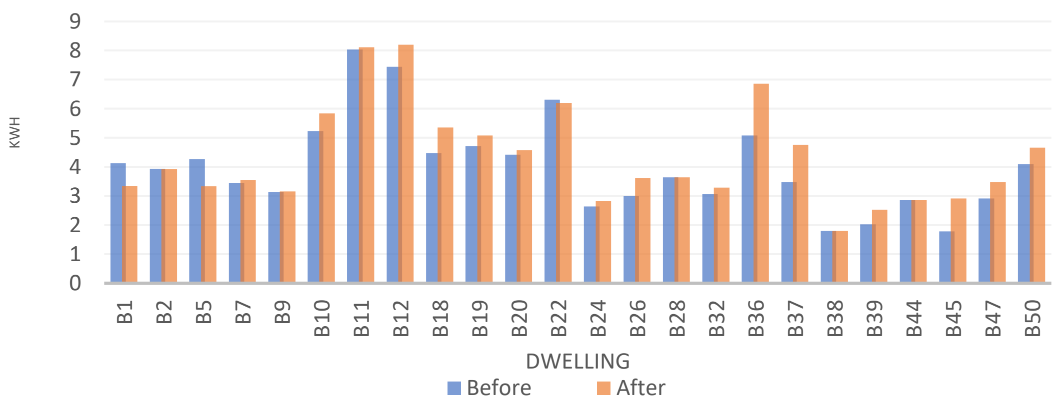

Figure 7. User energy self-consumption per dwelling before(blue) and after(orange) the algorithms

As expected, there is an overall increase in self-consumption among community members. However, some show a decrease, such as the users in dwellings B1, B5 and B22. These situations are explained by the Cautious buyer profile chosen by these users that prioritize the total energy cost minimization. The algorithms consider that fact, and schedule their FOs at the lowest price possible, even if it means consuming less self-produced energy at hours where energy costs are higher, as leftovers would lead to a higher total cost. In Section 5.2.2 the economic benefits are analyzed, and for example users B1, B5 and B22 show a decrease in total energy cost, as expected. Also note that not all cautious users suffer a decrease in self-consumption, that is the case of B12 and B50, with an increase of 10% and 13% in self-consumption, respectively, which shows that even cost oriented users can benefit from the environmentally friendly nature of the optimization.

The second KPI measures, in percentage, how much of the total energy consumption in the REC comes from intra-community energy trading after Level 2 algorithms are applied.

Table 3 shows the increase of the total REC energy production, which is consumed by REC members, for each test case scenario.

Table 3. REC consumption increase after scheduling for each test case scenario

Scenario	Community Consumption After Scheduling (% of total consumption)
1	16.5
2	25.7
3	28.3

When examining the second KPI in the case study, results show that on average of the three test scenarios, 23.5% of the total consumption registered in the REC was satisfied by intra-community energy trading. The results also show that the higher the number of self-producing users, the higher the community consumption achieved. In scenario 3, where 60% of the houses have self-production, approximately 28% of the total energy consumed in the community came from energy produced by other members in the community. Figure 8 shows, for test scenario 3, the percentage of the total energy consumption from each energy source before and after the algorithms were applied.

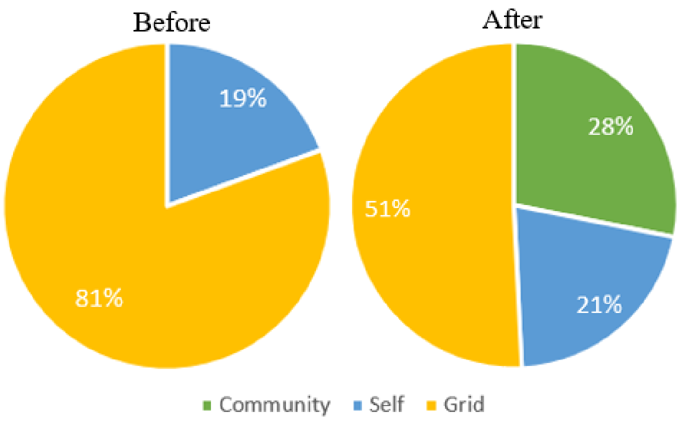

Figure 8. Energy consumption in the simulated REC before(left) and after(right) the algorithms were applied

As seen in the results of KPI 1 and KPI 2, REC members' renewable self-consumption is optimized according to their profiles, and a significant intra-community renewable based consumption is achieved. Not only they increase the integration of distributed RES in the grid, leading to higher renewable energy consumption, but also, as the energy is consumed locally, our approach helps to reduce energy transmission losses, accomplishing an environmental benefit.

5.2.2 Economical results

Similarly, to the previous section, to better comprehend and examine the accomplishment of economic goals the following KPI is introduced:

- KPI 3 – User total energy cost

The third KPI quantifies the total spending on energy, in euros, of each community member, and compares, in percentage, the values before and after the algorithms are applied. Once again, users with a Go-Ahead buyer profile are not considered for this indicator, as selling all their self-produced energy due to contractual terms impedes the cost optimization. Also note that the total cost regards only to consumption cost, since the profit made by selling self-production to other REC members is not considered.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 20 October 2022 doi:10.20944/preprints202210.0303.v1

Table 4 depicts the average reduction of users' total energy cost before and after the algorithms were applied, for each test case scenario.

Table 4. Average reduction, in percentage, of each prosumer total energy cost

Scenario	Average Reduction of Users' Total Energy Cost (%)
1	9.2
2	10.6
3	12.2

These results show a reduction in total energy cost in all test case scenarios, as on average, each user consumption cost is 10.6% less after the algorithms presented in this paper are applied to optimize their energy needs. Figure 9 details, the total cost before and after for some of the users in test scenario 3.

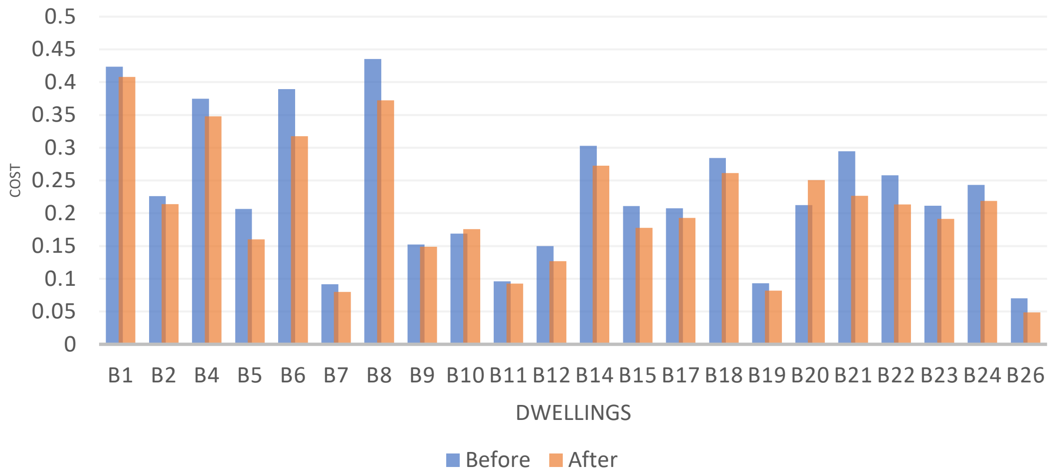

Figure 9. Dwelling total energy cost before(blue) and after(orange) the algorithms

As projected, the graph shows an overall reduction in users' total energy cost. As mentioned before, users with cautious profile, that had decreased their self-consumption before, such as the users in B1, B5, B22, now show a reduction in their total energy cost, attaining their profile objectives. Also, regarding buyer profiles, some Bold users like B10 and B20 saw their energy total cost increasing. However, since the Bold profile aims to maximize renewable energy consumption disregarding the cost, their personal objectives were accomplished.

5. Conclusions

Energy produced from RES has emerged as green, reliable, and environmentally friendly solution for the replacement of traditional energy production methods, which are heavily dependent on the burning of fossil fuels. Moreover, RES, such as sun and wind can be individually harnessed by citizens, allowing for energy self-sufficiency, and the reduction of transmission losses. As a result, RECs are emerging as an effective concept and model to empower the active involvement of citizens on the energy transition as promoters of RES and the participation on the energy markets.

This paper aimed to contribute to the management, scheduling and optimization of individual and community energy consumption and production in a REC. It follows on a previous REC architecture and introduces heuristic algorithms that aim to address different players' economic and social needs. The algorithms are organized in a distributed edge and fog approach and are architected for low computational overhead.

The carried-out test case scenario with 50 REC members aimed to simulate a real-world community, with diverse buyer and supplier profiles, energy flexibility and production capabilities. The results demonstrate very promising results as it encourages the use of RES, helping producers reduce the initial investment pay-out time by not only maximizing the use of self-produced energy but also by selling the energy surplus to other community members at a profitable price.

Current work is being developed to update the algorithms to consider the scheduling of optimized battery energy storage and consumption and the introduction of electric vehicles in a vehicle-to-grid fashion. Future work should evaluate these algorithms against real-world implementations, with a more diversified list of dwellings, appliances, flexibilities, and seasonal data.

Author Contributions: Conceptualization, T.F, L.L.F, J.L, L.K and P.S; methodology, T.F, L.L.F, J.L, L.K and P.S.; software, T.F.; validation, T.F, L.L.F, J.L, L.K and P.S.; formal analysis, L.L.F and P.S.; investigation, T.F, L.L.F, J.L, L.K and P.S; resources, X.X.; data curation, T.F.; writing—original draft preparation, T.F and L.L.F; writing—review and editing, T.F, L.L.F, J.L, L.K and P.S; visualization, T.F.; supervision, L.L.F.; project administration, L.L.F.; funding acquisition, L.L.F.. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by project FLEXIGY, nº 034067 (AAC nº 03/SI/2017) POCI-01-0247FEDER-034067, financed through National Funds through FCT/MCTES (Fundação para a Ciência e Tecnologia) and co-financed by Programa Operacional Regional do Norte (NORTE2020), through Portugal2020 and the European Fund for Reginal Development (FEDER).

Data Availability Statement: Simulation test data can be found here: https://github.com/calofonseca/flexigyData

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] K. Surana and S. M. Jordaan, "The climate mitigation opportunity behind global power transmission and distribution," *Nature Climate Change 2019 9:9*, vol. 9, no. 9, pp. 660–665, Aug. 2019, doi: 10.1038/s41558-019-0544-3.

[2] "Executive summary – Renewables 2021 – Analysis - IEA." https://www.iea.org/reports/renewables-2021/executive-summary?msclkid=ef4584aeb1ba11eca8d13ddfe4fdcea2 (accessed Sep. 01, 2022).

[3] S. Carley and D. M. Konisky, "The justice and equity implications of the clean energy transition," *Nature Energy 2020 5:8*, vol. 5, no. 8, pp. 569–577, Jun. 2020, doi: 10.1038/s41560-020-0641-6.

[4] "Flexigy." http://flexigy.vps.energy (accessed Sep. 02, 2022).

[5] T. Fonseca, L. L. Ferreira, L. Klein, J. Landeck, and P. Sousa, "Flexigy Smart-grid Architecture", doi: 10.5220/0010918400003118.

[6] I. Alotaibi, M. A. Abido, M. Khalid, and A. v. Savkin, "A Comprehensive Review of Recent Advances in Smart Grids: A Sustainable Future with Renewable Energy Resources," *Energies 2020, Vol. 13, Page 6269*, vol. 13, no. 23, p. 6269, Nov. 2020, doi: 10.3390/EN13236269.

[7] O. Jogunola *et al.*, "State-Of-The-Art and Prospects for Peer-To-Peer Transaction-Based Energy System," *Energies 2017, Vol. 10, Page 2106*, vol. 10, no. 12, p. 2106, Dec. 2017, doi: 10.3390/EN10122106.

[8] J. Abdella and K. Shuaib, "Peer to Peer Distributed Energy Trading in Smart Grids: A Survey," *Energies 2018, Vol. 11, Page 1560*, vol. 11, no. 6, p. 1560, Jun. 2018, doi: 10.3390/EN11061560.

[9] P. Siano, "Demand response and smart grids—A survey," *Renewable and Sustainable Energy Reviews*, vol. 30, pp. 461–478, Feb. 2014, doi: 10.1016/J.RSER.2013.10.022.

[10] H. T. Haider, O. H. See, and W. Elmenreich, "A review of residential demand response of smart grid," *Renewable and Sustainable Energy Reviews*, vol. 59, pp. 166–178, Jun. 2016, doi: 10.1016/J.RSER.2016.01.016.

[11] M. Esmaeil Honarmand, B. Hays, M. Shafie-Khan, and P. Siano, "An Overview of Demand Response: From Its Origins to the Smart Energy Community," *IEEE POWER & ENERGY SOCIETY SECTION*, Jul. 2021, doi: 10.1109/ACCESS.2021.3094090.

[12] D. Setlhaolo, X. Xia, and J. Zhang, "Optimal scheduling of household appliances for demand response," *Electric Power Systems Research*, vol. 116, pp. 24–28, Nov. 2014, doi: 10.1016/J.EPSR.2014.04.012.

[13] J. Torriti, "Price-based demand side management: Assessing the impacts of time-of-use tariffs on residential electricity demand and peak shifting in Northern Italy," *Energy*, vol. 44, no. 1, pp. 576–583, Aug. 2012, doi: 10.1016/J.ENERGY.2012.05.043.

[14] X. Jiang and L. Wu, "Residential power scheduling based on cost efficiency for demand response in smart grid," *IEEE Access*, vol. 8, pp. 197324–197336, 2020, doi: 10.1109/ACCESS.2020.3034767.

[15] S. V. Oprea and A. Bâra, "Edge and fog computing using IoT for direct load optimization and control with flexibility services for citizen energy communities," *Knowl Based Syst*, vol. 228, p. 107293, Sep. 2021, doi: 10.1016/J.KNOSYS.2021.107293.

[16] F. Luo *et al.*, "Optimal dispatch of air conditioner loads in southern China region by direct load control," *IEEE Trans Smart Grid*, vol. 7, no. 1, pp. 439–450, Jan. 2016, doi: 10.1109/TSG.2014.2388233.

[17] A. Izawa and M. Fripp, "Multi-Objective Control of Air Conditioning Improves Cost, Comfort and System Energy Balance," *Energies 2018, Vol. 11, Page 2373*, vol. 11, no. 9, p. 2373, Sep. 2018, doi: 10.3390/EN11092373.

[18] G. Serale, M. Fiorentini, A. Capozzoli, D. Bernardini, and A. Bemporad, "Model Predictive Control (MPC) for Enhancing Building and HVAC System Energy Efficiency: Problem Formulation, Applications and Opportunities," *Energies 2018, Vol. 11, Page 631*, vol. 11, no. 3, p. 631, Mar. 2018, doi: 10.3390/EN11030631.

[19] M. Boehm *et al.*, "Data Management in the MIRABEL Smart Grid System," *Proceedings of the 2012 Joint EDBT/ICDT Workshops on - EDBT-ICDT '12*, 2012, doi: 10.1145/2320765.

[20] R. Rocha, M. Albano, L. L. Ferreira, F. Relvas, and L. Matos, "The Arrowhead Framework applied to energy management," *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*, vol. 2018-June, pp. 1–10, Jul. 2018, doi: 10.1109/WFCS.2018.8402357.

[21] P. Olivella-Rosell *et al.*, "Local Flexibility Market Design for Aggregators Providing Multiple Flexibility Services at Distribution Network Level," *Energies 2018, Vol. 11, Page 822*, vol. 11, no. 4, p. 822, Apr. 2018, doi: 10.3390/EN11040822.

[22] X. Ayón, J. K. Gruber, B. P. Hayes, J. Usaola, and M. Prodanović, "An optimal day-ahead load scheduling approach based on the flexibility of aggregate demands," *Appl Energy*, vol. 198, pp. 1–11, Jul. 2017, doi: 10.1016/J.APENERGY.2017.04.038.

[23] M. Boehm *et al.*, "Data Management in the MIRABEL Smart Grid System," *Proceedings of the 2012 Joint EDBT/ICDT Workshops on - EDBT-ICDT '12*, 2012, doi: 10.1145/2320765.